

Министерство образования Российской Федерации

А.В. Стариков В.Н. Харин

**Управление сложными проектами
в интегрированных САПР**

Воронеж 2002

УДК 681.32

Стариков А.В., Харин В.Н. Управление сложными проектами в интегрированных САПР. – Воронеж. гос. университет. Воронеж, 2002. – 135 с.

В монографии рассмотрены концептуальные основы построения управляющих подсистем интегрированных САПР, известных как мониторинговые системы, предложены подходы и конкретные способы реализации мониторинговой системы в САПР микропроцессорных средств вычислительной техники (МСВТ), освещены методические и практические аспекты ее использования в управлении сложными проектами.

Издание предназначено для специалистов, занимающихся вопросами создания, сопровождения и эксплуатации современных интегрированных САПР, а также студентов и аспирантов, изучающих дисциплины “Автоматизированное проектирование изделий электронной и вычислительной техники” и “Разработка систем автоматизированного проектирования”.

Ил. 25. Табл. 3. Библиогр.: 114 назв.

Научный редактор д-р техн. наук, проф. В.Е. Межов

Рецензенты: кафедра САПР и информационных систем ВГТУ,
д-р техн. наук, проф. М.Г. Матвеев

© Стариков А.В., Харин В.Н., 2002
© Оформление. Воронежский
государственный университет, 2002

Введение

Организация сложных вычислительных процессов, связанных с проблемно-ориентированной обработкой данных (в частности при автоматизированном проектировании изделий электронной и вычислительной техники), требует создания специальных способов и программных средств поддержки стратегии обработки проектной информации. Эта потребность обусловлена:

высокой информационной сложностью обрабатываемых объектов, приводящей к необходимости декомпозиции проекта на составные части, обработке каждой части по отдельности и последующему “сшиванию” частей;

высокой функциональной сложностью решаемых задач, приводящей к необходимости декомпозиции процесса обработки на этапы и выполнения каждого этапа с помощью соответствующей функциональной подсистемы;

распределенным характером обработки, которая реализуется функциональными подсистемами, расположенными в различных узлах информационно-вычислительной сети;

динамически изменяющимися структурой и состоянием среды обработки, которая часто создается путем комплексирования существующих и вновь разработанных функциональных подсистем.

Средства поддержки стратегии сложной обработки данных составляют основу систем управления проектами (*Project Management System*). В разной степени они также представлены в системах автоматизированного проектирования (САПР), включая и CASE (*Computer-aided Software/System Engineering*), в системах поддержки принятия решений (*Decision Support System*) и в других типах программных продуктов. При этом проект расширенно трактуется как полная совокупность данных, специфицирующих обрабатываемый объект в различных информационных проекциях или аспектах. Проект подвергается процессу целенаправленной обработки в вычислительной среде, образованной множеством функциональных подсистем, каждая из которых реализует одну или несколько проектных процедур, генерирующих проектное решение.

К основным задачам управления проектами обычно относят: формирование структуры проекта; планирование процесса обработки проекта и его опера-

тивное перепланирование в соответствии с результатами промежуточных вычислений; обеспечение взаимодействия и координация действий различных функциональных групп – коллективов специалистов, участвующих в организации процесса вычислений и анализе результатов с применением соответствующих функциональных подсистем; обеспечение и контроль логической целостности проекта; мониторинг состояния проекта; учет сделанных в проекте изменений; контроль выполнения календарного плана работ; контроль и обеспечение достоверности информации при хранении и передаче проектных решений между функциональными подсистемами и другие.

Важность перечисленных задач управления проектами значительно возрастает при параллельном выполнении проектных процедур в среде распределенной обработки данных. Совокупная сложность задач управления в этом случае становится сопоставимой со сложностью вычислительных задач, а их решение требует разработки специальных способов и автоматизированных средств поддержки.

Традиционно функции по организации управления выполнением проектных процедур возлагаются на управляющую подсистему, называемую также мониторинг системой (МС). За последние 10-15 лет за рубежом сформировалось новое направление исследований, связанное с проблемой создания системных сред (инфраструктур) открытых САПР (*CAD Framework*), в рамках которого решаются перечисленные выше задачи управления. Поскольку концептуальные основы построения инфраструктур являются важнейшим элементом “ноу-хау” фирм, специализирующихся на их разработке, в научно-технической литературе они освещены недостаточно.

В нашей стране разработка соответствующих мониторинговых систем проводилась в “недрах” ведомственных НИИ и КБ и обычно сопровождалась реализацией конкретных САПР. Причем часто имела место недооценка важности общих задач управления проектными работами, и по давней традиции отечественных разработчиков САПР основной акцент делался на создании эффективной алгоритмической базы и прикладных программных модулей на ее основе. Со временем, однако, стало очевидным, что наличие разрозненных эффективных инструментальных средств само по себе не способно обеспечить разработку качественных проектов в сжатые сроки. Для этого требуется более тесная интегра-

ция программных модулей в рамках используемой САПР, а также применение общих механизмов и средств управления и координации работ при выполнении проектов.

В данной монографии рассмотрены вопросы создания адекватной концептуальной модели процесса распределенной обработки проектных данных и разработки на ее основе инвариантных компонентов математического, программного, лингвистического и информационного обеспечения мониторной системы интегрированной САПР микропроцессорных средств вычислительной техники (МСВТ).

В главе 1 дан аналитический обзор существующих моделей процесса распределенной обработки данных (на примере проектирования в интегрированных САПР), рассмотрены особенности этого процесса в среде, построенной путем комплексирования инструментальных программных средств, предложена интерпретация концепции сквозного проектирования, использованная при построении интегрированной САПР.

В главе 2 рассмотрены элементы лингвистического, математического и программного обеспечения мониторной системы: язык описания информационной модели процесса проектирования (ЯОМ), транслятор с ЯОМ – генератор рабочей модели (ГРМ), программы оперативной корректировки рабочей модели и ряд сервисных программ для работы с моделью.

В главе 3 сформулирован ряд задач поддержки оперативного управления проектом и процессом проектирования, представлены способы и средства их решения, базирующиеся на использовании рабочей модели процесса проектирования.

В главе 4 описаны структура мониторной системы и функциональные возможности важнейших ее подсистем, а также унифицированная схема информационного взаимодействия проектирующих подсистем и процедура аттестации инструментального программного обеспечения.

В заключении представлены основные научно-практические результаты, полученные авторами в ходе реализации мониторной системы интегрированной САПР МСВТ.

Глава 1. Информационная модель процесса проектирования

Мониторная система (МС) – это обслуживающая подсистема САПР, предназначенная для организации рационального управления выполнением проектных процедур и взаимодействием проектирующих подсистем. В общем случае процесс проектирования характеризуется наличием большого числа плохо формализуемых факторов, что предопределяет использование информационного моделирования для его исследования. В основе информационного (кибернетического) моделирования, являющегося одним из видов математического моделирования, лежит отражение некоторых информационных процессов управления, позволяющее оценить состояние и поведение реального процесса.

Следовательно, для формулирования требований к составу, структуре и функциональным возможностям базовых компонентов МС, прежде всего, необходимо разработать адекватную информационную модель процесса проектирования. В этой связи были исследованы свойства и особенности процессов проектирования различных типов, проанализированы существующие модели процессов проектирования, обоснован выбор концептуальной модели в виде асинхронного последовательно-параллельного конвейера проектирования, конкретизировано содержание информационной модели, базирующейся на модели конвейера проектирования.

Предложена информационная модель, представляющая суперпозицию двух подмоделей – информационной модели проекта и информационной модели технологического маршрута проектирования – и отражающая процесс проектирования в следующих проекциях: структура проекта, структура технологического маршрута, календарный план выполнения работ над проектом. Формализованным представлением частных моделей, образующих обобщенную информационную модель, являются: конечный гиперграф $H(X, E; I)$ – для информационной модели проекта, конечный орграф $G(P, D)$ – для информационной модели маршрута проектирования.

1.1. Обзор моделей процесса проектирования

Ряд теоретических работ в области САПР посвящен вопросам создания адекватной концептуальной модели процесса проектирования [1–3]. Основная цель этих работ заключается в обеспечении понятийной базы, необходимой для формулирования критериев выбора архитектуры САПР, а также для анализа функционирования и оптимизации различных проблемно-ориентированных приложений САПР. Применительно к общей проблеме управления проектированием концептуальная модель может рассматриваться в качестве отправной точки при исследовании вопроса об информационном моделировании процесса проектирования. Ниже представлены основные понятия и положения, полученные при анализе существующих концептуальных моделей процесса проектирования и учтенные при разработке информационной модели процесса проектирования в интегрированной САПР.

1.1.1. Концептуальная модель процесса проектирования

В работе [1] проанализированы две концептуальные модели процесса проектирования: упрощенная модель и уточненная модель, построенная на базе упрощенной модели.

Упрощенная концептуальная модель процесса проектирования (рис. 1.1) строится исходя из следующих предположений:

- цель проектирования неизменна (по крайней мере, в течение некоторого промежутка времени);

- для работы над проектом требуются знания технологии определенного типа;

- в процессе проектирования создается проект, представленный проектной информацией, которая может быть использована для реализации объекта проектирования тем или иным способом.

Очевидно, что в этой упрощенной концептуальной модели не нашел отражения ряд важных характеристик процесса проектирования. Она нуждается в уточнении, чтобы учесть следующие существенные обстоятельства.

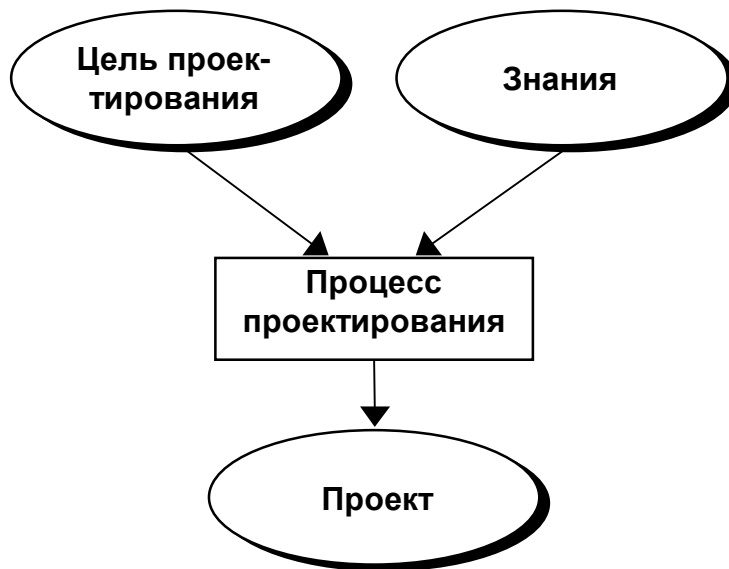


Рис. 1.1. Упрощенная концептуальная модель процесса проектирования

Процесс проектирования не является изолированным. Обычно он включается в процесс более высокого уровня, называемый средой проектирования и присутствующий во всех проектных организациях [1]. Процесс проектирования инициируется, протекает и завершается в среде

проектирования. Более подробно особенности среды проектирования описаны ниже.

Процесс проектирования, как правило, имеет итерационный характер. Часто начало проектирования связано с неопределенностью как в требованиях технического задания (ТЗ), так и в способах получения удовлетворительного решения. Проектные решения, полученные с помощью эвристических методов в начале процесса проектирования, подвергаются серии последовательных уточнений до тех пор, пока не будет получен приемлемый результат (с точки зрения требований ТЗ). Потребность в выполнении итераций в процессе проектирования обусловлена не только неопределенностью в требованиях ТЗ, что является следствием высокой функциональной сложности объектов проектирования, но и сложностью информационных аспектов проектирования высоконадежных систем [4, 5].

Процесс проектирования характеризуется вложенным циклом управления. Во внутреннем цикле, содержащем проектные процедуры синтеза, расчета и анализа, спецификация проекта, отраженная в ТЗ, неизменна. Информация об отклонении проекта от требований ТЗ, полученная процедурой верификации, передается процедуре оптимизации, в которой решается вопрос об изменении внутренних параметров объекта проектирования или его структуры и передачи управления проектной процедуре расчета. Внешний цикл управления замыкается на процессе более высокого уровня (например, на уровне проектной орга-

низации) и приводит, как правило, к изменению цели проектирования, т.е. к корректировке общего ТЗ. Как показано в [6], аналогичная схема управления описывает и каждый отдельный этап проектирования (рис. 1.2).

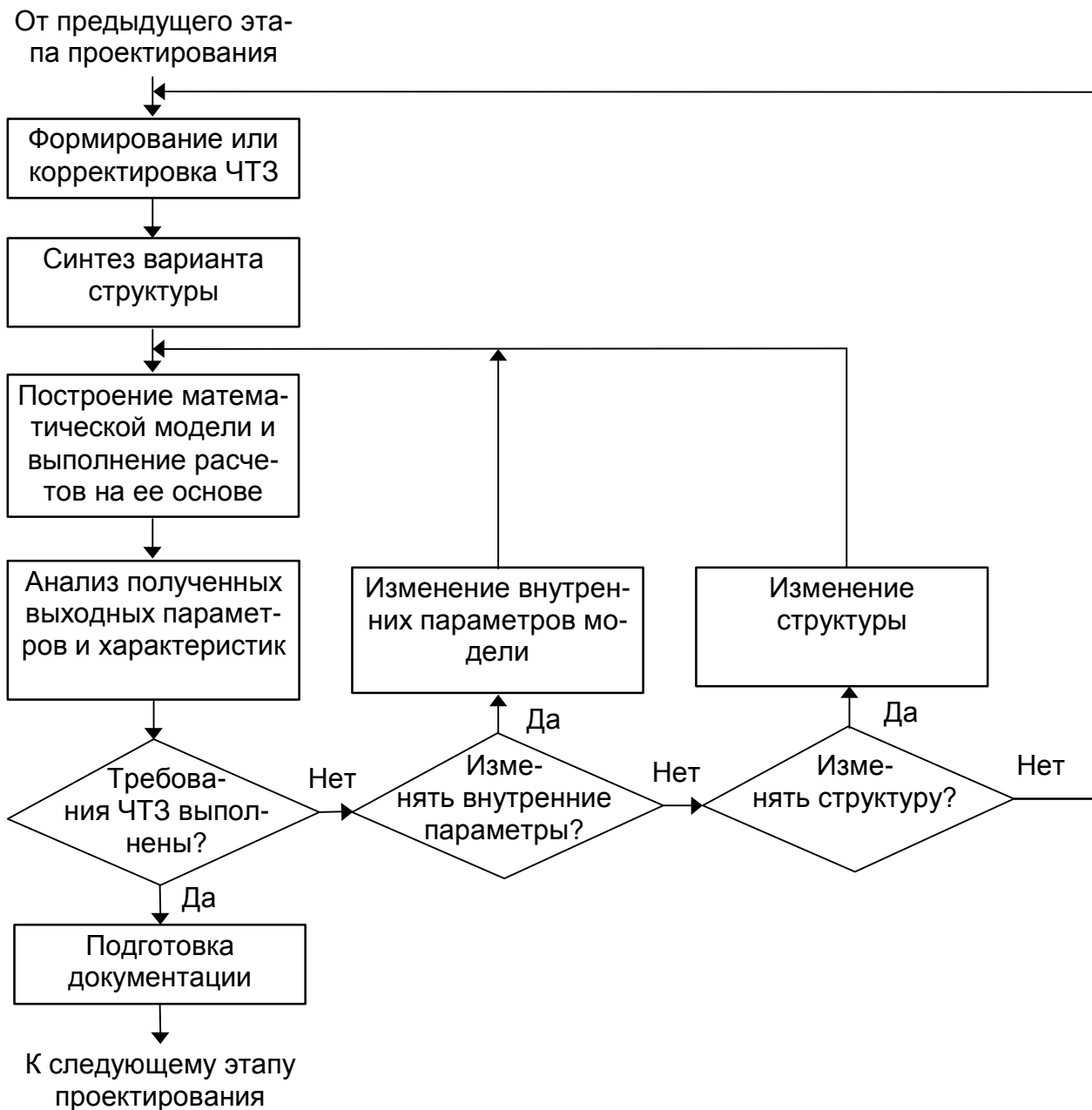


Рис. 1.2. Типовая схема отдельного этапа проектирования

В процессе проектирования формируется информация, которая необходима не только для реализации объекта проектирования. В концептуальной модели процесса проектирования должна быть представлена полная информация, необходимая как для этапа анализа, так и для всех других возможных этапов (процессов). Например, для испытания изделия, его продажи и сопровождения требуется часть информации, получаемой в процессе проектирования. В этой

связи особое значение приобретают базы данных проектов и, в частности, архивы проектов [7].

Функциональная сложность объектов проектирования, с одной стороны, и многоаспектный (а также многоуровневый – в рамках каждого аспекта) характер процесса проектирования, с другой стороны, обуславливают необходимость применения иерархической декомпозиции при проектировании. Иерархическая декомпозиция предполагает разбиение сложной проблемы на ряд более простых проблем, которые, в свою очередь, разделяются на ряд еще более простых проблем и так далее. Процесс декомпозиции продолжается до тех пор, пока каждая из полученных проблем сможет быть решена с использованием имеющихся ресурсов.

Строгая иерархическая декомпозиция должна удовлетворять следующим двум требованиям [8, 9]. Во-первых, на каждом уровне иерархии проект должен иметь законченный вид на данном уровне детализации. Во-вторых, на любом уровне иерархии декомпозиция должна полностью охватывать фрагмент, функцию или проблему. В общем случае иерархическая декомпозиция может существовать в различных формах, включая декомпозицию планов, декомпозицию проектов, нисходящее управление и другие [8].

Таким образом, сложность проблемы проектирования обуславливает необходимость декомпозиции объекта проектирования и процесса проектирования. Декомпозиция проекта является важнейшим этапом процесса проектирования. Как отмечается в [10], уже на этом этапе в значительной степени определяются сложность, временные параметры и показатели надежности проектируемого объекта. Критерии, которые используются при выполнении декомпозиции объекта, могут различаться в разных аспектах проектирования. В большинстве случаев вначале выполняется декомпозиция по функциональным признакам, т.е. система разбивается на ряд подсистем, каждая подсистема, в свою очередь, разбивается на ряд функциональных узлов и так далее, а лишь затем каждый из полученных фрагментов может быть разбит по другим (например, количественным) признакам. Может оказаться, что решение задачи декомпозиции с учетом важных критериев на поздних этапах проектирования, приводит к необходимости пересмотра результатов декомпозиции объекта, полученных на ранних этапах. В общем виде решение задачи декомпозиции объекта проек-

тирования не автоматизировано, хотя разработаны методы, алгоритмы и программные средства для ее решения применительно к некоторым аспектам проектирования (например, для топологического проектирования).

Процесс проектирования может создавать зависимые процессы (подпроцессы) проектирования, формируя частные ТЗ (ЧТЗ) для компонентов проектируемого объекта. На рис. 1.3 показана типовая схема проектирования, представленная несколькими иерархическими уровнями. На каждом иерархическом уровне решается задача проектирования, включающая подзадачи синтеза (С), расчета (Р), анализа (А), оптимизации (О) и документирования (Д). Совокупность этих уровней (этапов) во взаимодействии образует процесс проектирования [6]. Синхронизация подпроцессов проектирования и выделение необходимых им ресурсов являются функциями исходного (родительского) процесса проектирования.

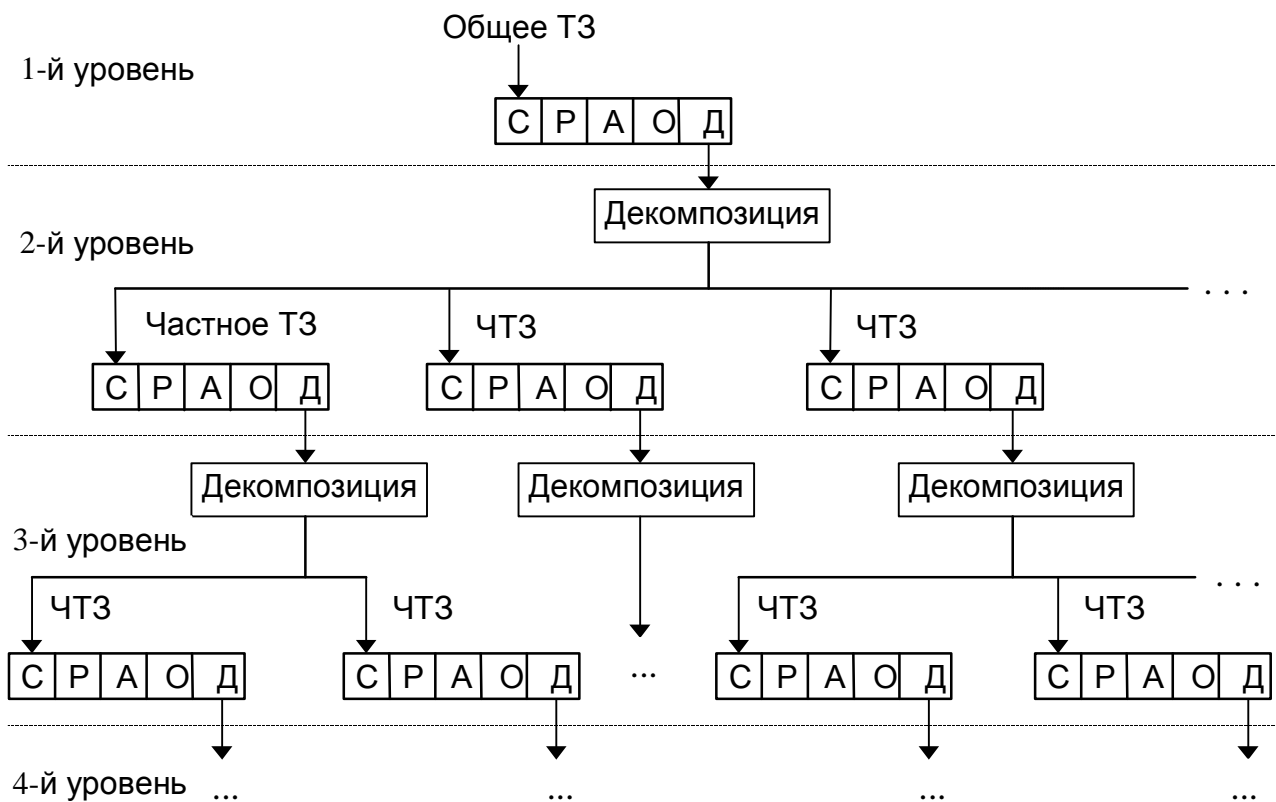


Рис. 1.3. Типовая иерархическая схема проектирования

Каждый процесс проектирования характеризуется не только своей функцией, но и ресурсами (к последним относятся проектировщики, средства автоматизации их труда, время, денежные средства и т.п.). Знание фактов и методов, необходимых для выполнения проектирования, также может рассматри-

ваться в качестве специфического ресурса. Ресурсы (и в частности знания), вовлекаемые в процесс проектирования, также не являются неизменными. В ходе работы над проектом могут привлекаться дополнительные силы проектировщиков, внедряться новые средства автоматизации их труда, продлеваться сроки выполнения проекта и увеличиваться капиталовложения. Знания накапливаются в процессе выполнения конкретной работы над данным проектом и привлекаются извне. Следует отметить, что при этом фактически параллельно выполняются два процесса – проектирование и осознание результатов проектирования. Причем осознание результатов проектирования может рассматриваться как своеобразная проектная процедура, т.е. можно говорить об одном процессе проектирования.

1.1.2. Особенности среды проектирования

Дальнейшее уточнение концептуальной модели процесса проектирования может быть продолжено путем раскрытия особенностей среды проектирования. Среда проектирования реализует следующие функции:

- прием запросов от других процессов на выполнение некоторых проектных работ, т.е. на создание новых процессов проектирования;

- согласование методов представления заданий на проектирование и результатов проектирования, а также методов создания, управления и завершения всех вновь создаваемых процессов проектирования;

- создание процессов проектирования и координация их выполнения;

- управление ресурсами с целью их распределения между процессами проектирования;

- предупреждение конфликтов при распределении имеющихся ресурсов и повышение эффективности использования этих ресурсов;

- управление накоплением, хранением и представлением знаний, которые, как отмечалось выше, являются специфическим ресурсом, используемым в процессе проектирования.

Иерархическая структура среды проектирования согласуется с существующей в проектных организациях структурой управления. Действительно, широкое применение методологии иерархического проектирования [3], обусловленное все более возрастающей функциональной сложностью объектов проек-

тирования, предполагает в большинстве случаев иерархическую структуру подразделений, занятых в работе над проектом. В свою очередь структурирование самого процесса проектирования, т.е. его функциональная декомпозиция на этапы, проектные процедуры и проектные операции, обусловленная нетривиальностью решаемых задач, приводит к иерархической структуре управления, показанной на рис. 1.4. Даже если проектирование выполняется в организации, имеющей матричную структуру управления [8], управление каждым проектом может быть приведено к иерархической структуре.

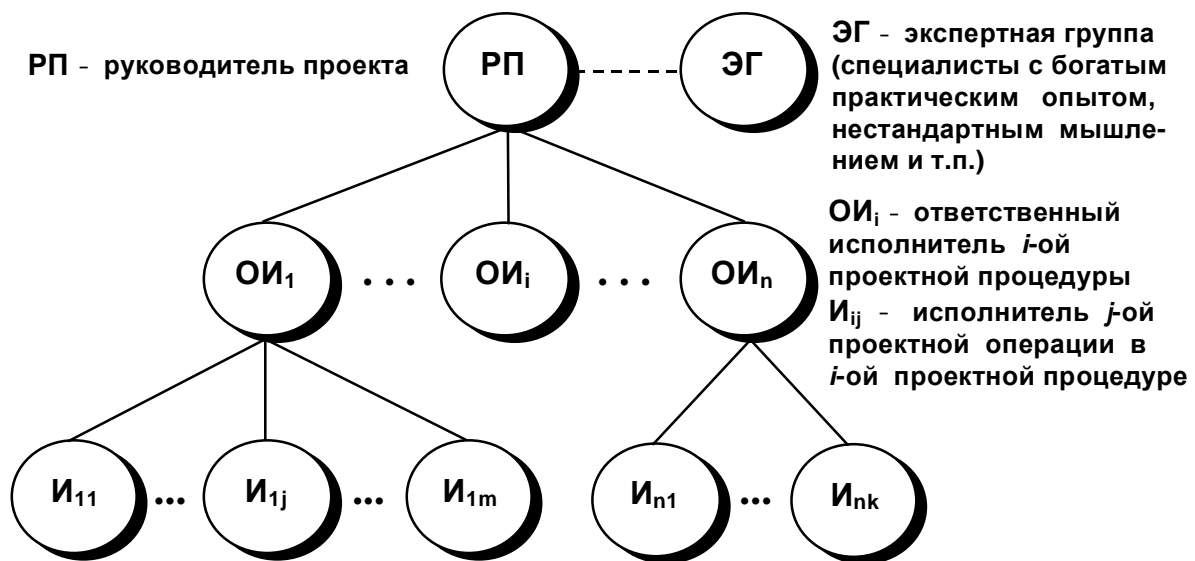


Рис. 1.4. Иерархическая структура управления проектированием

Внедрение средств вычислительной техники (ВТ) в процесс проектирования также приводит к необходимости дальнейшего уточнения модели процесса проектирования. В частности, возникает вопрос о формализации представления информации (спецификаций объекта проектирования и знаний), поскольку ЭВМ (точнее выполняемая ею программа) способна обрабатывать информацию, представленную только в формализованном виде. Так как ЭВМ является частью среды проектирования, другим вопросом, требующим своего решения, является учет ряда ограничений этой среды, обусловленных возможностями как самой ЭВМ (быстродействие, объемы оперативной и внешней памяти, конфигурация технических средств и ряд других характеристик), так и соответствующего программного обеспечения.

1.1.3. Сетевая модель процесса проектирования

Приведенный анализ модели процесса проектирования предполагал иерархическую структуру процессов среды проектирования, поскольку процесс проектирования имеет возможность порождать подпроцессы только в среде, подчиненной его собственной. Однако, как указывается в [1], это лишь частный случай более общей ситуации, которая состоит в том, что любой процесс проектирования может устанавливать связь с любым процессом среды проектирования и выполнять запрос на создание нового подпроцесса. Таким образом, для представления процесса проектирования более подходит сетевая модель с выделенными структурными (иерархическими) уровнями.

Представленная сетевая модель процесса проектирования недостаточно эффективна, поскольку процессы, находящиеся на одном и том же иерархическом уровне не имеют доступа к информации о процессах среды проектирования этого же уровня. По этой причине отдельные процессы проектирования в момент их создания получают некоторую информацию о процессах среды своего уровня. В частности, им могут быть переданы:

- информация о возможностях процессов среды проектирования;

- информация о выдаче запросов к процессам среды проектирования и обмене с ними данными.

Подобное допущение позволяет выполнить моделирование процесса проектирования и его интерфейсов без рассмотрения всех прочих процессов проектирования. Наличие последних предполагается, но интерфейс с ними организуется исключительно в рамках существующего процесса среды проектирования. На рис. 1.5 показан элементарный блок для построения сетевой модели процесса проектирования [1].

В рассмотренной выше модели процесса проектирования нигде явно не учитывался временной параметр. Тем не менее, каждый процесс является протяженным во времени, т.е. имеет начало, конец и существует в течение некоторого промежутка времени, часто называемого “временем жизни” процесса. Продолжительность “жизни” процесса часто зависит от других процессов. В частности, время “жизни” процесса проектирования не может превышать времени “жизни” его среды проектирования.

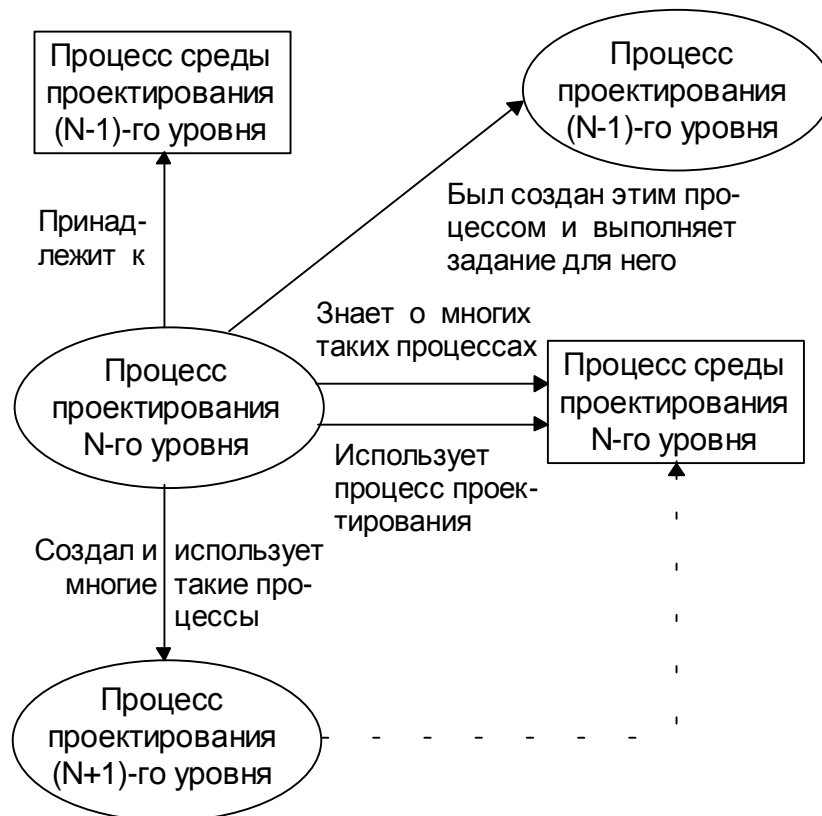


Рис. 1.5. Элементарный конструктивный блок сетевой модели процесса проектирования

При описании задач управления процессами часто прибегают к использованию диаграмм состояний. Диаграмма состояний – это конечный орграф, вершины которого обозначают множество некоторых рабочих состояний процесса, а дуги – множество возможных переходов из одного рабочего состояния в другое. В каждый момент времени любой процесс среды проектирования может находиться в одном из трех рабочих состояний [1]:

пассивном, если в среде не существует ни одного процесса проектирования;

выполнимом, если в среде существует хотя бы один выполнимый или выполняющийся процесс проектирования;

приостановленном, если в среде существует хотя бы один процесс проектирования, но ни один из существующих процессов не может выполняться (например, из-за недоступности в данный момент необходимых процессу ресурсов).

Рабочее состояние среды проектирования тесно связано с рабочими состояниями принадлежащих ей процессов проектирования. Можно выделить следующие рабочие состояния процесса проектирования [1]:

существование. Процесс переходит в это состояние сразу после того, как он создан в среде проектирования по запросу некоторого другого процесса проектирования;

выполнимый. Процесс переходит в это состояние после того, как он инициирован;

выполнение. Процесс переходит в это состояние для выполнения некоторой последовательности действий, направленных на решение задачи проектирования;

ожидание. Выполняющийся процесс переходит в это состояние, если ему требуется поддержка других процессов. Например, процессу могут потребоваться ресурсы для создания новых подпроцессов, поддержка процесса более высокого уровня или процесса среды проектирования;

восстановление. Процесс переходит в это состояние после того, как в ходе выполнения процесса проектирования происходит особая (исключительная) ситуация.

Понятия процесса проектирования и среды проектирования, неформально введенные выше, можно использовать как основу для описания процессов автоматизированного проектирования. В работе [1] процесс автоматизированного проектирования определяется как подчиненный процесс проектирования, который выполняется в среде, обеспечиваемой ЭВМ, или более точно – в рамках САПР. Поскольку САПР должна обеспечивать автоматизацию процесса проектирования в целом, а не только его отдельных частей, то сложную структуру процесса проектирования необходимо отразить в структуре самой САПР. Другими словами, САПР должна обеспечить возможность моделирования самого процесса проектирования, а полученная информационная модель может быть использована для организации и оптимизации управления этим процессом.

1.1.4. Методологии проектирования

В концептуальной модели процесса проектирования должны быть учтены особенности существующих методологий проектирования. С системной точки зрения процесс проектирования можно представить как последовательность преобразований (трансформаций) промежуточных описаний объекта проектирования, связывающих концептуальное описание объекта и его реализацию [2]. Эти преобразования выполняют проектные процедуры (Т), каждая из которых генерирует промежуточные проектные описания для следующего уровня на основе описаний предыдущего уровня (рис. 1.6).

В зависимости от направления выполнения этих преобразований различают два основных метода проектирования – восходящее и нисходящее проектирование, последовательности этапов для которых показаны на рис. 1.6 в виде правой и левой ветвей схемы соответственно.

При восходящем проектировании начинают с базовых функциональных элементов и затем, группируя их нужным образом, формируют узлы первого уровня, функционально более сложные, чем базовые элементы. В результате преобразования, повторяющего эту процедуру по отношению к элементам 1-го уровня, конструируют более сложные узлы, соответствующие 2-му уровню, и т.д. Требуемое описание объекта проектирования, т.е. проект, окончательно формируется в результате последнего из цепочки преобразований, которое и завершает процесс проектирования.

Недостатки, снижающие результативность методологии восходящего проектирования, обусловлены тем, что по мере усложнения проектируемых объектов происходит увеличение количества промежуточных уровней описания, а это, в свою очередь, увеличивает количество возможных проектных решений (экспоненциальная зависимость). Поэтому переход с i -го уровня проектирования на $(i+1)$ -й представляет собой не только задачу преобразования описания, но еще и комбинаторную задачу выбора вариантов. В случае проектирования сложного объекта трудно получить эффективные критерии выбора проектного решения на конкретном этапе, которые учитывали бы требования последующих этапов проектирования. В результате непосредственное применение метода восходящего проектирования приводит к трудно управляемой по-

следовательности преобразований, которая недостаточно эффективна по отношению к проектируемому объекту в целом.

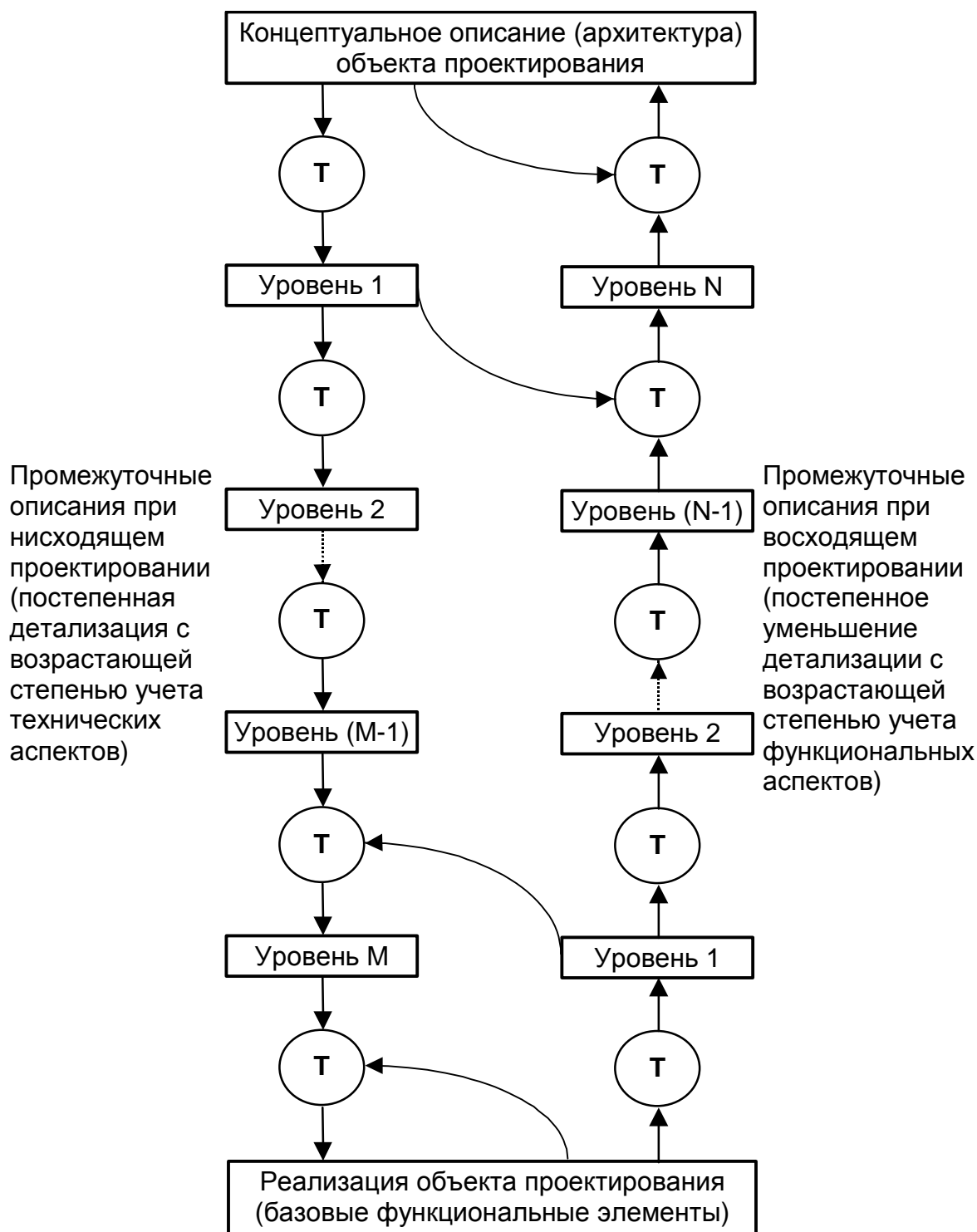


Рис. 1.6. Схематическое представление процесса проектирования

По ряду причин нисходящее проектирование является более предпочтительным методом, чем восходящее проектирование. Во-первых, часто разработка проекта начинается при отсутствии каких-либо определенных решений отно-

сительно базовых функциональных элементов, т.е. восходящее проектирование просто невозможно. В таких случаях используют нисходящее проектирование как единственно возможное, поскольку оно не связано с какой-либо конкретной технологией в течение максимально длительного периода времени.

Во-вторых, методология нисходящего проектирования лучше удовлетворяет проектным ограничениям, когда различные части системы должны разрабатываться в разное время различными группами проектировщиков, т.е. она хорошо согласуется с широко используемой методологией иерархического проектирования [3].

В-третьих, в настоящее время наибольшее распространение получили САПР, обеспечивающие сквозной цикл проектирования с возможностью многоуровневого моделирования, в основе которого лежит описание объекта на языках *HDL*, *HHDL*, *VHDL* и им подобных [11–15]. Базовой методологией проектирования в подобных САПР является нисходящее проектирование.

Основной недостаток у нисходящего проектирования тот же, что и у восходящего проектирования. Декомпозиция больших узлов на меньшие с учетом технических ограничений также представляет собой комбинаторную задачу экспоненциальной сложности, что приводит к необходимости выбора приемлемых вариантов из большой совокупности проектных решений, т.е. к необходимости отыскания эффективных критериев выбора.

В силу перечисленных выше причин на практике редко применяют в чистом виде какой-либо один из двух методов проектирования. Обычно попеременно применяют оба метода либо используют информацию, относящуюся к альтернативному методу с тем, чтобы корректно и эффективно управлять процессом выбора вариантов (на рис. 1.6 это показано стрелками, соединяющими в горизонтальном направлении различные последовательности этапов проектирования).

1.1.5. Конвейер проектирования

С системной точки зрения процесс проектирования представляет собой последовательность преобразований описания объекта проектирования. Например, процесс проектирования СБИС, выполняемый по методологии нисходящего проектирования, можно представить как последовательность преобразований

описательных уровней схемы (рис. 1.7). Каждый уровень этой последовательности характеризуется некоторыми средствами описания – языками, описательными элементами и так далее [16].



Рис. 1.7. Последовательность преобразований описаний СБИС

Декомпозиция каждого описательного уровня схемы, обусловленная невозможностью обработки проекта на данном уровне в целом, приводит к совокупности объектов обработки (фрагментов проекта на данном уровне). Из теории систем массового обслуживания (СМО) известно, что если обработке подлежит массовая совокупность объектов, причем обработка каждого из них состоит из нескольких последовательных этапов и эти этапы обеспечены разными исполнителями, целесообразно организовать конвейерную обработку объектов [17].

Работа конвейера может быть организована в синхронном или асинхронном режимах. В первом случае конвейер можно представить в виде последовательности независимых этапов, в начале каждого из которых, кроме первого, установлен специальный синхронизирующий элемент. Во втором случае ис-

пользуется специальное хранилище (накопитель) для объектов, прошедших один этап обработки и ожидающих начала следующего этапа. В [18] для хранения результатов обработки каждого этапа информационного конвейера предлагается использовать свой накопитель.

Таким образом, процесс проектирования – независимо от используемой методологии – является многоэтапным, т.е. его можно представить в виде последовательности преобразований промежуточных описаний объекта проектирования, выполняемых проектными процедурами. Этап проектирования – условно выделенная часть этой последовательности, состоящая из одной или нескольких проектных процедур. Основным понятием процесса проектирования является проектная процедура, представляющая формализованную совокупность действий, направленных на получение проектного решения. Последовательность проектных процедур или этапов проектирования, приводящая к получению проектного решения, образует так называемый маршрут проектирования. Применительно к автоматизированному проектированию вводят понятие технологического маршрута проектирования, представляющего любой согласованный между собой набор программ из пакетов прикладных программ (ППП) для различных этапов, обеспечивающий выполнение полного цикла проектирования конкретного объекта [19–21].

В САПР обработка каждого проекта организуется по принципу информационного конвейера, обрабатываемыми элементами которого являются прикладные программы, образующие технологический маршрут проектирования, а обрабатываемыми объектами – фрагменты, полученные в результате декомпозиции проекта.

В работе [22] идея информационного конвейера проектирования получила дальнейшее развитие. Для модельного описания процесса сквозного проектирования в интегрированной САПР введено понятие последовательно-параллельного конвейера проектирования, составленного из набора независимых проектных процедур. Рабочие места этого конвейера, представленные автоматизированными рабочими местами (АРМ) и интерактивно-графическими системами (ИГС), обеспечивают выполнение проблемно-ориентированных проектных процедур, результаты работы которых в виде проектных решений, помещенных в информационный контейнер, передаются в оперативное хранили-

ще контейнеров. Из хранилища контейнер можно запросить на следующее рабочее место конвейера в соответствии с технологическим маршрутом обработки проектной информации.

Работа последовательно-параллельного конвейера проектирования осуществляется в асинхронном режиме благодаря наличию хранилища контейнеров с проектными решениями. Это позволяет объединить в одном технологическом маршруте проектные процедуры различной производительности и обеспечить максимальную загрузку конвейера. Асинхронный режим работы конвейера обусловлен, главным образом, стохастическим характером временных затрат, требующихся каждой проектной процедуре на выработку проектного решения.

1.1.6. Организация параллельного проектирования

Эффективное использование САПР предполагает не только использование иерархического проектирования и конвейерной обработки проекта, но и возможность выполнения параллельного проектирования [23–26]. В современных САПР сквозного проектирования можно выделить три уровня параллельности при выполнении проектных работ, а именно:

одновременное выполнение нескольких различных проектов;

одновременное выполнение ряда проектных процедур одного и того же проекта;

параллельная обработка различных частей одного и того же проекта.

В наиболее полной мере задача организации параллельных работ может быть решена в рамках интегрированной САПР. Однако при этом следует учитывать специфические факторы, определяющие состояние и функционирование конкретных САПР [25]. К числу таких факторов относятся:

наличие разнородной техники, составляющей информационно-вычислительную инфраструктуру интегрированных САПР;

наличие программных средств проектирования, которые разрабатывались в разное время и, как правило, вне рамок единой концепции;

наличие разнородных баз данных, используемых проектными процедурами (в данном случае – проектирующими подсистемами);

высокая информационная сложность объекта проектирования, не позволяющая обрабатывать объект целиком (возникает необходимость декомпозиции объекта с последующим “сшиванием” его частей);

наличие локальных и глобальных критериев оценки качества проектных решений, используемых проектными процедурами;

специализация технических и программных средств, направленная на выполнение проектных процедур разных видов;

выполнение ряда работ в условиях частичной неопределенности в отношении корректности и достаточности исходной информации для отдельно взятой проектной процедуры;

частое отсутствие однозначных формализованных стратегий проектирования, гарантирующих получение заданного результата;

использование эвристических алгоритмов для получения приближенных решений;

наличие большого числа интерактивных процедур (имеется в виду вмешательство человека в автоматизированный процесс проектирования) в силу невозможности получения приемлемого проектного решения иным способом;

наличие большого количества ошибок в проектных решениях, связанных с использованием интерактивных процедур.

Перечисленные выше факторы предъявляют особые требования к организации процесса проектирования. При больших объемах информации и заданных жестких сроках выполнения проектных работ становится целесообразным начинать последующие (по технологическому маршруту) проектные процедуры с момента достижения корректных (с определенной вероятностью) результатов предыдущих проектных процедур. В большинстве случаев выявляемые на более поздних этапах проектирования ошибки в локальных проектных решениях приводят к необходимости внесения локальных изменений в ранее полученное проектное решение. Если в ходе работы над проектом выясняется необходимость внесения глобальных изменений в проект, то переработке подлежит значительная часть проектных решений. Но эта необходимость, как правило, была неочевидна на начальных этапах проектирования и, следовательно, для ее выявления все равно потребовались бы затраты, сопоставимые с теми, которые привели к обнаружению данной ошибки указанным путем.

По оценкам специалистов на устранение ошибок проектирования приходится в среднем 20% стоимости проекта и расходуется более 30% времени, затраченного на разработку проекта [27]. Поэтому любая технология, позволяющая существенно уменьшить количество ошибок, имеет исключительно важное значение. Одним из способов решения данной проблемы является организация сквозного проектирования в интегрированной САПР, другим – создание инфраструктур САПР, обеспечивающих интеграцию инструментальных средств проектирования и данных с управлением при помощи унифицированного интерфейса пользователя [28, 29].

1.1.7. Концепция сквозного проектирования в интегрированной САПР

В основу построения концепции сквозного проектирования в интегрированной САПР положена следующая интерпретация понятия “сквозное проектирование” с системной точки зрения:

- обеспечение модульной структуры системы с четкой регламентацией внешних спецификаций модулей;

- обеспечение унифицированного системного интерфейса, не зависящего от вида информации, вырабатываемой модулями системы;

- организация контроля и управления проектами на уровне проектных решений, вырабатываемых модулями системы;

- отчуждение проектных решений от их авторов;

- иерархическая структура объекта проектирования, представленная в виде графа вложений фрагментов объекта проектирования (с соотносением узлам графа смысловой информации, характеризующей данный фрагмент);

- идентификация проектных процедур и проектных решений;

- наличие в системе только двух типов модулей: управляющих (УМ) и обрабатывающих (ОМ);

- организация межмодульных обменов только по схеме: УМ ↔ ОМ;

- высокая степень автономности модулей;

- слабое взаимодействие между модулями системы (ненагруженный трафик передачи больших информационных массивов).

На основе выбранных системных критериев можно представить понятийную модель системы сквозного проектирования в виде асинхронного последовательно-параллельного конвейера с унифицированным системным интерфейсом контейнерного типа, не зависящим ни от конкретного вида проектной информации, ни от физической среды передачи информации.

Согласно этой модели, в качестве конвейера рассматривается мониторинговая система, а в качестве рабочих мест конвейера – обрабатывающие подсистемы. К понятию обрабатывающих подсистем отнесены все проектирующие и обслуживающие подсистемы.

Предложенная понятийная модель позволяет построить интегрированную САПР сквозного проектирования. Интеграция проектных решений обеспечивается за счет конвертирования данных, полученных в контейнере от предыдущей проектной процедуры. Конвертирование данных осуществляет последующая по маршруту проектная процедура.

1.1.8. Системные среды (инфраструктуры) САПР

В течение последних 10–15 лет за рубежом наблюдается повышенная маркетинговая активность в области так называемых системных сред, или инфраструктур САПР (*CAD Framework*). Десятки компаний, среди которых как малоизвестные, так и всемирно признанные фирмы (например, *Digital Equipment Corp.*, *Mentor Graphics Corp.*, *Cadence Design Systems Inc.* и другие), предлагают инфраструктуры собственной разработки в качестве единого решения проблемы интеграции различных инструментальных программных средств проектирования. В США в 1988 году была создана некоммерческая организация – консорциум *CAD Framework Initiative (CFI)*, работающий над созданием и внедрением стандартов на инфраструктуры САПР и объединяющий около 40 фирм-поставщиков средств САПР [30]. В Западной Европе действует родственная организация – консорциум *EuroCFI* [31].

Основными направлениями исследований, проводимых в настоящее время под эгидой *CFI*, являются [31]:

разработка архитектуры системной среды – анализ взаимозависимостей и взаимосвязей функциональных частей инфраструктуры;

создание подходов и средств представления проекта – разработка информационной модели для описания проекта и определение спецификаций программного интерфейса для доступа к элементам проекта;

исследование механизмов управления проектными данными – обеспечение защиты данных, создание минимального процедурного интерфейса с версиями, поддержание конфигурации проекта;

выработка подходов к управлению методологией проектирования – разработка средств метаописания прикладных программных модулей и проектных задач.

Ключевыми элементами любой инфраструктуры САПР являются интерфейс пользователя, методы доступа к базам данных и средства организации взаимодействия между прикладными программами. В настоящее время существуют два основных типа инфраструктур: инфраструктуры, ориентированные на управление процессом проектирования, и инфраструктуры, ориентированные на управление проектом [27].

Инфраструктуры первого типа, как правило, являются специализированными и обеспечивают выполнение тесно связанных задач в среде интегрированной САПР. Поскольку эти инфраструктуры рассчитаны на работу с такими специфическими для проектируемых изделий объектами, как описания выводов, вентиляей, геометрии и т.п. (см. выше рис. 1.7), для тесной интеграции входящих в них наборов инструментальных средств обычно требуется модификация их исходного кода. Хотя инфраструктуры этого вида упрощают преобразование и передачу проектных данных между различными прикладными программами, они, как правило, не рассчитаны на сбор данных для управления процессом проектирования в целом с целью достижения максимальной эффективности работы.

Инфраструктуры второго типа ориентированы на управление данными проектирования и предоставляют возможность для руководителя проекта получать информацию, необходимую для эффективного управления проектом. С точки зрения руководителя проекта эффективная инфраструктура должна обладать более широкими функциями, чем предоставление удобного пользовательского интерфейса, запуск необходимой прикладной программы проектирования и автоматическая подача на ее вход требуемых файлов данных. Руководителю

проекта приходится решать проблему проектирования изделия в целом, а не относительно узкие проблемы отдельных разработчиков. В большинстве случаев в совместной работе над проектом заняты специалисты по нескольким дисциплинам. При этом для каждой дисциплины (механики, электроники и программного обеспечения) имеется собственный набор инструментальных средств и ряд проблем процесса проектирования. Инфраструктура должна управлять проектными данными, формируемыми различными инструментальными программными средствами на различных инструментальных машинах при достаточно произвольном их территориальном (или даже географическом) размещении. Кроме того, инфраструктура должна работать с данными, необходимыми для управления версиями проекта и обеспечения его трассируемости.

По мнению экспертов *CFI* для эффективной работы необходима инфраструктура более высокого уровня, чем представленные выше типы инфраструктур САПР. Эта инфраструктура, называемая комплексной средой проектирования с автоматизированным управлением, должна обеспечивать когерентность и стабильность среды проектирования в целом, а также поддерживаемого ею процесса проектирования. Комплексная среда проектирования должна сбалансированной, т.е. в равной степени учитывать потребности разработчика в отношении гибкости проектирования и требования руководства к эффективности управления. Кроме того, она должна представлять собой открытую систему, предусматривающую возможность совместной работы инструментальных средств проектирования различных производителей.

В качестве примера развитой инфраструктуры САПР, удовлетворяющей целому ряду целевых требований *CFI*, можно привести программный комплекс *PowerFrame*, разработанный фирмой *Digital Equipment Corp. (DEC)* [27]. В частности, инфраструктура *PowerFrame* реализует возможности параллельного проектирования, поскольку содержит автоматизированные средства для координации работ и продвижения разрабатываемого проекта при участии специалистов многих смежных дисциплин.

Программный комплекс *PowerFrame* поддерживает разнородную сеть АРМ, которая может быть представлена компьютерами различных производителей и архитектур: компьютеры семейства *VAX* и компьютеры с *RISC*-архитектурой, производимые фирмой *DEC* и функционирующие под управле-

нием операционных систем (ОС) *VMS* и *ULTRIX* соответственно, рабочие станции *Sun*, *SPARCStation* и *SPARCServer* фирмы *Sun Microsystems Inc.* и *Apollo* фирмы *Hewlett-Packard Corp.*, управляемые ОС *UNIX*. Для взаимодействия представленных в сети АРМ, управляемых ОС *UNIX*, используется сетевой протокол *TCP/IP*.

В заключение следует отметить следующее. Хотя за последние десять лет в научно-технических периодических изданиях довольно часто появлялись заметки и статьи по тематике инфраструктур САПР, все они имели либо анонсный, либо обзорный характер, описывая функциональные возможности инфраструктур без раскрытия концептуальных основ их построения. Концептуальные основы построения инфраструктур САПР представляют важнейший элемент “ноу-хау” (*know how*) фирм, разрабатывающих и поставляющих подобные инструментальные средства интеграции на рынок САПР. Именно этим обстоятельством, по нашему мнению, объясняется закрытый характер информации.

1.2. Управление проектом и процессом проектирования в САПР

Анализ проблемы управления проектом и процессом проектирования в САПР приводит к следующим основным задачам: разработка структуры проекта, планирование выполнения проектных работ, настройка и оперативная перенастройка технологического маршрута проектирования, обеспечение взаимодействия подсистем и координация действий различных групп проектировщиков, обеспечение и контроль логической целостности проекта, мониторинг состояния проекта, учет сделанных в проекте изменений, контроль выполнения календарного плана работ, контроль и обеспечение достоверности информации при хранении и передаче проектных решений между подсистемами и другие.

1.2.1. Общие понятия и задачи управления

Управление – это информационный процесс, заключающийся в сборе информации о состоянии объекта управления, передаче ее в пункты накопления и переработки, анализе поступающей, накопленной и справочной информации,

принятии решения на основе выполненного анализа, выработке соответствующего управляющего воздействия и доведении его до объекта управления.

Анализ управления позволяет выделить тройку элементов, называемых *средой*, *объектом* и *субъектом управления* соответственно [32, 33]. Объект управления – выделенная часть среды, на которую воздействует субъект управления для реализации поставленных целей. При этом субъект ощущает на себе воздействия среды X и объекта Y . Состояние среды он изменить не может, но может управлять состоянием объекта путем специально организованного воздействия U , которое и является управлением (рис. 1.8).

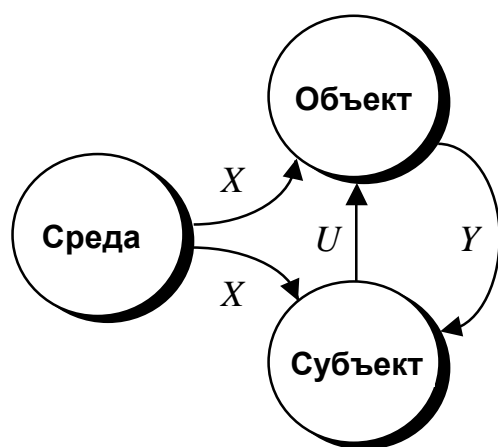


Рис. 1.8. Кибернетическая модель процесса управления

Состояние объекта управления влияет на состояние информационных потребностей субъекта управления. Обозначим информационные потребности субъекта как $A = (a_1, \dots, a_k)$, где a_i – состояние i -й потребности, которое можно выразить неотрицательным числом, характеризующим актуальность этой потребности. Свое поведение субъект управления строит так, чтобы минимизировать насущность своих ин-

формационных потребностей, т.е. решает задачу многокритериальной оптимизации [33]:

$$a_i(X, U) \rightarrow \min_{r \in R}, \quad (i = 1, 2, \dots, k), \quad (1.1)$$

где R – наличные ресурсы субъекта. Эта зависимость выражает связь информационных потребностей с состоянием X среды и поведением U субъекта управления.

Способ решения задачи (1.1), позволяющий определить оптимальное поведение U^*_X субъекта, называется алгоритмом управления, т.е. можно записать:

$$U^*_X = j(A_t, X),$$

(1.2)

где j – алгоритм, позволяющий синтезировать управление по состоянию среды X и информационных потребностей субъекта A_t . В общем случае информационные потребности субъекта не являются постоянными, т.е. они изменяются со

временем (это отмечается индексом t) в результате влияния среды и объекта – X и Y , соответственно, а также самостоятельно, отражая процессы развития субъекта.

Используемый алгоритм управления j определяет эффективность функционирования субъекта в данной среде. Он имеет рекуррентный характер:

$$U_{N+1} = j(U_N, A_t, X), \quad (1.3)$$

т.е. позволяет на каждом шаге улучшать управление.

Для удовлетворения своих информационных потребностей субъект декомпозирует алгоритм управления j , вводя промежуточную стадию – формулирование цели управления Z^* , т.е. действует по следующей двухэтапной схеме:

$$A_t \rightarrow Z^* \rightarrow U^*.$$

На первом этапе определяется цель управления Z^* , исходя из имеющихся информационных потребностей A_t :

$$Z^* = j_1(X, A_t), \quad (1.4)$$

где j_1 – алгоритм синтеза цели Z^* по информационным потребностям A_t и состоянию среды X .

На втором этапе синтезируется управление U^*_X , реализация которого обеспечивает достижение цели Z^* , сформированной на первой стадии, т.е. решается следующая задача:

$$U^*_X = j_2(Z^*, X), \quad (1.5)$$

где j_2 – алгоритм управления.

Разделение процесса управления на два этапа обуславливает выполнение функций j_1 и j_2 разными структурными элементами – субъектом управления и устройством управления (УУ) соответственно. Устройство управления и объект управления образуют *систему управления* (СУ), выполняющую функцию реализации целей управления Z^* , формируемых субъектом управления.

Цели, принципы и границы управления зависят от сущности решаемой задачи. Как правило, управление сложной системой состоит из нескольких этапов, которые включают: 1) формулирование целей управления; 2) определение объекта управления; 3) структурный синтез модели; 4) идентификация параметров модели объекта; 5) планирование эксперимента; 6) синтез управления; 7) реализация управления; 8) адаптация. Не все из перечисленных восьми этапов

присутствуют при синтезе СУ, ряд из них может быть опущен, исходя из конкретных условий решения задачи управления.

Ниже рассмотрены особенности организации управления процессом сквозного проектирования в интегрированной САПР.

1.2.2. Задачи управления процессом проектирования

Если объектом управления является процесс проектирования, то на его входе и выходе фиксируются не материальные потоки, как в случае процессов, протекающих в технических системах [33], а информационные (рис. 1.9). Информация о событиях процесса проектирования фиксируется в информационной модели, являющейся важнейшим компонентом системы управления. Эта информация анализируется субъектом управления с целью оценки текущего состояния процесса проектирования и выработки управляющих воздействий, оптимизирующих выполнение этого процесса.

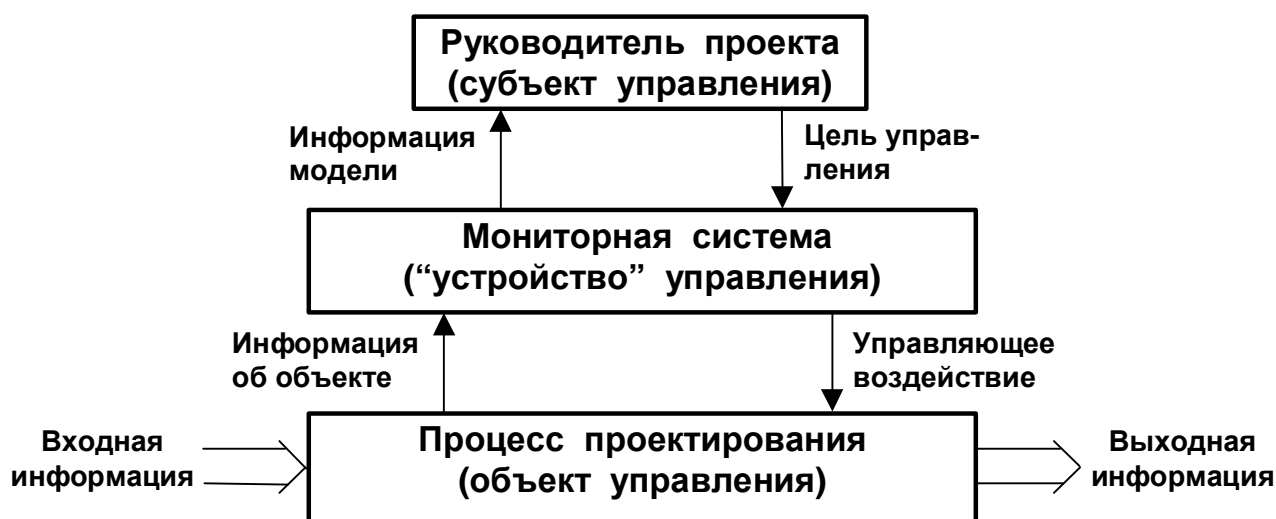


Рис. 1.9. Управление проектированием как информационный процесс

В общем случае процесс проектирования характеризуется наличием большого числа плохо формализуемых факторов, которые не позволяют организовать управление этим процессом в соответствии с принципами, разработанными для автоматических систем. В САПР в контур управления процессом проектирования включен человек, деятельность которого трудно формализо-

вать, а само управление, как правило, выполняется в автоинтерактивном режиме.

При рассмотрении процесса проектирования в качестве объекта управления необходимо решить ряд важных задач, среди которых следует выделить такие, как формулировка целей управления, определение (выделение) объекта управления, структурный синтез модели объекта управления, синтез системы управления. Ниже каждая из перечисленных задач рассмотрена более подробно.

Формулировка целей управления. Большинство САПР, используемых в настоящее время, построены путем комплексирования ранее разработанных, апробированных и достаточно эффективных программных инструментальных средств. Отличительной чертой подобных систем является то, что в них по-прежнему значительна роль человека в управлении проектными процедурами и в организации взаимодействия между ними.

С точки зрения управления процесс проектирования определяется планами, графиками, другими документами и контролируется по представлению отчетной документации ответственными исполнителями проектных процедур. По окончании каждой проектной процедуры или определенного этапа проектирования их ответственные исполнители передают выходную информацию в следующие проектные процедуры или на следующие этапы проектирования.

Подобная схема управления проектированием имеет ряд слабых мест, к которым относятся:

- значительные трудности при составлении координационных планов, сетевых графиков, увязывающих во времени взаимосвязанные этапы проектных работ;

- отсутствие средств контроля логической целостности проекта в целом и его составных частей;

- отсутствие оперативной информации о ходе выполнения работ над проектом в целом и каждой из его составных частей в отдельности;

- невозможность оперативного вмешательства в процесс управления проектированием руководителя проекта и/или его ответственных исполнителей;

- большая вероятность потери информации в процессе передачи ее между проектными процедурами и/или этапами проектирования;

возможность искажения результирующей информации (проектных решений) в результате отказов или сбоев оборудования;

возможность искажения информации или ее полной потери (замены) в результате запрещенных действий лиц, участвующих в выполнении процесса проектирования;

возможность доступа к проектной информации лиц, не имеющих на это соответствующего разрешения (несанкционированный доступ), поскольку защита информации часто осуществляется только путем реализации организационных мероприятий.

Перечисленные выше недостатки организации процесса управления проектированием и взаимодействия различных обрабатывающих подсистем значительно снижают научно-технический уровень проработки проекта и усложняют задачу получения конечного проектного решения. Они и определяют цели управления, а также требуемые процедуры и операции управления, которые возможно и целесообразно автоматизировать путем создания мониторинной системы (МС) САПР.

Поскольку общие требования, предъявляемые к процессу проектирования, не зависят от конкретной предметной области проектирования, разрабатываемые базовые компоненты МС должны быть инвариантными составляющими САПР различных типов.

Определение объекта управления. Очевидно, что рассмотрение процесса проектирования в виде некоторого “монолитного” объекта (см. выше рис. 1.9) или в виде иерархии подпроцессов проектирования (см. выше рис. 1.3), не позволяет четко сформулировать основные требования к системе управления этим объектом. Поэтому необходимо продолжить уточнение процесса проектирования и, в частности, выполнить его структурирование с целью выделения в нем базового, т.е. минимального объекта управления (с точки зрения субъекта управления – руководителя проекта). Это тем более важно, что свойства и характеристики объекта управления во многом определяют состав, структуру и функциональные возможности мониторинной системы.

Объектом управления для мониторинной системы является проектная процедура, поскольку только она позволяет получить промежуточный результат проектирования или проектное решение [34, 35]. Проектное решение, получае-

мое в результате выполнения проектной процедуры, может быть отнесено к одному из двух видов: промежуточное и окончательное. Отнесение проектного решения к тому или иному виду может быть выполнено как на этапе данной проектной процедуры, так и на более поздних (по технологическому маршруту) этапах проектирования. Вследствие этого перевод сформированного проектного решения в результат проектирования должен инициироваться с более высокого уровня.

Для целей формализации процесса проектирования, выполняющегося в рамках интегрированной среды обработки, ниже введены два вспомогательных понятия – комплексная проектная процедура и комплексное проектное решение.

В общем случае в состав интегрированной среды обработки могут входить различные инструментальные средства – от отдельных программ, реализующих какую-либо одну проектную процедуру или даже одну проектную операцию (например, ввод описания схемы на некотором языке), до функционально полных (интегрированных) пакетов прикладных программ (ППП), реализующих ряд связанных проектных процедур, т.е. автоматизирующих выполнение целого этапа проектирования (например, функционально-логического).

С точки зрения управления и проектная процедура, реализуемая отдельной программой, и ряд проектных процедур, реализуемых ППП, рассматриваются как отдельная *комплексная проектная процедура* (КПП). В последнем случае это обусловлено тем, что иногда модификацию монитора (управляющего модуля) пакета достаточно сложно выполнить (например, при отсутствии исходных текстов программ и использовании изоэренной логики управления). Кроме того, часто проектные процедуры, выполняющиеся в рамках того или иного ППП, сконфигурированы на достижение оптимальных характеристик, что ставит под сомнение целесообразность введения “внешнего” управления для этих процедур, которое может только ухудшить эти характеристики.

Проектное решение, сформированное КПП, рассматривается как *комплексное проектное решение* (КПР). В отличие от обычного проектного решения, представленного совокупностью файлов того или иного вида, КПР оформляется в виде информационного контейнера, содержащего как проектную, так и сопровождающую информацию, и передается в оперативное хранилище кон-

тейнеров информационной системы (ИС) интегрированной САПР [22, 25]. Если КПП представлена несколькими проектными процедурами, выполняющимися в рамках некоторого ППП, то в пределах этого пакета передача и хранение проектной информации осуществляется способом, определяемым методикой эксплуатации этого пакета. Вне пределов ППП способ передачи проектной информации определяется исходя из требований среды проектирования (и в частности – мониторинной системы САПР).

Таким образом, с точки зрения управления проектом и процессом проектирования в интегрированной САПР минимальным объектом управления является комплексная проектная процедура (КПП). Результатом работы КПП является комплексное проектное решение (КПР), оформленное в виде информационного контейнера и предназначенное для передачи в оперативное хранилище контейнеров информационной системы САПР. Введенные понятия расширяют стандартные понятия “проектная процедура” и “проектное решение” в части описательной информации, необходимой для их семантической идентификации.

1.2.3. Формализация процесса проектирования

Технологически процесс проектирования может быть представлен в виде набора комплексных проектных процедур, связанных определенным образом. Математическим представлением процесса проектирования является ориентированный граф (орграф) $G(V, D)$, вершины V которого ассоциированы с комплексными проектными процедурами технологического маршрута, а дуги D – с информационными связями между этими процедурами [24–26, 36].

Последовательно-параллельный конвейер проектирования.

В технологическом маршруте проектирования могут содержаться как информационно-зависимые комплексные проектные процедуры, допускающие только последовательное выполнение, так и информационно-независимые, допускающие при наличии необходимых ресурсов параллельное выполнение. Другими словами, конвейерная схема обработки проекта, реализующая процесс проектирования в САПР, может иметь как последовательные, так и параллельные участки выполнения. Таким образом, схематично гипотетический последовательно-параллельный конвейер обработки данных, состоящий из десяти рабочих

мест и включающий два параллельных участка, может быть изображен в виде орграфа, как показано на рис. 1.10. Параллельные участки конвейера содержат комплексные проектные процедуры (ПП₃, ПП₄ и ПП₆, ПП₇, ПП₈), допускающие одновременное (параллельное) выполнение, т.е. информационно независимые комплексные проектные процедуры.

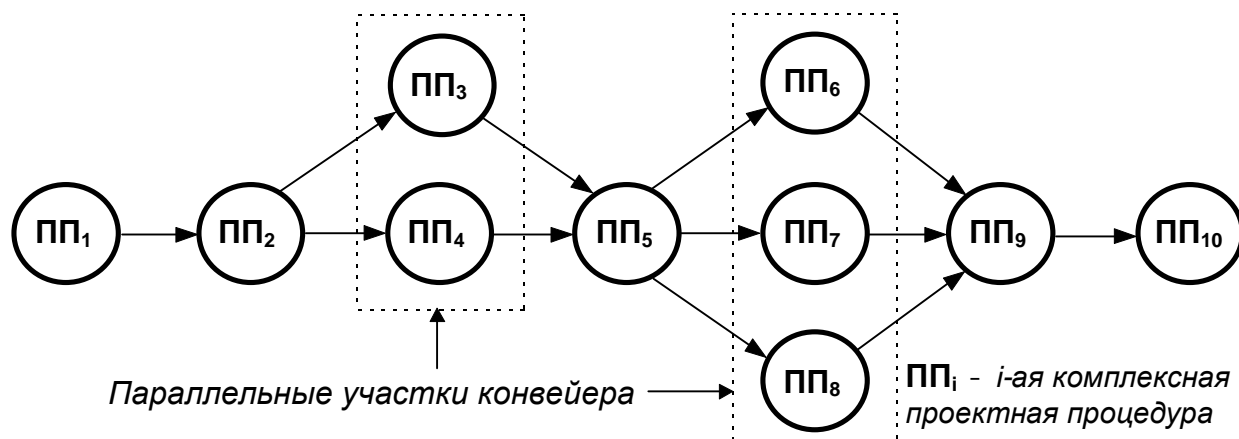


Рис. 1.10. Гипотетический последовательно-параллельный конвейер обработки данных

Откаты в технологическом маршруте проектирования. Как было показано выше, одним из базовых принципов проектирования является итерационный принцип, суть которого состоит в том, что проектные процедуры выполняются многократно до тех пор, пока не будут достигнуты приемлемые с точки зрения ТЗ результаты. Итерации могут выполняться как в пределах одного этапа проектирования, так и между этапами. Проектирование сложных технических объектов сопряжено с выполнением многочисленных “откатов” к ранее выполненным проектным процедурам в технологическом маршруте. Это связано с тем, что ошибка в проектном решении, сформированном проектной процедурой, может быть обнаружена не сразу, а лишь после того, как это проектное решение использовано в последующих процедурах маршрута.

Возврат управления к ранее выполненным КПП для доработки или полной переработки сформированных ими КПП называется *откатом* в технологическом маршруте проектирования. В общем случае количество откатов и их “глубина” определяются как объективными, так и субъективными факторами процесса проектирования. В качестве условной единицы измерения глубины отката можно использовать некоторую “эталонную” КПП (без учета существа и

продолжительности выполняемых ею проектных работ), т.е. фактически вершину в орграфе. Часто необходимо знать временные затраты на выполнение итерации, а не то, на сколько вершин в орграфе необходимо отступить, чтобы повторить последовательность проектных процедур. В этом случае глубину отката можно измерять в календарных днях, просуммировав продолжительность работ по каждой из проектных процедур, вовлекаемых в итерацию. При наличии параллельно выполняемых КПП в суммировании должна участвовать та КПП параллельного участка конвейера, которая имеет большую продолжительность работы.

Для каждой КПП технологического маршрута необходимо обеспечить возможность возврата управления на одну из предыдущих процедур маршрута. В результате множество дуг D в орграфе G можно условно разделить на три подмножества D_1 , D_2 и D_3 , т.е. $D = D_1 \cup D_2 \cup D_3$, где D_1 – прямые связи между смежными вершинами орграфа (см. выше рис. 1.10), D_2 – “ближние” обратные связи между смежными вершинами, D_3 – “дальние” обратные связи между вершинами орграфа, которые не являются смежными (с точки зрения прямых связей), но между которыми существует опосредствованная информационная зависимость.

Перегруженность орграфа $G(V, D)$ дугами ведет к существенному увеличению накладных расходов при его хранении и обработке. Например, только одна проектная процедура ПП₉ (см. выше рис. 1.10) имеет 8 обратных дуг, характеризующих:

“ближние” связи с процедурами ПП₆, ПП₇ и ПП₈;

“дальние” связи с процедурами ПП₁, ПП₂, ПП₃, ПП₄ и ПП₅.

Анализ информационных связей между КПП в технологическом маршруте проектирования с точки зрения автоматизации выполнения процедуры управления типа “глубокий откат” показывает, что дуги, соответствующие дальним обратным связям, можно исключить из рассмотрения без ущерба для общности решения задачи. Необходимость учета ближних обратных связей между вершинами орграфа можно объяснить возможностью обхода орграфа в двух направлениях – от КПП, в которой допущена ошибка проектирования, к КПП, в которой обнаружена эта ошибка, и противоположное этому. Если при решении задачи глубокого отката выбрать только одно направление (от начала

маршрута проектирования к его концу), то можно исключить из рассмотрения дуги, соответствующие ближним обратным связям КПП в маршруте проектирования. Более подробная информация о способах выполнения процедуры управления “глубокий откат” содержится в Главе 3.

Структурная модель проекта. В качестве объекта обработки последовательно-параллельного конвейера проектирования выступает проект, представляющий многоуровневое описание объекта проектирования (ОП). Каждая КПП конвейера отвечает за “проработку” одного из уровней в рамках того или иного аспекта проектирования. При проектировании сложных объектов используется блочно-иерархический подход, заключающийся в иерархической декомпозиции ОП на ряд фрагментов по функциональным признакам и обработке каждого из фрагментов по отдельности. Предельные количественные характеристики КПП (точнее – реализующих их АРМ) часто приводят к необходимости дальнейшего разбиения функциональных узлов объекта на фрагменты по количественным признакам. Таким образом, проект в стадии разработки представляет собой совокупность описаний фрагментов ОП в различных аспектах проектирования.

Математическим представлением информационной модели проекта является конечный гиперграф $H(X, E; R)$, где $X = \{x_1, x_2, \dots, x_n\}$ – множество вершин, ассоциированных с описаниями функциональных подсистем (узлов) ОП в различных аспектах, $E = \{e_1, e_2, \dots, e_m\}$ – множество ребер, представляющих отношения между вершинами, т.е. между частями проекта, $R = f(x, e)$ – инцидентор, определенный для $\forall x_i \in X$ и $\forall e_j \in E$ и принимающий значение 1 (истина), если x_i и e_j инцидентны, или 0 (ложь) – в противном случае.

Схематично гиперграф информационной модели проекта можно представить в виде совокупности “слоев” – информационных проекций или аспектов обрабатываемого объекта, пронизанных “вектором” обработки, как показано на рис. 1.11. Этот вектор отражает общее направление обработки проекта и может быть представлен пронумерованным перечнем аспектов обработки.

Иерархическая структура проекта, представленная в отдельном слое, является структурным представлением проекта в рамках одного аспекта обработки. В общем случае в пределах различных аспектов структура проекта может различаться (например, за счет учета количественных ограничений КПП, ответ-

ственных за проработку ОП в том или ином аспекте, степень структурной декомпозиции проекта может быть разной).

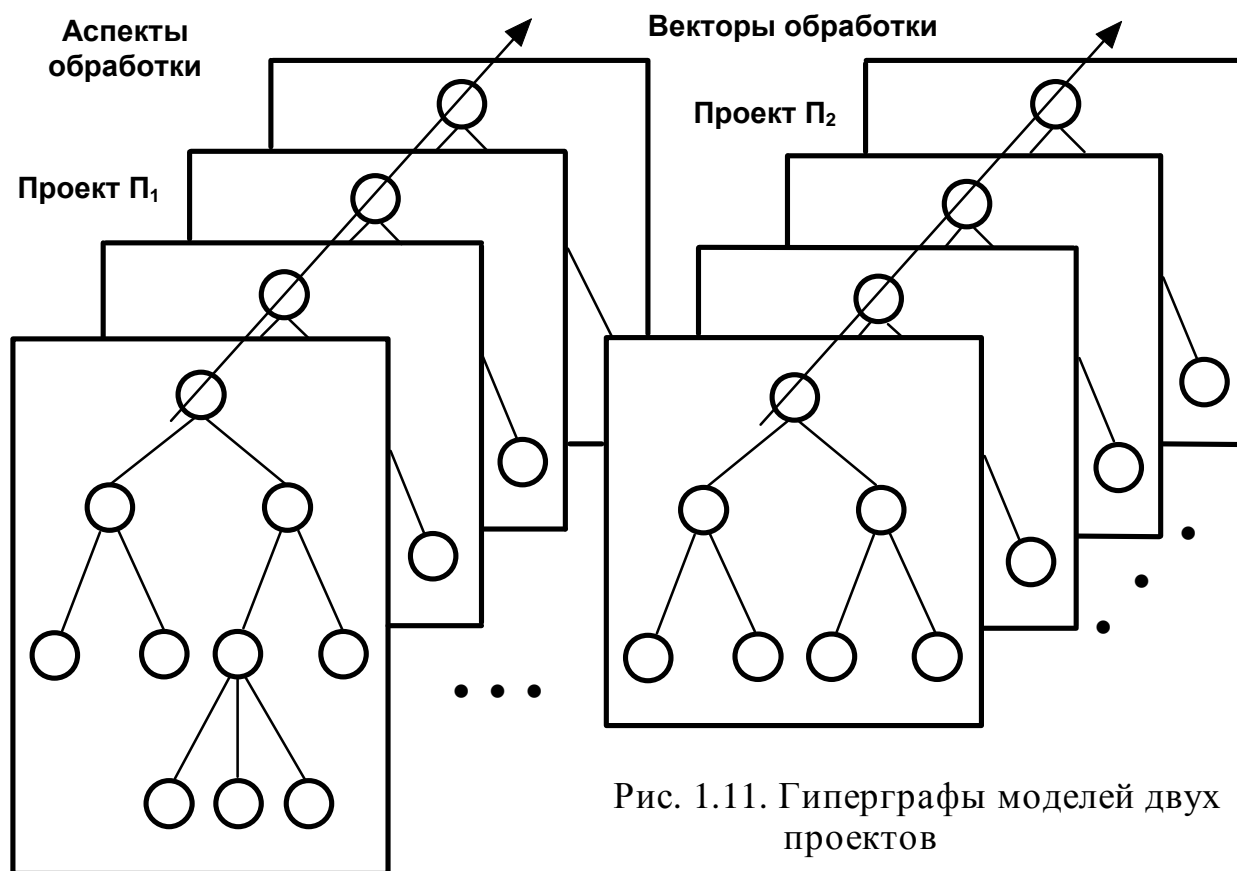


Рис. 1.11. Гиперграфы моделей двух проектов

1.3. Информационная модель процесса проектирования

Сложность проблемы моделирования процесса проектирования прежде всего обусловлена следующими факторами:

высокой структурной и функциональной сложностью объекта и процесса проектирования;

мультипроектным характером обработки, предполагающим одновременное выполнение в распределенной вычислительной среде нескольких проектов;

многоаспектным представлением обрабатываемых проектных данных;

динамическими свойствами распределенной среды обработки данных.

Один из возможных подходов к формализованному описанию процесса проектирования заключается в рассмотрении конвейера проектирования как системы массового обслуживания (СМО). Действительно, с точки зрения теории СМО [17], каждая проектная процедура конвейера может рассматриваться

как обрабатывающий элемент (ОЭ), а множество фрагментов проекта, т.е. описаний фрагментов объекта проектирования – как множество очередей заявок (ОЗ) на обслуживание к ОЭ (рис. 1.12). Длина очереди заявок может варьироваться как в пространстве, т.е. от одного ОЭ к другому (за счет декомпозиционных различий для каждой из проектных процедур), так и во времени, т.е. по мере того, как ОЭ удовлетворяет заявки из очереди.

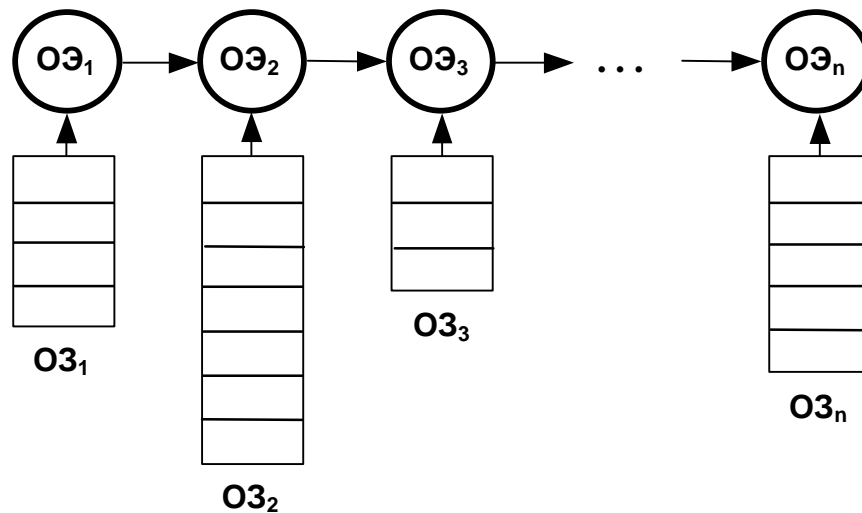


Рис. 1.12. Пример рассмотрения гипотетического конвейера проектирования как СМО

Проектирование в интегрированной САПР ведется, как правило, в мультипроектном режиме, т.е. параллельно в среде проектирования разрабатывается множество проектов. Таким образом, для каждого ОЭ формируется множество ОЗ различной длины, которая к тому же изменяется со временем – по мере того, как удовлетворяются заявки из очереди. Задача обработки множества очередей переменной длины, относящихся к разным проектам, достаточно сложна. Ее сложность многократно возрастает, если учесть ретрофитный характер процесса проектирования, обусловленный необходимостью повторной обработки фрагментов в случае обнаружения ошибок в ранее сформированных проектных решениях.

На наш взгляд, более простой подход к решению задачи создания информационной модели процесса проектирования состоит в следующем. Выполняется декомпозиция проекта и процесса проектирования, в результате которой создаются структурные модели проекта и технологического маршрута проекти-

рования, представляемые в виде гиперграфа H и орграфа G соответственно. Затем выполняется суперпозиция этих моделей, в результате которой образуется обобщенная структурная модель процесса проектирования S , представляющая собой ациклический орграф сложной структуры.

Индивидуальность информационной модели S для каждого проекта, разрабатываемого в САПР, обеспечивается за счет “поименной” привязки фрагментов ОП (и соответственно частей проекта) к КПП конвейера. Вследствие этого, количество вершин в S для разных проектов оказывается различным и прямо пропорционально как количеству КПП, так и количеству частей проекта (т.е. листьев дерева информационной модели проекта) для i -го аспекта проектирования.

С точки зрения идентификации комплексных проектных процедур, реализации по отношению к ним процедур управления и защиты проектной информации для каждой проектной процедуры должны быть назначены следующие атрибуты (внешние реквизиты, не зависящие от существа выполняемых процедурой действий):

идентификатор;

идентификатор проектируемого узла;

идентификатор типа проектного решения;

идентификатор ответственного исполнителя;

идентификатор проектирующего подразделения;

время функционирования:

1) плановый срок выполнения (даты начала и окончания);

2) фактический срок выполнения (даты начала и окончания);

код рабочего состояния;

входные ссылки (список идентификаторов КПП, которые формируют КПП, поступающие вход данной проектной процедуры).

Каждая комплексная проектная процедура информационной модели в любой момент процесса проектирования находится в одном из следующих рабочих состояний: пассивное, ожидания, выполнение, приостановленное, локально завершена. Сразу после генерации рабочей модели проектные процедуры находятся либо в состоянии “пассивное” (если они не имеют входных ссылок, т.е. являются начальными процедурами маршрута), либо в состоянии

“ожидания” (если им на вход не поступило проектное решение от предшествующей по маршруту проектной процедуры).

В дальнейшем важные – с точки зрения управления – события в системе могут привести к смене рабочего состояния комплексной проектной процедуры. Эти события (передача, прием, удаление или замена контейнера с проектным решением, откат в технологическом маршруте) контролируются средствами мониторинг системы, которые обновляют состояние информационной модели. С другой стороны, можно выполнить сканирование информационной модели, чтобы получить сведения о текущем состоянии проекта.

Смену рабочих состояний комплексной проектной процедурой можно представить с помощью диаграммы состояний (рис. 1.13).

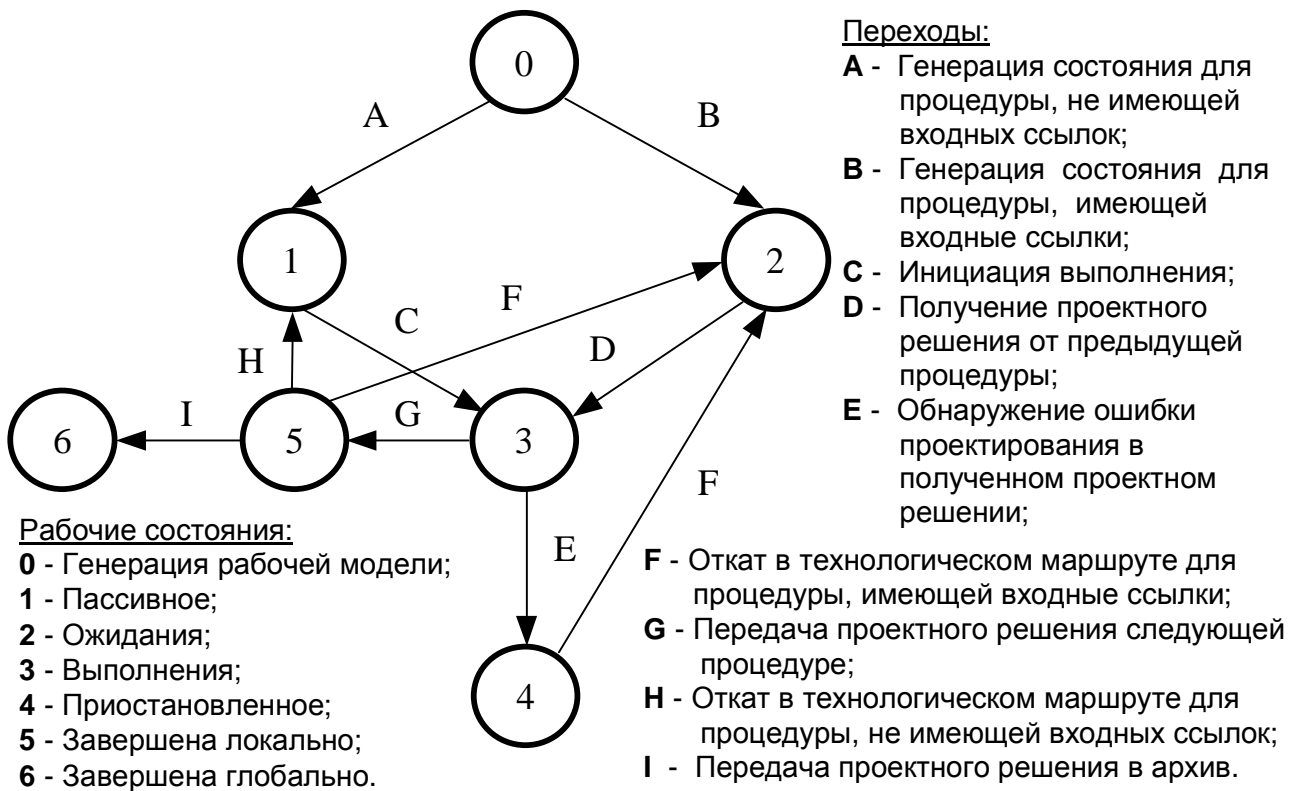


Рис. 1.13. Диаграмма состояний комплексной проектной процедуры

Для каждого проекта составляется временной график работ $T(P)$, определяющий календарные даты начала и окончания работ каждой КПП по каждой из частей проекта. Затем выполняется “наложение” графика T на соответствующую сетевую модель S . Полученная в результате обобщенная информационная модель $M(H, G, T)$ является основой для моделирования процесса проекти-

рования, которая может быть использована при решении задач управления проектированием. Для работы с этой моделью разработаны специальные способы и программные средства поддержки, описание которых приведено в следующих главах монографии.

Глава 2. Язык описания и программные средства поддержки информационной модели процесса проектирования

Информационная модель процесса проектирования, используемая для автоматизированного решения задач управления в САПР, может существовать в статической (исходное описание) и динамической (рабочая модель) формах.

Для создания исходного описания информационной модели процесса проектирования предложен специальный язык, который подобно большинству искусственных языков, ориентированных на машинную обработку, является языком непосредственных составляющих, более точно – контекстно-свободным языком. Практическая реализация этого языка представлена в мониторинной системе генератором рабочей модели – трехпроходным транслятором, выполняющим синтаксический и семантический анализ исходного описания, поданного на его вход, и автоматически формирующим файл рабочей модели процесса проектирования.

Рабочая модель процесса проектирования является динамической моделью, отражающей текущее состояние процесса проектирования. Важные – с точки зрения управления проектом – события процесса проектирования (передача, прием, удаление и замена контейнеров с проектными решениями, выполнение откатов в технологическом маршруте проектирования) контролируются мониторинной системой, автоматически модифицирующей рабочую модель. Кроме того, для ведения рабочей модели процесса проектирования, т.е. для поддержания ее в актуальном состоянии, в состав мониторинной системы включены сервисные программы, обеспечивающие выполнение интерактивной (оперативной) корректировки рабочей модели и ряд других вспомогательных функций.

В данной главе приведены описания синтаксиса и семантики операторов языка и даны примеры использования этих операторов, рассмотрены состав, структура и функциональные возможности генератора рабочей модели, а также назначение сервисных программ, манипулирующих с рабочей моделью процесса проектирования.

2.1. Язык описания информационной модели процесса проектирования

Язык описания информационной модели процесса проектирования (ЯОМ) – язык непосредственных составляющих, в основе которого лежит контекстно-свободная грамматика. Контекстно-свободная грамматика может быть представлена в нормальной форме Хомского [37, 38] как $G = (N, T, P, S)$, где N – конечное множество нетерминальных символов (нетерминалов) языка; T – конечное множество терминальных символов (терминалов) языка; P – конечное множество продукций (порождающих правил или правил подстановки) вида $A \rightarrow b$, где $A \in N$, $b \in (N \cup T)$; S – начальный символ (аксиома) грамматики, $S \in N$. Кроме того, дополнительным условием является $N \cap T = \emptyset$, т.е. ни один из символов языка не может быть одновременно терминалом и нетерминалом.

2.1.1. Синтаксическая нотация

Для описания синтаксиса контекстно-свободных языков часто используют формы Бэкуса-Наура (БНФ) [39]. В работах Н. Вирта этот формализм получил дальнейшее развитие и стал известен как расширенные формы Бэкуса-Наура (РБНФ) [40–42].

Описание синтаксиса языка с помощью РБНФ состоит из набора продукций, определяющих процесс образования конструкций языка. Набор таких продукций называется грамматикой языка [38]. Каждая продукция РБНФ имеет следующий вид:

$$S = E.$$

где S – некоторое синтаксическое понятие (нетерминальный символ) языка, а E – синтаксическое выражение, являющееся определением этого синтаксического понятия. Синтаксические понятия языка в РБНФ (так же как и в БНФ) представляются словами, смысл которых интуитивно понятен. Терминальные символы языка заключаются в кавычки. Множество метасимволов, использующихся в РБНФ, приведено в табл. 2.1.

Метасимволы РБНФ

Метасимвол	Смысловое значение
=	Определяется как
	Или (альтернатива)
.	Конец продукции
[X]	Нуль или одно вхождение X
{X}	Нуль или более вхождений X
(X Y)	Группирование: или X, или Y
“XYZ”	Терминальный символ XYZ
Метаимя	Нетерминальный символ с именем <i>Метаимя</i>

Метаязык РБНФ обладает достаточной “выразительной мощностью”, чтобы описать синтаксис любого контекстно-свободного языка. В качестве одного из подтверждений его “самодостаточности” можно привести описание собственного синтаксиса РБНФ с помощью продукций РБНФ [42]:

Синтаксис = { *Продукция* }.

Продукция = *НетерминальныйСимвол* “=” *Выражение* “.”.

Выражение = *Терм* { “|” *Терм* }.

Терм = *Множитель* { *Множитель* }.

Множитель = *ТерминальныйСимвол* | *НетерминальныйСимвол* |
 “(” *Выражение* “)” / “[” *Выражение* “]” / “{” *Выражение* “}”.

2.1.2. Синтаксис ЯОМ

Неформально структуру исходного описания информационной модели процесса проектирования можно представить в следующем виде [43]:

Заголовок_проекта

Описание_проектной_процедуры_1

Описание_проектной_процедуры_2

...

Описание_проектной_процедуры_N

где **Заголовок_проекта** определяется с помощью оператора **PROJECT** Идентификатор_проекта, а каждое **Описание_проектной_процедуры** можно представить как

Заголовок_проектной_процедуры

Тело_проектной_процедуры

Заголовок_проектной_процедуры определяется с помощью оператора **PROCEDURE** Идентификатор_проектной_процедуры, а **Тело_проектной_процедуры** содержит следующие операторы для описания реквизитов КПП:

```

NODE  Идентификатор_проектируемого_узла
TYPE  Идентификатор_типа_проектного_решения
USER  Идентификатор_ответственного_исполнителя
SUBDIV Идентификатор_проектирующего_подразделения
START Плановая_дата_начала_работ_для_данной_процедуры
FINISH Плановая_дата_окончания_работ_для_данной_процедуры
INPUT BEGIN
    Входная_ссылка_1
    Входная_ссылка_2
    . . .
    Входная_ссылка_N
END,

```

где элементы *Входная_ссылка_1*, *Входная_ссылка_2*, ..., *Входная_ссылка_N* представляют собой идентификаторы тех КПП, которые вырабатывают проектные решения, использующиеся данной КПП.

Формальное описание синтаксиса ЯОМ, выполненное с помощью РБНФ, представлено ниже.

Описание-информационной-модели-проекта = Заголовок-проекта

Описание-проектной-процедуры
{ Описание-проектной-процедуры }.

Заголовок-проекта = "PROJECT" Идентификатор-проекта.

Идентификатор-проекта = Идентификатор.

Описание-проектной-процедуры = Заголовок-проектной-процедуры

Тело-проектной-процедуры.

Заголовок-проектной-процедуры =

"PROCEDURE" Идентификатор-проектной-процедуры.

Тело-проектной-процедуры = Информация-о-проектируемом-узле

Информация-о-типе-проектного-решения

Информация-об-ответственном-исполнителе

Информация-о-проектирующем-подразделении

Информация-о-начале-проектных-работ

Информация-об-окончании-проектных-работ

Информация-о-входных-ссылках.

Информация-о-проектируемом-узле =

"NODE" Идентификатор-проектируемого-узла.

Идентификатор-проектируемого-узла = Идентификатор.

Информация-о-типе-проектного-решения =

"TYPE" Идентификатор-типа-проектного-решения.

Идентификатор-типа-проектного-решения = Идентификатор.

Информация-об-ответственном-исполнителе = "USER" Имя-исполнителя.

Имя-исполнителя = Идентификатор.

Информация-о-проектирующем-подразделении =

"SUBDIV" Название-подразделения.

Название-подразделения = Идентификатор.

Информация-о-начале-проектных-работ = "START" Плановая-дата-начала-работ.

Плановая-дата-начала-работ = Дата.

Информация-об-окончании-проектных-работ =

"FINISH" Плановая-дата-окончания-работ.

Плановая-дата-окончания-работ = Дата.

Дата = Число-месяца "-" Месяц "-" Год.

Число-месяца = Цифра [Цифра].

Месяц = "JAN" | "FEB" | "MAR" | "APR" | "MAY" | "JUN" | "JUL" | "AUG" | "SEP" | "OCT" | "NOV" | "DEC" | "ЯНВ" | "ФЕВ" | "МАР" | "АПР" | "МАЙ" | "ИЮН" | "ИЮЛ" | "АВГ" | "СЕН" | "ОКТ" | "НОЯ" | "ДЕК".

Год = Цифра Цифра Цифра Цифра.

Информация-о-входных-ссылках = "INPUT"

"BEGIN"

{ Идентификатор-проектной-процедуры }

"END".

Идентификатор-проектной-процедуры = Идентификатор.

Идентификатор = Буква { Буква | Цифра | Символ-подчеркивания |

Знак-денежной-единицы }.

Буква = Буква-латинского-алфавита | Буква-русского-алфавита.

Буква-латинского-алфавита = "A" | "B" | "C" | "D" | "E" | "F" | "G" | "H" | "I" | "J" | "K" | "L" | "M" | "N" | "O" | "P" | "R" | "S" | "T" | "U" | "V" | "W" | "X" | "Y" | "Z" | "a" | "b" | "c" | "d" | "e" | "f" | "g" | "h" | "i" | "j" | "k" | "l" | "m" | "n" | "o" | "p" | "r" | "s" | "t" | "u" | "v" | "w" | "x" | "y" | "z".

Буква-русского-алфавита = "А" | "Б" | "В" | "Г" | "Д" | "Е" | "Ж" | "З" | "И" | "Й" | "К" | "Л" | "М" | "Н" | "О" | "П" | "Р" | "С" | "Т" | "У" | "Ф" | "Х" | "Ц" | "Ч" | "Ш" | "Щ" | "Ъ" | "Ы" | "Ь" | "Э" | "Ю" | "Я" | "а" | "б" | "в" | "г" | "д" | "е" | "ж" | "з" | "и" | "й" | "к" | "л" | "м" | "н" | "о" | "п" | "р" | "с" | "т" | "у" | "ф" | "х" | "ц" | "ч" | "ш" | "щ" | "ъ" | "ы" | "ь" | "э" | "ю" | "я".

Цифра = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9".

Символ-подчеркивания = "_".

Знак-денежной-единицы = "\$".

2.1.3. Семантика ЯОМ

В отличие от синтаксиса контекстно-свободных языков, для описания которого разработаны и успешно применяются различные формальные подходы, их семантика обычно описывается неформально, т.е. на вербальном уровне. Существующие методологии формального определения семантики контекстно-

свободных языков не получили достаточно широкого распространения [44].
Ниже приведено неформальное описание семантики ЯОМ.

В заголовке проекта задается идентификатор проекта – уникальное в рамках САПР имя проекта, определенное при создании информационной модели проекта и необходимое для идентификации данного проекта среди множества других проектов, разрабатываемых в САПР. Длина идентификатора проекта ограничена 35 символами. Под данным именем проект регистрируется в журнале регистрации идентификаторов проектов (файл MS\$JNL:IDOPRJ.IDX) при создании информационной модели проекта. Пример заголовка проекта:

PROJECT СБИС_контроллера

В заголовке каждой проектной процедуры задается ее идентификатор – уникальное в рамках данного проекта имя проектной процедуры. Требование уникальности имени проектной процедуры обусловлено тем, что это имя используется в качестве значения главного ключа, однозначно идентифицирующего данную процедуру в файле рабочей модели процесса проектирования. Длина идентификатора проектной процедуры ограничена 35 символами. Пример заголовка процедуры:

PROCEDURE Генерация_тест_наборов

В теле каждой проектной процедуры указываются следующие информационные элементы, определяющие реквизиты проектной процедуры:

идентификатор функционального узла, проектирование которого выполняется с помощью данной проектной процедуры. Длина идентификатора ограничена 35 символами. Полный перечень идентификаторов функциональных узлов объекта проектирования и их иерархия определяются в информационной модели проекта. Пример идентификатора проектируемого узла:

NODE Блок_декодирования_команд

идентификатор типа проектного решения, вырабатываемого данной проектной процедурой для проектируемого функционального узла. Длина идентификатора ограничена 35 символами. Перечень допустимых идентификаторов типов проектных решений для каждого узла задается в информационной модели проекта. Пример идентификатора типа проектного решения:

TYPE Тест_наборы

фамилия и, возможно, инициалы разработчика, ответственного за выполнение данной проектной процедуры. Длина последовательности символов,

представляющей фамилию и инициалы ответственного исполнителя, ограничена 20 символами. Например:

USER Иванов_И_И

название проектирующего подразделения, выполняющего данную проектную процедуру. Длина последовательности символов, представляющей название подразделения, ограничена 20 символами. Например:

SUBDIV Лаборатория_1

плановая дата начала работ по данной проектной процедуре. Дата указывается в системном формате: ЧЧ-МММ-ГГГГ, где ЧЧ – число месяца, МММ – сокращенное название месяца, ГГГГ – год. Например:

START 15-АПП-1998

Для обеспечения компактности при хранении (16-разрядное машинное слово) и удобства при сравнении все даты кодируются (упаковываются) по следующей формуле:

$$\text{Дата} = 12 \times 31 \times (\text{Год} - 1900) + 31 \times \text{Номер_месяца} + \text{Число_месяца} - 1,$$

где *Номер_месяца* = 1..12, *Число_месяца* = 1..31;

плановая дата окончания работ по данной проектной процедуре. Дата указывается в системном формате (см. выше). Например:

FINISH 15-ОКТ-1998

перечень (возможно пустой) идентификаторов проектных процедур, вырабатывающих проектные решения, которые используются данной проектной процедурой. Например, для проектной процедуры **Генерация_тест_наборов** можно определить следующую входную ссылку:

INPUT

BEGIN

Логическое_Моделирование

END

Примечание. В исходном описании ключевые слова операторов, длина которых превышает 4 символа, могут быть сокращены до первых четырех символов. Например, можно использовать **PROJ** вместо **PROJECT**, **PROC** вместо **PROCEDURE**, **SUBD** вместо **SUBDIV** и т.д. Данное допущение обусловлено упрощением, принятым при реализации языка. Трансляция исходного описания, выполняемая генератором рабочей модели, осуществляется с использованием таблицы ключевых слов операторов, в которой идентификаторы ключевых слов представлены в сокращенном варианте.

2.2. Схема создания информационной модели процесса проектирования

Разработка информационной модели процесса проектирования осуществляется в три этапа. На первом этапе осуществляется выбор одного из множества технологических маршрутов проектирования, обеспечиваемых интегрированной САПР, т.е. задается (и прорабатывается) так называемый вектор обработки. При этом определяются идентификаторы проектирующих подсистем, реализующих необходимые проектные процедуры. Перечень идентификаторов устанавливается и утверждается руководителем САПР по согласованию с администратором информационной системы. В дальнейшем это множество идентификаторов, включенное в базу данных информационной системы, используется при разработке и сопровождении информационных моделей проектов с целью указания подсистем, имеющих право доступа к проектным решениям, выработанным для каждого функционального узла каждого разрабатываемого проекта (см. ниже).

На втором этапе создается иерархическая схема проекта, адекватно отражающая структуру объекта проектирования. На этом этапе выполняется иерархическая декомпозиция объекта на фрагменты, в отношении которых также может быть выполнена декомпозиция, и так далее – до тех пор, пока сложность каждого фрагмента не станет приемлемой для выполнения его проектирования с помощью имеющихся инструментальных средств. В дальнейшем фрагменты объекта проектирования получают относительную свободу при прохождении по маршруту проектирования, т.е. их проектирование в определенном смысле выполняется самостоятельно [45, 46].

Иерархическая схема проекта используется при разработке исходного описания информационной модели проекта и создается в среде информационной системы интегрированной САПР. Для подготовки исходного описания разработан специальный язык описания проекта (ЯОП) [47]. Этот язык позволяет задавать следующие элементы информационной модели проекта:

уникальный идентификатор, однозначно определяющий данный проект среди множества разрабатываемых в САПР проектов;

перечень уникальных (в рамках данного проекта) идентификаторов проектируемых функциональных узлов объекта проектирования;

иерархическую подчиненность (структуру) функциональных узлов объекта проектирования;

перечень идентификаторов типов проектных решений, вырабатываемых для каждого функционального узла объекта проектирования и выбираемых из множества (домена) типов проектных решений, определенного в базе данных информационной системы;

перечень идентификаторов проектирующих подсистем, реализующих проектные процедуры, которые имеют право доступа к проектному решению указанного типа для каждого проектируемого узла;

перечень идентификаторов проектирующих подсистем, реализующих проектные процедуры, ответственным исполнителям которых необходимо послать оповещение в случае изменения проектного решения для проектируемого узла.

Трансляция исходного описания информационной модели проекта выполняется с помощью генератора модели проекта (ГМП). Для визуальной проверки сгенерированной рабочей модели проекта используется листинг рабочей модели, формируемый одной из сервисных программ. При необходимости рабочую модель проекта можно модифицировать либо путем корректировки исходного описания модели и повторной его трансляции, либо с помощью программ интерактивной (оперативной) корректировки. Рабочая модель проекта используется информационной системой для решения задач управления фондом проектных решений и, в частности, для предотвращения несанкционированного доступа к контейнерам с проектными решениями со стороны проектирующих подсистем.

На третьем этапе создается информационная модель процесса проектирования путем выполнения “привязки” фрагментов объекта проектирования к проектным процедурам конвейера обработки, представляющего технологический маршрут проектирования. На этом этапе решается вопрос о возможности параллельного выполнения некоторых проектных процедур маршрута, а также осуществляется календарное планирование работ, выполняемых проектными процедурами маршрута. Информация о векторе проектирования, т.е. о техноло-

гическом маршруте проектирования, и результаты иерархической декомпозиции проекта, полученные на предыдущих этапах, используются при разработке исходного описания информационной модели процесса проектирования.

Исходное описание, записанное на ЯОМ (см. выше), транслируется с помощью генератора рабочей модели. Для проверки сформированной рабочей модели процесса проектирования используется листинг модели, формируемый одной из сервисных программ. При необходимости полученную рабочую модель процесса проектирования можно модифицировать либо путем корректировки исходного описания модели и повторной его трансляции, либо с помощью программ диалоговой (оперативной) корректировки. Укрупненная структурная схема, описывающая этапы процесса разработки рабочей модели, приведена на рис. 2.1.

2.3. Программные средства поддержки модели процесса проектирования

К программным средствам поддержки информационной модели процесса проектирования относятся: генератор рабочей модели, программы интерактивной (оперативной) корректировки рабочей модели, сервисные программы для работы с рабочей моделью. Ниже перечисленные программные средства описаны более подробно.

2.3.1. Генератор рабочей модели процесса проектирования

Генератор рабочей модели (ГРМ) предназначен для выполнения синтаксического и семантического анализа исходного описания информационной модели, записанного на ЯОМ, и автоматического формирования файла рабочей модели процесса проектирования. Исходное описание информационной модели представляет собой текстовый файл, созданный с помощью одного из редакторов текста, входящих в состав базового программного обеспечения операционной системы.

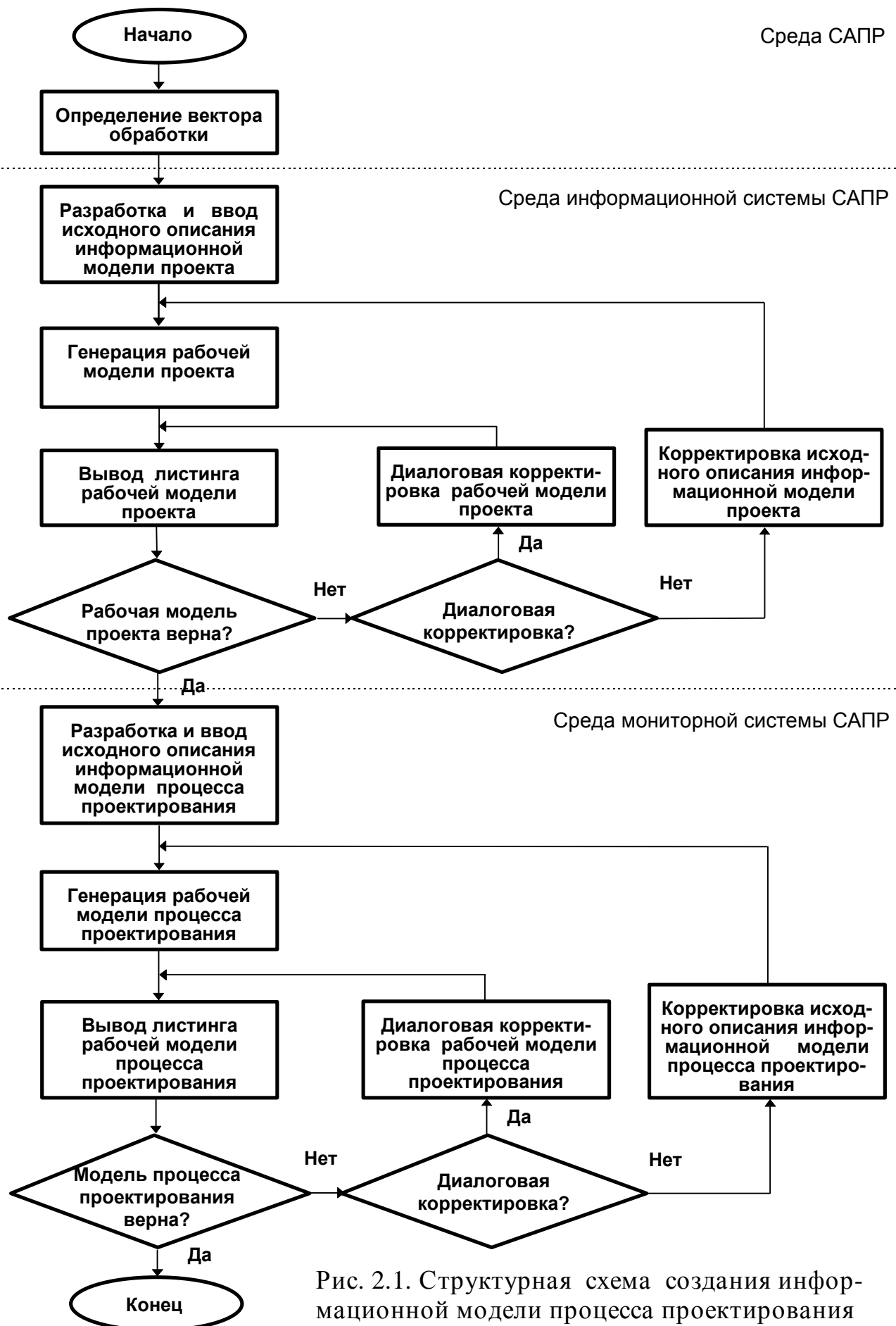


Рис. 2.1. Структурная схема создания информационной модели процесса проектирования

Функциональная схема ГРМ. Генерация рабочей модели процесса проектирования выполняется по трехпроходной схеме. Хотя многопроходная схема трансляции требует несколько больших временных затрат на генерацию рабочей модели, чем однопроходная, она позволяет существенно снизить сложность процедур, реализующих каждый проход генератора. Это предопределило выбор многопроходной схемы трансляции для генератора – тем более, что для каждого проекта ГРМ используется, как правило, однократно – в начале информационного моделирования процесса проектирования, а все дальнейшие манипуляции с рабочей моделью выполняется с помощью программ оперативной корректировки.

Во время первого прохода проверяется, что проект с указанным идентификатором зарегистрирован в журнале идентификаторов проектов (файл MS\$JNL:IDOPRJ.IDX), т.е. для данного проекта существует иерархическая модель проекта, созданная в среде информационной системы САПР (см. выше рис. 2.1), и формируется таблица идентификаторов проектных процедур.

Во время второго прохода осуществляется синтаксический и семантический анализ операторов языка ЯОМ, содержащихся в исходном описании информационной модели, и создается таблица входных ссылок для проектных процедур. Если во время второго прохода обнаруживается синтаксическая или семантическая ошибка, дальнейшая обработка исходного описания прекращается и на экран терминала выводится соответствующее сообщение. Перечень ошибок, распознаваемых ГРМ при трансляции исходного описания, приведен в табл. 2.2.

Таблица 2.2.

Ошибочные ситуации, распознаваемые ГРМ

Текст сообщения об ошибке	Причина
Превышена предельно допустимая длина идентификатора в операторе <i>ключевое-слово</i> (каждому из операторов <i>PROJECT</i> , <i>PROCEDURE</i> , <i>TYPE</i> , <i>USER</i> , <i>SUBDIV</i> , <i>NODE</i> соответствует собственное сообщение)	В указанном в сообщении операторе обнаружен идентификатор, длина которого превышает предельно допустимую длину (35 или 20 символов – в зависимости от оператора)

Продолжение табл. 2.2

Текст сообщения об ошибке	Причина
Дублирование идентификатора проектной процедуры <i>имя-процедуры</i>	Нарушено требование уникальности имени проектной процедуры в файле рабочей модели процесса проектирования
Дублирование идентификатора проекта <i>наименование-проекта</i>	Нарушено требование уникальности наименования проекта в файле учета проектов (MS\$JNL:IDOPRJ.IDX)
Неверный формат даты в операторе <i>START</i> или <i>FINISH</i>	В операторе <i>START</i> или <i>FINISH</i> использован формат даты, отличный от формата, определяемого синтаксисом ЯОМ
Некорректное задание дат начала и окончания проектных работ в процедуре <i>имя-процедуры</i>	В операторе <i>FINISH</i> указана более ранняя дата, чем дата, использованная в операторе <i>START</i>
Несоответствие плановой даты окончания работ в проектной процедуре <i>имя-процедуры-1</i> и плановой даты начала работ в проектной процедуре <i>имя-процедуры-2</i>	В операторе <i>FINISH</i> в описании предшествующей процедуры указана более поздняя дата, чем дата начала работ, указанная в операторе <i>START</i> в описании текущей процедуры
Отсутствие ожидаемого оператора (каждому из операторов <i>PROJECT</i> , <i>PROCEDURE</i> , <i>TYPE</i> , <i>USER</i> , <i>SUBDIV</i> , <i>NODE</i> , <i>INPUT</i> соответствует собственное сообщение)	В указанной строке исходного описания отсутствует один из перечисленных операторов, который должен быть использован в соответствии с синтаксисом ЯОМ
Отсутствие ожидаемой операторной скобки <i>BEGIN</i> или <i>END</i> (каждой из операторных скобок соответствует собственное сообщение)	В указанной строке исходного описания отсутствует одна из перечисленных операторных скобок, которая должна быть использована в теле оператора <i>INPUT</i> в соответствии с синтаксисом ЯОМ

Продолжение табл. 2.2

Текст сообщения об ошибке	Причина
Недопустимое использование оператора <i>ключевое-слово</i> (каждому из операторов <i>PROJECT</i> , <i>PROCEDURE</i> , <i>TYPE</i> , <i>USER</i> , <i>SUBDIV</i> , <i>NODE</i> , <i>INPUT</i> соответствует свое сообщение)	В указанной строке использован один из перечисленных операторов, который не может применяться в данном месте исходного описания в соответствии с синтаксисом ЯОМ
Недопустимое использование операторной скобки <i>BEGIN</i> или <i>END</i> (каждой из операторных скобок соответствует свое сообщение)	В указанной строке использована одна из перечисленных операторных скобок, которая не может применяться в данном месте исходного описания в соответствии с синтаксисом ЯОМ
Проектная процедура <i>имя-процедуры</i> не содержит ни входных, ни выходных ссылок	Указанной проектной процедуре соответствует изолированная вершина в орграфе рабочей модели процесса проектирования
Ссылка на дублированный идентификатор проектной процедуры <i>имя-процедуры</i>	В качестве входной ссылки использован идентификатор проектной процедуры, для которого обнаружена ситуация дублирования (см. выше)
Ссылка на неопределенный идентификатор проектной процедуры <i>имя-процедуры</i>	В качестве входной ссылки использован неопределенный идентификатор проектной процедуры
Циклическая ссылка в проектной процедуре <i>имя-процедуры</i>	В области входных ссылок проектной процедуры обнаружен идентификатор данной проектной процедуры, т.е. проектная процедура ссылается на саму себя
Ссылка на идентификатор проектной процедуры <i>имя-процедуры</i> , длина которого превышает предельно допустимую	В области ссылок оператора <i>INPUT</i> обнаружен идентификатор, длина которого превышает 35 символов

мую длину	
-----------	--

Во время третьего прохода генерируется файл рабочей модели процесса проектирования, который представляет собой индексный файл, поддерживаемый файловой системой операционных систем МВС версии 3.7 и МОС-32М версии 4.5 (и выше). Укрупненная функциональная схема ГРМ представлена на рис. 2.2 (сплошные направленные линии отражают информационные потоки, пунктирные – поток управления).

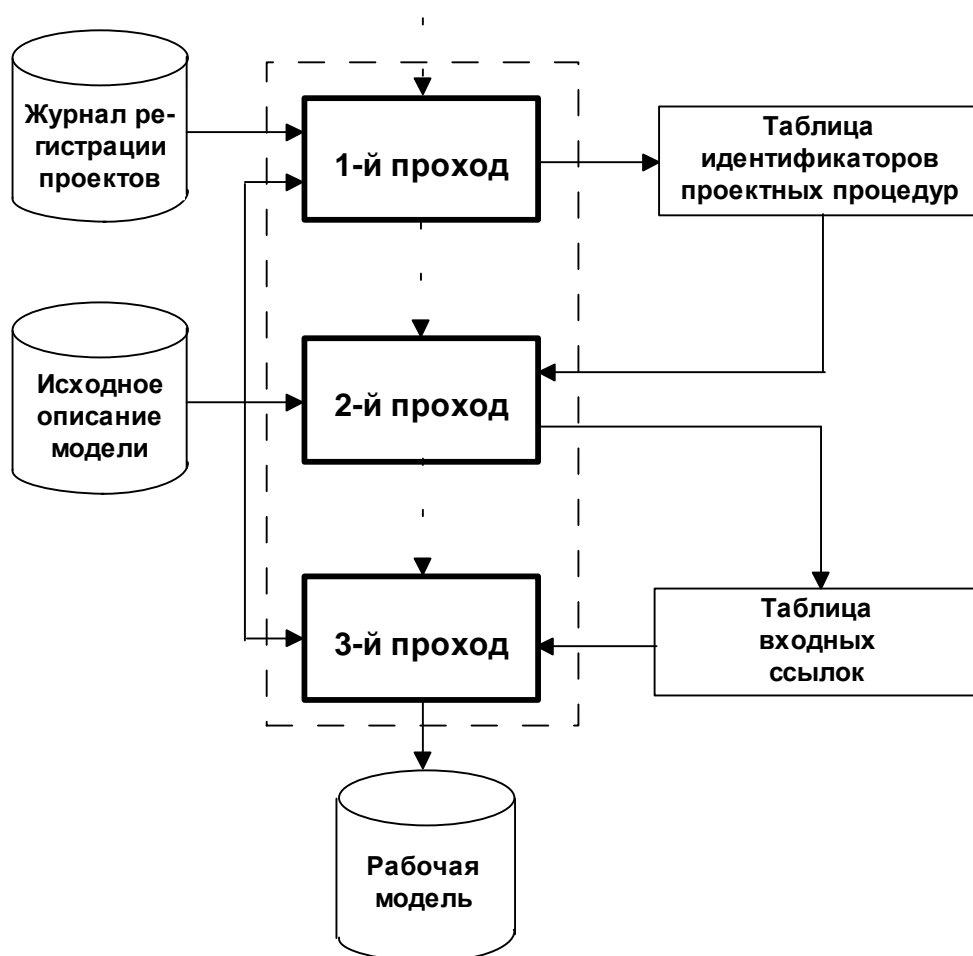


Рис. 2.2. Укрупненная функциональная схема ГРМ

Структурная схема ГРМ. Структура ГРМ представлена четырьмя уровнями (рис. 2.3). Верхний уровень образован головной процедурой программы – *GEN_Main*. Эта процедура выполняет последовательный вызов процедур *GEN_Pass1*, *GEN_Pass2* и *GEN_Pass3*, расположенных на следующем и-

рархическом уровне и осуществляющих первый, второй и третий проходы генератора соответственно.

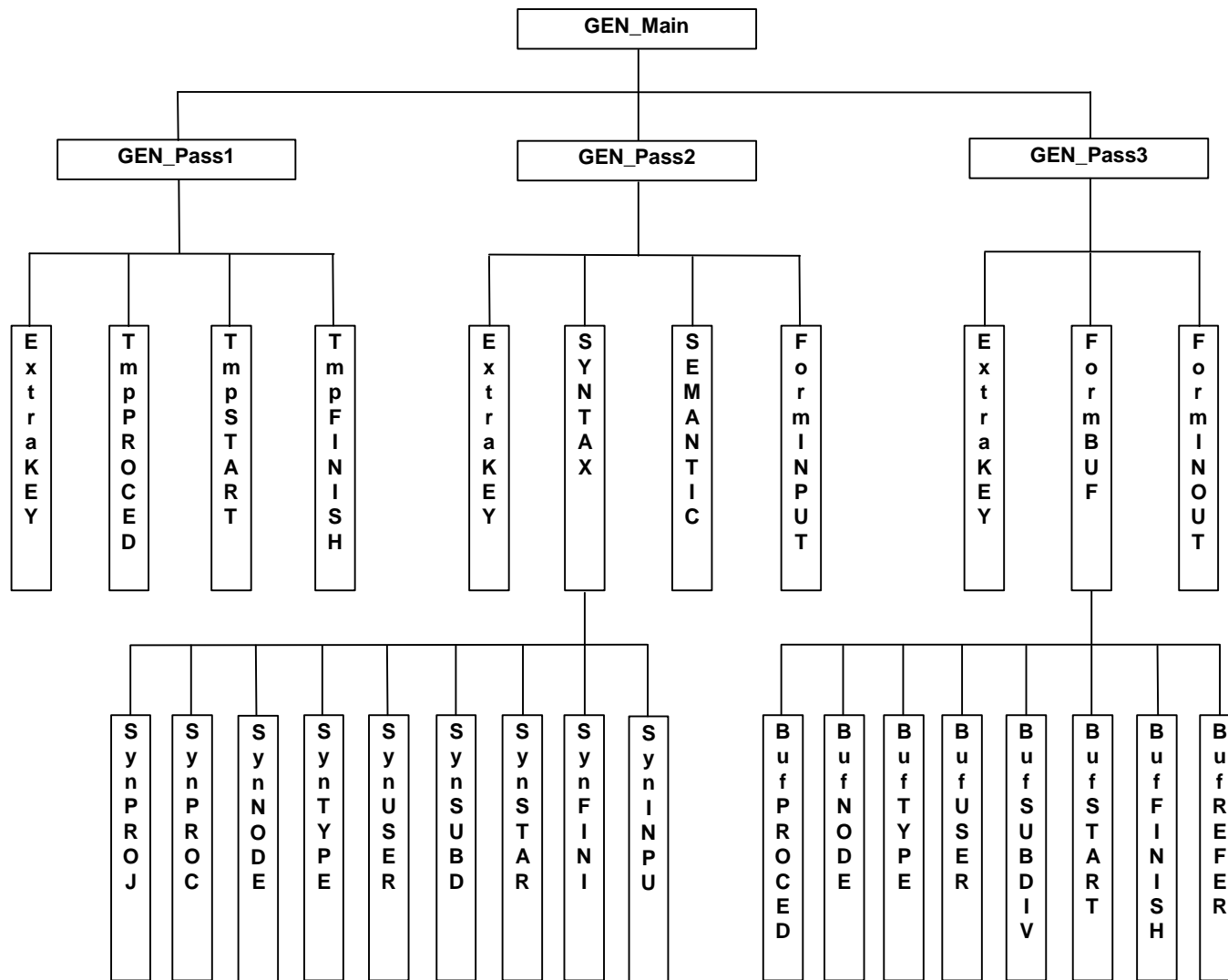


Рис. 2.3. Структурная схема ГРМ

Процедура *GEN_Pass1* выполняет 1) настройку таблицы ключевых слов, 2) чтение (в цикле) записей из файла исходного описания модели и 3) вызов для каждой считанной записи следующих процедур третьего уровня:

ExtraKEY – выделение ключевого слова оператора, поиск его сокращенного аналога в таблице ключевых слов и возврат адреса соответствующей процедуры обработки;

TmpPROCED – занесение идентификатора проектной процедуры в рабочую таблицу (таблица идентификаторов проектных процедур);

TmpSTART – занесение плановой даты начала работ по данной проектной процедуре в рабочую таблицу;

TmpFINISH – занесение плановой даты окончания работ по данной проектной процедуре в рабочую таблицу.

Процедура *GEN_Pass2* выполняет 1) перенастройку таблицы ключевых слов, 2) чтение (в цикле) записей из файла исходного описания модели и 3) вызов для каждой считанной записи следующих процедур третьего уровня:

ExtraKEY – выделение ключевого слова оператора, поиск его сокращенного аналога в таблице ключевых слов и возврат адреса соответствующей процедуры синтаксического анализа оператора;

SYNTAX – вызов процедуры синтаксического анализа оператора, используя ее адрес, полученный процедурой *ExtraKEY*;

SEMANTIC – семантический анализ операторов *START*, *FINISH* и *INPUT* (анализ семантической корректности дат начала и окончания работ, наличие циклических ссылок проектной процедуры);

FormINPUT – занесение входных ссылок проектной процедуры в рабочую таблицу (таблица входных ссылок проектных процедур).

Процедура *GEN_Pass3* выполняет 1) перенастройку таблицы ключевых слов, 2) чтение (в цикле) записей из файла исходного описания модели и 3) вызов для каждой считанной записи следующих процедур третьего уровня:

ExtraKEY – выделение ключевого слова оператора, поиск его сокращенного аналога в таблице ключевых слов и возврат адреса соответствующей процедуры обработки;

FormINOUT – формирование “плавающей” части буфера вывода, содержащего входные и выходные ссылки проектной процедуры;

FormBUF – формирование буфера вывода и запись его содержимого в выходной файл.

Процедура *SYNTAX*, используя адрес, полученный процедурой *ExtraKEY* (см. выше), вызывает для выполнения синтаксического анализа одну из следующих процедур четвертого (нижнего) уровня:

SynPROJ – синтаксический разбор оператора *PROJECT*;

SynPROC – синтаксический разбор оператора *PROCEDURE*;

SynNODE – синтаксический разбор оператора *NODE*;

SynTYPE – синтаксический разбор оператора *TYPE*;

SynUSER – синтаксический разбор оператора *USER*;

SynSUBD – синтаксический разбор оператора *SUBDIV*;

SynSTAR – синтаксический разбор оператора *START*;

SynFINI – синтаксический разбор оператора *FINISH*;

SynINPU – синтаксический разбор оператора *INPUT*.

Процедура *FormBUF*, используя адрес, полученный процедурой *ExtraKEY* (см. выше), вызывает одну из следующих процедур четвертого (нижнего) уровня:

BufPROCED – обработка идентификатора проектной процедуры;

BufNODE – обработка идентификатора проектируемого узла;

BufTYPE – обработка идентификатора типа проектного решения;

BufUSER – обработка идентификатора (имени) ответственного исполнителя;

BufSUBDIV – обработка идентификатора (наименования) подразделения, выполняющего работы в рамках проектной процедуры;

BufSTART – обработка плановой даты начала работ для проектной процедуры;

BufFINISH – обработка плановой даты окончания работ для проектной процедуры;

BufREFER – формирование входных и выходных ссылок проектной процедуры.

Особенности реализации ГРМ. К особенностям ГРМ можно отнести отсутствие у него фазы лексического анализа в классическом понимании данного понятия [48–53]. Вместо этого лексический анализ как бы “размыт” по

проходам ГРМ. Ядром транслирующей схемы ГРМ является таблица ключевых слов операторов ЯОМ, состоящая из двух частей – фиксированной, содержащей символьные строки сокращенных ключевых слов операторов языка (***PROJ, PROC, TYPE, USER, SUBD, NODE, STAR, FINI, INPU***) и изменяемой (перенастраиваемой), содержащей адреса соответствующих процедур для обработки операторов. При запуске ГРМ выполняется инициализация таблицы ключевых слов операторов (*KEYWORD_TABLE*). В начале каждого прохода выполняется ее настройка, заключающаяся в установке адресов соответствующих процедур обработки для каждого из операторов ЯОМ (рис. 2.4).

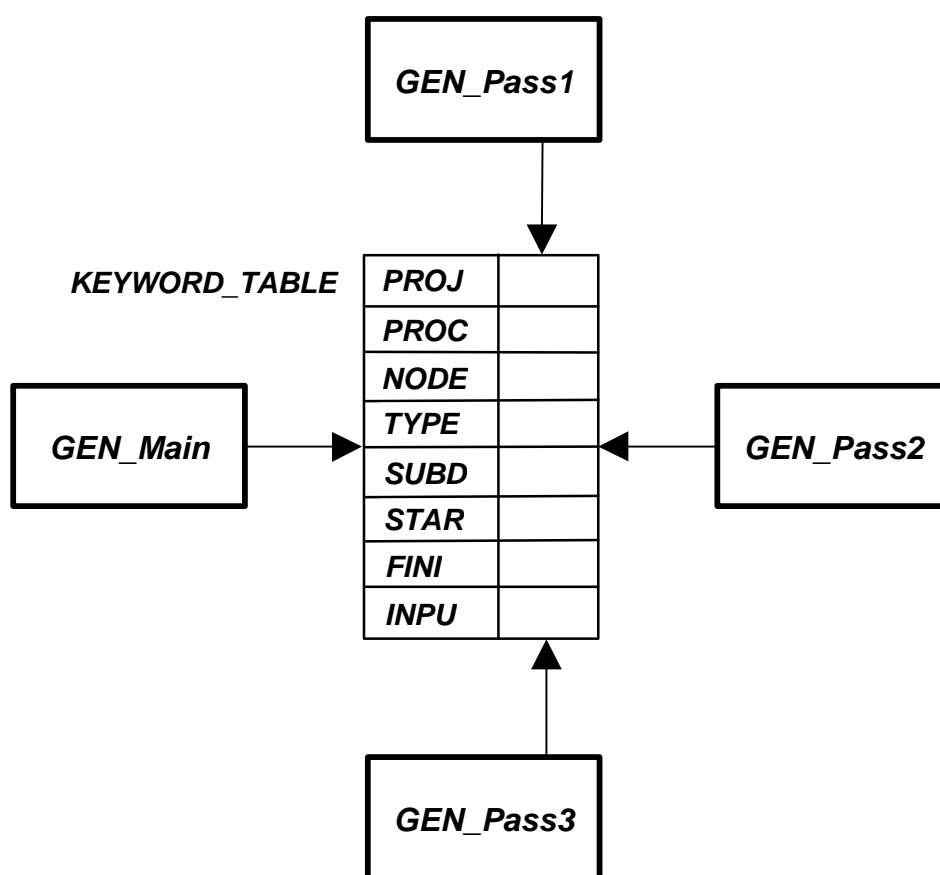


Рис. 2.4. Процедуры ГРМ, выполняющие инициализацию и настройку таблицы ключевых слов операторов ЯОМ

Процедура *GEN_Pass1* устанавливает в *KEYWORD_TABLE* адреса процедур *TmpPROCED*, *TmpSTART* и *TmpFINISH*. В свою очередь процедура *GEN_Pass2* устанавливает адреса процедур *SynPROJ*, *SynPROC*, *SynNODE*, *SynTYPE*, *SynUSER*, *SynSUBD*, *SynSTAR*, *SynFINI* и *SynINPU*, а процедура *GEN_Pass3* – адреса процедур *BufPROCED*, *BufNODE*, *BufTYPE*, *BufUSER*,

BufSUBDIV, *BufSTART*, и *BufFINISH*. Для незадействованных на данном проходе входов таблицы *KEYWORD_TABLE* поля адреса устанавливаются в 0.

Затем вызывается процедура *ExtraKEY*, выполняющая выделение ключевого слова оператора в считанной строке, сопоставление его с ключевыми словами, имеющимися в таблице (сравнение выполняется по первым четырем символам), и возврат адреса соответствующей процедуры обработки при совпадении.

ГРМ включен в состав подсистемы управления центральной МС интегрированной САПР МСВТ. Запуск его выполняется выбором пункта “Генерация рабочей модели” в меню “Работа с информационной моделью”.

В качестве инструментального языка при разработке ГРМ использован язык макроассемблера *MACRO-32* [54–59] и программные средства его поддержки, входящие в состав базового системного ПО 32-разрядных мини- и микроЭВМ семейства “Электроника”: макроассемблер, редактор связей (компоновщик), системные библиотеки подпрограмм [60–64]. Файл выполняемого образа программы (*GENMTP.EXE*) имеет размер приблизительно равный 47 Кбайтам.

Структура файла рабочей модели. Как отмечалось в Главе 1, логическим представлением модели процесса проектирования является сетевая структура, узлы которой представляют проектные процедуры, для которых выполнена поименная привязка к фрагментам объекта проектирования.

Физическим представлением рабочей модели процесса проектирования является индексный файл, поддерживаемый файловой системой СУЗ-32 операционных систем МВС версии 3.7 и МОС-32 версии 4.5 [65, 66]. Каждая запись этого файла имеет структуру, представленную в табл. 2.3. Главным ключом записи является поле, содержащее идентификатор проектной процедуры.

Таблица 2.3

Структура записи файла рабочей модели

Содержимое поля	Размер поля (в байтах)
Идентификатор проектной процедуры	35
Идентификатор типа проектного решения	35
Код состояния проектной процедуры	1
Плановая дата начала работы (в системном формате)	2
Фактическая дата начала работы (в системном формате)	2
Плановая дата окончания работы (в системном формате)	2
Фактическая дата окончания работы (в системном формате)	2
Имя ответственного исполнителя	20
Название проектирующего подразделения	20
Идентификатор функционального узла объекта	35
Счетчик входных ссылок	2
Счетчик выходных ссылок	2
Входные и выходные ссылки (см. Примечание)	1050

Примечание. Область ссылок содержит идентификаторы проектных процедур, в которых ссылаются на данную процедуру (входные ссылки), и идентификаторы проектных процедур, на которые ссылается данная проектная процедура (выходные ссылки).

2.3.2. Программы оперативной корректировки рабочей модели

После того, как сгенерирована рабочая модель процесса проектирования, может возникнуть необходимость ее модификации (например, из-за перенастройки технологического маршрута проектирования). Первоначально корректировку рабочей модели процесса проектирования можно выполнить двумя способами (см. выше рис. 2.1): либо, изменив (скорректировав) исходное описание информационной модели, заново сгенерировать рабочую модель, либо

воспользоваться программами интерактивной (оперативной) корректировки рабочей модели, имеющимися в мониторной системе интегрированной САПР [67–71].

Однако после того как инициировано выполнение одной или нескольких проектных процедур конвейера проектирования, изменение рабочей модели следует выполнять только с помощью программ оперативной корректировки. Это требование обусловлено тем, что рабочая модель процесса проектирования является динамической моделью, отражающей текущее состояние процесса проектирования (с точки зрения управления этим процессом). Значения ряда реквизитов проектной процедуры автоматически изменяются мониторной системой по мере того, как в процессе проектирования происходят различные события (например, передача контейнера с проектным решением информационной системе приводит к замене текущего рабочего состояния процедуры на состояние “локально завершена”). В то же время ГРМ не может выполнить произвольную установку значений для этих реквизитов, поскольку в ЯОМ отсутствуют соответствующие средства. Например, диапазон значений рабочего состояния процедуры в ГРМ ограничен всего двумя значениями – “пассивна” и “ожидания”, т.е. не включает остальных четырех значений: “активна”, “приостановлена”, “завершена локально” и “завершена глобально”.

Задача оперативной корректировки рабочей модели процесса проектирования распадается на ряд подзадач, среди которых можно выделить следующие:

- удаление проектной процедуры из рабочей модели;

- включение проектной процедуры в рабочую модель;

- модификация реквизитов проектной процедуры (плановых дат начала и окончания работ, имени ответственного исполнителя, наименования проектирующего подразделения, состояния процедуры и других);

- удаление входной или выходной ссылки из проектной процедуры;

- включение входной или выходной ссылки в проектную процедуру.

Кроме того, к указанному выше перечню основных задач можно добавить следующие три вспомогательные задачи:

- поиск проектных процедур с пустой областью ссылок, т.е. поиск изолированных вершин в орграфе рабочей модели процесса проектирования;

поиск циклических ссылок проектных процедур, т.е. поиск контурных путей в орграфе рабочей модели процесса проектирования (элементарные контурные пути, называемые петлями, обнаруживает ГРМ);

вывод листинга рабочей модели процесса проектирования.

Ниже кратко описана каждая из перечисленных задач, выполняемых в интерактивном режиме.

Удаление проектной процедуры. Данная задача выполняется с помощью программы *DeleteRecord* (файл DELRCRD.EXE), запускаемой выбором пункта “Удаление проектной процедуры” в меню “Работа с информационной моделью”. Структурным представлением рабочей модели процесса проектирования является орграф, вершины которого – проектные процедуры, а дуги – информационные связи между проектными процедурами. Таким образом, задача удаления проектной процедуры эквивалентна задаче удаления вершины орграфа и модификации дуг, инцидентных удаленной вершине. Логически проектная процедура представляет собой запись индексного файла, а сама рабочая модель – индексный файл. Таким образом, задача удаления проектной процедуры сводится к задаче удаления записи из индексного файла и выполнению модификации в полях других записей, связанных с удаленной записью. После удаления проектной процедуры из рабочей модели производится соответствующая запись в журнал учета корректировок рабочей модели (файл MS\$JNL:CORJNL.IDX) и автоматически рассылается оповещение о выполненной корректировке модели ответственным исполнителям КПП.

Программа *DeleteRecord* написана на языке системного программирования *Bliss-32* [71, 72]. Файл выполнимого образа программы (DELRCRD.EXE) имеет размер приблизительно равный 13 Кбайтам.

Включение проектной процедуры. Данная задача выполняется с помощью программы *InsertRecord* (файл INSRCRD.EXE), запускаемой выбором пункта “Включение проектной процедуры” в меню “Работа с информационной моделью”. Задача модификации реквизитов проектной процедуры сводится к задаче ввода реквизитов для новой проектной процедуры, включения новой записи в индексный файл и выполнению модификации в полях других записей, связанных с включенной записью. После включения новой проектной процедуры в рабочую модель производится соответствующая запись в журнал учета

корректировок рабочей модели (файл MS\$JNL:CORJNL.IDX) и автоматически рассылается оповещение о выполненной корректировке модели ответственным исполнителям КПП.

Программа *InsertRecord* написана на языке *Bliss-32* [71, 72]. Файл выполняемого образа программы (INSRCRD.EXE) имеет размер равный 18 Кбайтам.

Модификация реквизитов проектной процедуры. Данная задача выполняется с помощью программы *ModifyRek* (файл MODRKVS.EXE), запускаемой выбором пункта “Модификация реквизитов проектной процедуры” в меню “Работа с информационной моделью”. Поочередно на экран выводятся запросы на ввод ряда внутренних реквизитов проектной процедуры (плановая дата начала работ, плановая дата окончания работ, имя ответственного исполнителя проектной процедуры, название проектирующего подразделения), а затем производится замена старых реквизитов проектной процедуры введенными новыми. После изменения реквизитов проектной процедуры производится соответствующая запись в журнал учета корректировок рабочей модели (файл MS\$JNL:CORJNL.IDX) и автоматически рассылается оповещение о выполненной корректировке модели ответственным исполнителям КПП.

Программа *ModifyRek* написана на языке *Bliss-32* [71, 72]. Файл выполняемого образа программы (MODRKVS.EXE) имеет размер приблизительно равный 14 Кбайтам.

Удаление ссылки из проектной процедуры. Данная задача выполняется с помощью программы *DeleteReference* (файл DELREF.EXE), запускаемой выбором пункта “Удаление ссылки из проектной процедуры” в меню “Работа с информационной моделью”. Необходимость в данной операции не очевидна, так как при выполнении операции “Удаление проектной процедуры” обновление связей в проектных процедурах, ссылающихся на удаленную процедуру, выполняется автоматически. Однако эта операция является составной частью более сложной операции, заключающейся в перенастройке связей между проектными процедурами, и выполняется совместно с операцией “Включение ссылки в проектную процедуру” (см. ниже).

Программа *DeleteReference* написана на языке *Bliss-32* [71, 72]. Файл выполняемого образа программы (DELREF.EXE) имеет размер равный 12 Кбайтам.

Включение ссылки в проектную процедуру. Данная задача выполняется с помощью программы *InsertReference* (файл INSREF.EXE), запускаемой выбором пункта “Включение ссылки в проектную процедуру” в меню “Работа с информационной моделью”. Эта операция является составной частью более сложной операции, заключающейся в перенастройке связей между проектными процедурами, и выполняется совместно с операцией “Удаление ссылки из проектной процедуры”.

Программа *InsertReference* написана на языке *Bliss-32* [71, 72]. Файл полного образа программы (INSREF.EXE) имеет размер равный 13 Кбайтам.

Поиск проектных процедур с пустой областью ссылок. Данная задача выполняется с помощью программы *SearchNodes* (файл SEANDS.EXE), запускаемой выбором пункта “Поиск проектных процедур с пустой областью ссылок” в меню “Работа с информационной моделью”. Если орграф рабочей модели имеет изолированные вершины, т.е. рабочая модель содержит проектные процедуры с пустой областью ссылок, на экран терминала выводится перечень идентификаторов этих процедур. Наличие в рабочей модели проектных процедур с пустой областью ссылок является признаком ее некорректности, возникшей возможно при ее модификации. В орграфе корректной рабочей модели изолированные вершины отсутствуют.

Программа *SearchNodes* написана на языке *Bliss-32* [71, 72]. Файл полного образа программы (SEANDS.EXE) имеет размер приблизительно равный 6 Кбайтам.

Поиск циклических ссылок проектных процедур. Данная задача выполняется с помощью программы *SearchCycles* (файл SEACYCL.EXE), запускаемой выбором пункта “Поиск циклических ссылок проектных процедур” в меню “Работа с информационной моделью”. Если в орграфе рабочей модели имеются контурные пути, т.е. пути у которых начало и конец представлены одной и той же вершиной, на экран терминала выводится перечень идентификаторов процедур, образующих эти контуры. Наличие в рабочей модели контурных путей является признаком ее некорректности, возникшей, возможно, при ее модификации. В орграфе корректной рабочей модели контурные пути отсутствуют.

Программа *SearchCycles* написана на языке *Bliss-32* [71, 72]. Файл выполняемого образа программы (SEACYCL.EXE) имеет размер приблизительно равный 9 Кбайтам.

Формирование и вывод листинга рабочей модели. Данная задача выполняется с помощью программы *Listing* (файл LISTING.EXE), запускаемой выбором пункта “Вывод листинга рабочей модели” в меню “Работа с информационной моделью”. Программа обеспечивает следующие четыре варианта формирования листинга рабочей модели процесса проектирования, отраженные в меню:

1. Вывод по идентификатору процедуры.
2. Вывод по рабочему состоянию процедуры.
3. Вывод входных и выходных ссылок.
4. Вывод полного листинга.

При выборе первого варианта меню и вводе идентификаторов проекта и проектной процедуры формируется листинг, содержащий информацию об указанной проектной процедуре.

При выборе второго варианта меню и вводе кода (номера) конкретного рабочего состояния формируется листинг, содержащий информацию о всех проектных процедурах модели, имеющих указанный код рабочего состояния.

При выборе третьего варианта меню формируется листинг, содержащий информацию о счетчиках входных и выходных ссылок и идентификаторах самих ссылок для каждой проектной процедуры модели.

При выборе четвертого варианта меню формируется полный листинг рабочей модели процесса проектирования.

Листинг рабочей модели, отображенный на экране терминала, при необходимости может быть записан в текстовый файл на диске или выведен на принтер для дальнейшего анализа и использования.

Программа *Listing* написана на языке *Фортран/МВС* (расширение стандартного языка Фортран-77) [73, 74]. Файл выполняемого образа программы (LISTING.EXE) имеет размер приблизительно равный 27 Кбайтам.

Глава 3. Основные задачи поддержки оперативного управления проектом, способы и средства их решения

В настоящей главе представлен ряд основных задач поддержки оперативного управления проектом и процессом проектирования, решение которых возлагается на руководителя проекта, а именно: мониторинг текущего состояния проекта; координация действий различных групп проектировщиков, занятых в работе над проектом; обеспечение и контроль логической целостности проекта; перепланирование и временная увязка различных работ над проектом; реализация глубокого отката в технологическом маршруте проектирования.

Способы и средства решения перечисленных выше задач основываются на использовании информационной модели процесса проектирования, рассмотренной в предыдущих двух главах монографии. Описанные ниже программные средства составляют инструментальный комплекс, предназначенный для поддержки процесса принятия решения руководителем проекта при реализации оперативного управления процессом проектирования.

3.1. Мониторинг текущего состояния проекта

Необходимым условием для принятия обоснованных решений при управлении проектированием является наличие достоверной информации о ходе процесса проектирования. Как отмечалось в Главе 2, рабочая модель является динамической моделью, отражающей текущее состояние проекта и процесса проектирования. В этой связи задача мониторинга текущего состояния проекта может быть сведена к задаче обработки файла рабочей модели с целью извлечения из него необходимой информации.

Наиболее часто руководителю проекта требуется получить ответ на вопрос о состоянии работ, выполняющихся над проектом, т.е. какие проектные процедуры в данный момент уже выполнены, выполняются, ожидают получения проектных решений от других проектных процедур.

В каждой записи файла рабочей модели, соответствующей некоторой проектной процедуре, представлено поле, описывающее рабочее состояние этой

процедуры. Существуют следующие шесть рабочих состояний, каждому из которых соответствует свой код (указан в скобках):

1. Пассивна (0).
2. Ожидания (1).
3. Активна (2).
4. Приостановлена (3).
5. Завершена локально (4).
6. Завершена глобально (5).

В любой момент проектирования каждая проектная процедура в рабочей модели может находиться в одном из следующих состояний: пассивна, ожидания, активна, приостановлена или локально завершена. В состояние “завершена глобально” каждая процедура модели переводится после передачи проекта в архив САПР.

Сразу после генерации файла рабочей модели все проектные процедуры, не имеющие входных ссылок, т.е. начальные процедуры маршрута, установлены в состояние “пассивна”, остальные процедуры – в состояние “ожидания”. Смена текущего состояния проектной процедуры вызывается рядом событий процесса проектирования. Начало работ по данной проектной процедуре, ассоциируемое обычно с приемом контейнера с проектным решением от информационной системы САПР, приводит к смене ее состояния на “активна”. Окончание работ, ассоциируемое с передачей контейнера с проектным решением в информационную систему, вызывает смену состояния на “завершена локально”. Перевод проектных процедур в состояние “завершена глобально” осуществляется при передаче проектной информации в архивную систему САПР. Поскольку центральная МС контролирует выполнение всех информационных обменов в САПР, она автоматически выполняет смену рабочих состояний проектных процедур по результатам этих обменов.

Откат в технологическом маршруте проектирования, являющийся следствием обнаружения ошибки в проектном решении, сформированном одной из ранее выполненных процедур, также вызывает перевод локально завершенных и выполняющихся проектных процедур в состояния “ожидания” и “приостановлена” соответственно. Смена рабочих состояний проектных процедур при откате выполняется автоматически средствами центральной МС.

Для мониторинга текущего состояния проекта используются следующие две программы: *Listing* (файл LISTING.EXE) и *Search_Proc* (файл SEAPRC.EXE). Запуск программы *Listing* выполняется выбором пункта “Вывод листинга рабочей модели” в меню “Работа с информационной моделью”. Программа обеспечивает следующие четыре варианта формирования листинга рабочей модели процесса проектирования, отраженные в меню:

1. Вывод по идентификатору процедуры.
2. Вывод по рабочему состоянию процедуры.
3. Вывод входных и выходных ссылок.
4. Вывод полного листинга.

Для решения задачи мониторинга проекта используются 1-й, 2-й и 4-й пункты указанного меню. При выборе 1-го пункта меню и ввода идентификаторов проекта и проектной процедуры формируется листинг, содержащий информацию об указанной проектной процедуре.

При выборе 2-го пункта меню и ввода кода (номера) конкретного рабочего состояния формируется листинг, содержащий информацию о всех проектных процедурах модели, имеющих указанный код рабочего состояния.

При выборе 4-го пункта меню формируется полный листинг рабочей модели процесса проектирования.

Листинг рабочей модели, отображенный на экране терминала, при необходимости может быть записан в текстовый файл на диске или выведен на принтер для дальнейшего анализа и использования. Пример фрагмента листинга рабочей модели приведен ниже.

```
Идентификатор проектной процедуры: Генерация_тест_наборов
Рабочее состояние: локально завершена
Плановая дата начала работ: 15-АПР-1998
Плановая дата окончания работ: 15-ОКТ-1998
Фактическая дата начала работ: 20-АПР-1998
Фактическая дата окончания работ: 17-ОКТ-1998
Ответственный исполнитель: Иванов_И_И
Подразделение: Лаборатория_1
Наименование узла: Блок_декодирования_команд
Кол-во входных ссылок: 1
Идентификаторы входных ссылок: Логическое_моделирование
Кол-во выходных ссылок: 2
Идентификаторы выходных ссылок: Компоновка_блока_декодирования
Трассировка_блока_декодирования
```

Программа *Listing* написана на языке высокого уровня *Фортран/МВС* [74, 75]. Файл выполнимого образа программы (LISTING.EXE) имеет размер приблизительно равный 27 Кбайтам.

Другой вопрос, на который можно получить ответ, основываясь на данных рабочей модели процесса проектирования, связан с предоставлением информации о проектных процедурах, в которых были превышены плановые сроки выполнения проектных работ, т.е. информации о просроченных работах, и насколько велико это превышение.

В каждой записи файла рабочей модели содержатся поля, предназначенные для хранения плановых и фактических дат начала и окончания работ по соответствующей проектной процедуре. Плановые даты начала и окончания работ формируются ГРМ в соответствии со значениями, указанными в операторах **START** и **FINISH** в исходном описании информационной модели процесса проектирования. Фактические даты начала и окончания работ автоматически формируются средствами центральной МС после того, как подтверждено выполнение запроса на прием или передачу контейнера с проектным решением. Более подробно схема информационных обменов между подсистемами интегрированной САПР описана в Главе 4.

Запуск программы *Search_Proc* выполняется выбором пункта “Поиск процедур с просроченной датой начала работ” в меню “Работа с информационной моделью”. Программа выполняет последовательное чтение (сканирование) всех записей файла рабочей модели и выбор в них содержимого полей “Плановая дата окончания работ”, “Фактическая дата начала работ” и “Фактическая дата окончания работ”.

Если оказывается, что в считанной записи поле “Фактическая дата начала работ” не пустое, т.е. данная проектная процедура получила контейнер с проектным решением от предшествующей процедуры маршрута, а поле “Фактическая дата окончания работ” пустое и плановая дата окончания работ превышает текущую дату (выполняется сравнение упакованных дат), идентификатор данной проектной процедуры выводится на экран терминала. Полученный таким образом список идентификаторов проектных процедур (вместе с их плановыми датами окончания работ) записывается в текстовый файл. Затем эта информация может быть выведена на экран или на принтер и использована для оценки

ситуации, последующего принятия оперативных мер по устранению допущенного срыва плановых сроков работ, перепланирования и временной увязки всех дальнейших проектных работ.

Программа *Search_Proc* написана на языке *Bliss-32* [72, 73]. Файл выполняемого образа программы (SEAPRC.EXE) имеет размер приблизительно равный 7 Кбайтам.

3.2. Координация действий проектировщиков

Проблема координации действий различных групп проектировщиков, занятых в работе над проектом, обусловлена распределенным характером процесса проектирования в интегрированной САПР, когда проектные работы выполняются на совокупности АРМ и ИГС, объединенных в локальную вычислительную сеть (более подробно см. Главу 4).

Базовая операционная система (МВС, МОС-32М, VMS), функционирующая на 32-разрядных АРМ и ИГС, предоставляет возможности для организации межмашинной связи. Эти возможности обеспечиваются двумя обслуживающими программами (утилитами) системы – PHONE и MAIL.

Утилита PHONE является удобным средством для ведения диалога (путем обмена сообщениями) между любыми двумя пользователями, одновременно работающими в системе. Однако эта утилита имеет ряд ограничений, несколько снижающих ее ценность. Во-первых, как уже отмечалось, пользователи, участвующие в диалоге, должны одновременно работать в системе (на одном и том же локальном узле или на разных узлах сети). Во-вторых, передаваемые сообщения не сохраняются, т.е. их нельзя затем повторно просмотреть с целью более тщательного анализа. В-третьих, пользователю для обмена сообщениями приходится прерывать текущую работу, что не всегда удобно. Наконец, для использования утилиты PHONE АРМ пользователя необходимо оснастить терминалом, эмулирующим режимы работы терминалов VT100 и VT220 фирмы DEC. Последнее требование связано с тем, что утилита PHONE делит экран на две области: верхняя половина экрана используется для ввода сообщений, передаваемых пользователем, нижняя – для отображения получаемых им сообщений.

Терминалы младших моделей (например, VT52 и их отечественные аналоги) не поддерживают работу в подобном мультиэкранном режиме.

Утилита PHONE имеет собственный набор команд, включающий:

ANSWER	Ответ на полученный вызов для установки связи.
DIAL [узел::]имя_пользователя	Вызов другого пользователя, который может работать на удаленном узле ЛВС.
DIRECTORY [узел::]	Вывод списка пользователей, работающих в системе. Эту команду можно использовать для получения списка имен пользователей, работающих в данный момент на удаленном узле ЛВС.
EXIT	Возврат на уровень интерпретатора командного языка, т.е. в режим ввода команд DCL.
FACSIMILE имя_файла	Передача в качестве сообщения содержимого указанного файла.
HANGUP	Прерывание сеанса связи.
HELP	Развернутая справка по командам утилиты PHONE.
HOLD	Задержка сеанса связи на некоторое время, пока пользователь занят чем-то другим.
MAIL [узел::]имя_пользователя "сообщение"	Передача указанному пользователю краткого сообщения, заключенного в кавычки. Эта команда утилиты PHONE реализуется средствами утилиты MAIL (см. ниже).
REJECT	Аннулирование ранее установленной связи.
UNHOLD	Действие этой команды противоположно действию команды HOLD.

В мониторной системе утилита PHONE явно не используется. Однако, пользователи, выбрав пункт "Командный язык ОС" в главном меню МС, могут осуществить временный выход в режим ввода команд DCL с целью вызова утилиты PHONE для обмена сообщениями с любым пользователем, работающим в данный момент на любом удаленном узле ЛВС. Для этого в соответ-

вующей команде к имени пользователя добавляют спецификацию узла ЛВС. При обмене сообщениями с пользователями, работающими на этом же локальном узле, спецификация узла обычно не указывается.

Для рассылки системных оповещений в мониторной системе используется утилита MAIL, обеспечивающая полномасштабные возможности электронной почты. В отличие от PHONE, утилита MAIL позволяет передавать сообщения множеству пользователей. Причем необязательно, чтобы все эти пользователи работали в ЛВС одновременно с пользователем, передающим сообщение. Однако каждый из них в процессе работы или при входе в систему будет предупрежден о переданных для него сообщениях. В дальнейшем пользователь, используя возможности утилиты MAIL, может прочитать полученные сообщения, удалить их, переслать их другому пользователю и т.д. Посланные пользователю сообщения помещаются в файл MAIL.MAI в текущем справочнике. Сообщения в этом файле организованы в виде папок, стандартными из которых являются следующие: NEWMAIL (новая почта) – для еще не просмотренных сообщений, MAIL (почта) – для просмотренных сообщений, WASTEBASKET (букв. перевод с англ. “корзина для бумаг”) – для удаленных сообщений.

Утилита MAIL, подобно утилите PHONE, собственный набор команд для работы с сообщениями, который включает:

COPY *название_папки*

Копирование последнего прочитанного или найденного сообщения в указанную папку. Если указанная папка не найдена, то она будет создана по этой команде.

DELETE

DELETE #

DELETE/ALL

Удаление текущего сообщения, сообщения с указанным номером или всех сообщений из текущей папки.

DIRECTORY

DIRECTORY/FOLDER

DIRECTORY *название_папки*

Вывод списка сообщений из текущей папки, списка названий всех имеющихся папок в файле MAIL.MAI или списка сообщений в указанной папке.

EXTRACT *имя_файла*

EXTRACT/NOHEADER

имя_файла

Копирование текущего сообщения, последнего прочитанного (READ) или найденного (SEARCH) сообщения в указанный файл. Если

файл не существует, то он будет создан по этой команде. Переключатель /NOHEADER предотвращает копирование заголовков и сопроводительной информации, формируемых утилитой MAIL, в указанный файл.

FORWARD

[узел::]имя_пользователя

Передача последнего прочитанного сообщения указанному пользователю.

HELP

Развернутая справка по командам утилиты MAIL.

MOVE *название_папки*

Помещает последнее прочитанное или найденное сообщение в указанную папку. Исходное сообщение при этом удаляется. Если указанная папка не существует, то она создается по этой команде.

READ #

READ *название_папки*

Чтение сообщения с указанным номером из текущей папки или чтение первого сообщения из указанной папки.

REPLY

Передача ответного сообщения отправителю последнего считанного сообщения.

SEARCH *“текст”*

Поиск указанной строки текста во всех сообщениях текущей папки и отображение первого из сообщений, в котором эта строка найдена.

SELECT *название_папки*

Установка указанной папки в качестве текущей.

SEND

[узел::]имя_пользователя

SEND/EDIT

Передача сообщения указанному пользователю. Подготовка сообщения с помощью редактора текстов (например, EDT) и последующая его передача указанному пользователю.

С помощью перечисленных выше команд утилиты MAIL справочная подсистема МС реализует рассылку системных оповещений пользователям МС и прием уведомлений, подтверждающих получение сообщений адресатами и их прочтение. Подобная схема взаимодействия различных проектировщиков, занятых в работе над проектом, позволяет скоординировать их усилия в достижении

поставленных целей, а также своевременно довести до каждого ответственного исполнителя указания руководителя проекта.

Рабочая модель процесса проектирования занимает ключевое место в рассматриваемой схеме взаимодействия. Для согласованной работы различных проектных процедур их ответственные исполнители должны быть своевременно оповещены о важнейших событиях процесса проектирования, к которым относятся:

передача контейнера с проектным решением в оперативное хранилище информационной системы САПР;

замена контейнера с проектным решением, хранящегося в хранилище информационной системы САПР, на новый контейнер (обычно после доработки или полной переработки ранее сформированного проектного решения);

корректировка рабочей модели проекта, в результате которой было выполнено удаление проектной процедуры, включение новой проектной процедуры, изменение ряда основных реквизитов и т.п.;

реализация отката в технологическом маршруте проектирования вследствие обнаружения ошибки в проектном решении, сформированном одной из проектных процедур.

В случае возникновения какого-либо из перечисленных выше событий для одной из проектных процедур технологического маршрута МС должна оповестить всех ответственных исполнителей последующих процедур, информационно связанных с первой. Для этой цели выполняется чтение записей файла рабочей модели с целью подготовки списка идентификаторов проектных процедур, а также ответственных исполнителей, которым требуется послать сообщение. Далее выполняется обработка файла полномочий пользователей с целью извлечения из него такой необходимой информации, как входное (учетное) имя пользователя и имя узла ЛВС, выделенного для его работы. Из полученных информационных элементов формируется соответствующая командная строка для запуска утилиты MAIL в качестве порожденного подпроцесса (SPAWN), который и осуществляет рассылку сообщений. Формат команды для рассылки сообщения:

MAIL имя_файла имя_пользователя

где *имя_файла* – спецификация текстового файла, который формируется соответствующими средствами МС и содержит сообщения об изменениях в рабочей модели процесса проектирования.

Пользователь локальной МС для просмотра адресованных ему сообщений может выбрать пункт меню “Просмотр полученных сообщений”. После этого на узел ЛВС с центральной МС (логическое имя MS\$CMS) пользователю, выполнившему модификацию рабочей модели, будет возвращено уведомление о том, что сообщение было прочитано. Для этой цели формируется команда **MAIL REPLY**, которая затем запускается как порожденный подпроцесс. Этот способ предоставляет возможность руководителю проекта проконтролировать получение и прочтение сообщений адресатами.

3.3. Контроль логической целостности проекта

Данная задача выполняется с помощью программы *Report* (файл REPORT.EXE), запускаемой выбором пункта “Вывод отчета о проектных процедурах” в меню “Работа с информационной моделью”. Программа обеспечивает следующие 9 вариантов формирования отчетов о проектных процедурах:

1. Отчет о проектных процедурах, находящихся в заданном состоянии (аналогичная задача выполняется программой *Listing*; см. выше).
2. Отчет о проектных процедурах, выполняемых указанным подразделением.
3. Отчет о проектных процедурах, не имеющих входных ссылок.
4. Отчет о проектных процедурах, не имеющих выходных ссылок.
5. Отчет о проектных процедурах, плановые даты окончания которых меньше заданной даты.
6. Отчет о проектных процедурах, плановые даты окончания которых больше заданной даты.
7. Отчет о рабочих состояниях проектных процедур для данного проекта.
8. Отчет о проектных процедурах, ожидающих входную информацию от указанной проектной процедуры.
9. Отчет об узлах объекта проектирования, разрабатываемых в рамках заданного проекта.

Выбор нужного отчета осуществляется путем ввода номера отчета. Затем в ответ на подсказку необходимо ввести дополнительную информацию, требующуюся для отбора информации из файла рабочей модели процесса проектирования.

Сформированный отчет можно отобразить на экране терминала, записать на диск в виде текстового файла или вывести на принтер.

Программа *Report* написана на языке *Фортран/МВС* [74, 75]. Файл выполняемого образа программы (REPORT.EXE) имеет размер приблизительно равный 49 Кбайтам.

Использование этой программы обеспечивает контроль проекта по различным аспектам. Кроме того, при передаче проекта в архив САПР выполняется тотальная проверка проектных процедур на их завершенность для данного проекта, что гарантирует логическую целостность передаваемых проектных данных. Другими словами, пока все проектные процедуры, реализующие данный проект, не будут локально завершены, передача проекта в архив не состоится.

3.4. Перепланирование и временная увязка проектных работ

Задача перепланирования и временной увязки различных работ над проектом связана с корректировкой календарных планов и обусловлена следующими причинами: включением дополнительных работ, ранее не планировавшихся; перерасходом планового времени выполнения КПП; досрочным окончанием КПП. Информационную поддержку в решении данной задачи обеспечивает программа отыскания критического, т.е. самого длинного, пути в орграфе.

Расчет длин путей в орграфе требует, чтобы каждая дуга была ассоциирована с некоторой положительной величиной, называемой весом дуги. Применительно к задаче перепланирования такой величиной является продолжительность выполнения работ по каждой проектной процедуре рабочей модели. Таким образом, при решении данной задачи весом каждой дуги орграфа рабочей модели является целое количество календарных дней необходимых для выполнения соответствующей проектной процедуры.

Поиск критического пути в ациклическом орграфе осуществляется на основе модифицированного алгоритма Дейкстры для отыскания кратчайшего пути с заменой веса каждой дуги на противоположное значение.

Структурно алгоритм состоит из трех следующих частей:

1. Инициализация матрицы смежности. Выполняется сканирование записей рабочей модели, вычисляется вес каждой дуги и заносится с противоположным знаком в соответствующий элемент матрицы. Следует отметить, что 1) в качестве веса дуги выбирается продолжительность выполнения проектной процедуры, представленной в орграфе вершиной, в которую заходит эта дуга; 2) веса всех дуг орграфа являются отрицательными числами.

2. Поиск кратчайшего пути в соответствии с алгоритмом Дейкстры [76, 77]. В орграфе задаются две вершины – начальная и конечная, между которыми осуществляется поиск кратчайшего пути. Затем в цикле выполняются следующие шаги:

Шаг 1. Определяются веса для дуг, исходящих из заданной вершины поиска во все остальные.

Шаг 2. Просматриваются веса исходящих дуг с целью отыскания наименьшего значения и запоминается новая вершина, в которую заходит дуга с наименьшим весом.

Шаг 3. Далее алгоритм добавляет этот наименьший вес к весам всех дуг, которые исходят из новой вершины поиска ко всем остальным.

Шаг 4. В цикле сравниваются полученные суммы с наименьшим весом и, если сумма оказывается меньше прежнего минимального веса, прежнее значение заменяется значением минимальной суммы.

Шаг 5. Затем новая вершина поиска удаляется из списка вершин, до которых еще не определены кратчайшие расстояния. Если новая вершина является заданной конечной вершиной, то поиск завершается.

3. Вывод абсолютного значения вычисленного суммарного веса дуг – длины критического пути (в календарных днях) и перечня вершин, через который проходят дуги образующие этот путь.

Данная задача выполняется с помощью программы *CriticalPath*, запускаемой выбором пункта “Длина критического пути” в меню “Работа с информационной моделью”. В основе программы, реализующей алгоритм, неформально описанный выше, лежит модифицированный вариант подпрограммы DIJKSTRA [77]. Программа написана на языке *Фортран/МБС* [74, 75]. Файл выполняемого образа программы (файл CRITIC.EXE) имеет размер приблизительно равный 42 Кбайтам.

3.5. Реализация отката в технологическом маршруте

Одной из важных задач оперативного управления является выполнение отката в технологическом маршруте проектирования. Процесс проектирования современных изделий электронной техники (ИЭТ) и МСВТ имеет выраженный итерационный характер. Каждой итерации предшествует “откат” к ранее выполненной КПП в технологическом маршруте. Откаты могут иметь различную “глубину”, которая определяется как объективными (например, сложность проектируемого изделия, наличие или отсутствие подходящих инструментальных средств проектирования и др.), так и субъективными факторами (например, опыт проектировщиков, квалифицированное руководство проектом, конкретный момент обнаружения некорректных проектных данных и др.) процесса проектирования. При проектировании сложных объектов нередко возникает необходимость выполнения “глубоких” откатов, когда допущенная на ранних этапах ошибка обнаруживается лишь на одном из заключительных этапов и требует возврата в какую-либо из ранее выполненных проектных процедур или даже в начало технологического маршрута. Задача реализации отката в технологическом маршруте проектирования сводится к одной из задач глобального анализа графов, а именно к задаче обхода вершин орграфа.

Наиболее распространенными методами, использующимися при решении задач глобального анализа графов, являются поиски в глубину и в ширину [76–79]. Метод поиска в глубину получил широкое распространение благодаря большей универсальности. Поиск в глубину – это регулярный обход вершин графа по следующим правилам:

1. Находясь в вершине x , нужно двигаться в любую другую, ранее не пройденную вершину y (если таковая найдется), одновременно запоминая дугу, по которой впервые попали в нее.

2. Если из вершины x не удастся попасть в ранее не пройденную вершину или таковой вообще нет, то необходимо вернуться в вершину z , из которой впервые попали в x , и продолжить поиск в глубину из вершины z .

Таким образом, при выполнении обхода вершин графа по этим правилам стремятся проникнуть в глубь графа так далеко, как это возможно, затем выполняют отход на шаг назад и снова стремятся пройти вперед и т.д.

Методу поиска в глубину можно в известном смысле противопоставить метод поиска в ширину, который предполагает последовательный просмотр всех вершин графа на уровне k , затем всех вершин на уровне $k+1$ и т.д.

Разработан итеративный алгоритм обхода ациклического орграфа, использующий вспомогательный список корневых вершин (под)деревьев, который является своеобразным “симбиозом” алгоритмов, реализующих методы поиска в глубину и ширину [80, 81]. В отличие от известного рекурсивного алгоритма поиска в глубину, предложенного Р. Тарьяном [82], данный алгоритм обладает двумя следующими преимуществами: 1) не требует наличия (или запоминания) обратных дуг для вершин орграфа, т.е. экономит оперативную память и/или дисковое пространство, и 2) сокращает время обработки вершин орграфа за счет отказа от возврата в корневые вершины поиска по обратным дугам.

Итеративный алгоритм выполняет регулярный обход вершин ациклического орграфа по следующим правилам:

1. Попав в некоторую вершину x , нужно проанализировать ее тип (по количеству исходящих дуг). Если вершина корневая, то занести ее в список корневых вершин (при условии, что она не была ранее занесена в этот список) и прекратить продвижение вглубь орграфа. Если вершина обычная, то продвигаться в другую, ранее не пройденную вершину y .

2. Если из вершины x не удастся попасть в ранее не пройденную вершину или таковой вообще нет, то необходимо выбрать очередную корневую вершину z из списка и продолжить поиск в глубину из этой вершины.

3. Если список корневых вершин оказывается исчерпанным, т.е. все обнаруженные корневые вершины орграфа обработаны, то алгоритм прекращает свою работу.

Укрупненное описание итеративного алгоритма для обхода вершин ациклического орграфа, приведено ниже.

1. НАЧАЛО. Проинициализировать список корневых вершин.
2. Получить вершину, с которой необходимо начать обход орграфа.
3. Пронумеровать эту вершину, т.е. присвоить ей номер $N = 1$.
4. Получить информацию о количестве исходящих дуг для вершины и занести ее в переменную *Cnt*.
5. Если $Cnt \neq 0$, выполнить переход на шаг 8.
6. Если список корневых вершин исчерпан, то КОНЕЦ.
7. Получить номер корневой вершины из списка и выполнить переход на шаг 4.
8. Если $Cnt > 1$, выполнить переход на шаг 11.
9. Перейти в следующую вершину по исходящей дуге.
10. Пронумеровать вершину, т.е. $N = N + 1$, и выполнить переход на шаг 4.
11. Если номер вершины N есть в списке корневых вершин, выполнить переход на шаг 14.
12. Занести номер вершины N в список корневых вершин.
13. Выбрать левую исходящую дугу и пронумеровать ее, т.е. $M = 1$, и выполнить переход на шаг 16.
14. Проинициализировать номер дуги, т.е. $M = 0$.
15. Пронумеровать следующую исходящую дугу, т.е. $M = M + 1$.
16. Если $Cnt < M$, установить указатель на следующий элемент списка корневых вершин и выполнить переход на шаг 6.
17. Перейти в следующую вершину по исходящей дуге с номером M .
18. Если эта вершина пронумерована, то выполнить переход на шаг 15.
19. Пронумеровать вершину, т.е. присвоить ей номер $N = N + 1$.
20. Получить информацию о количестве исходящих дуг для вершины и занести ее в переменную *Cnt2*.

21. Если $Cnt2 = 0$, выполнить переход на шаг 15.

22. Если $Cnt2 > 1$, занести номер вершины N в список корневых вершин и выполнить переход на шаг 15.

23. Перейти в следующую вершину по исходящей дуге и выполнить переход на шаг 18.

Структурная схема итеративного алгоритма для обхода вершин ациклического орграфа показана на рис. 3.1.

Данный алгоритм обеспечивает обход и нумерацию всех вершин ациклического орграфа. В качестве побочного результата от выполнения алгоритма можно рассматривать нумерацию исходящих дуг для каждой корневой вершины орграфа. Небольшая модификация приведенного выше алгоритма позволяет выполнить сквозную нумерацию дуг в ациклическом орграфе.

Следует также отметить, что хотя данный алгоритм разрабатывался в расчете на обработку ациклических орграфов, исходя из условий конкретной предметной задачи, стоящей перед авторами, после небольшой модификации он сможет корректно работать и в случае орграфов, содержащих контурные пути.

Действительно, в предложенном алгоритме одним из условий прекращения продвижения в глубину орграфа является обнаружение пронумерованной вершины, что исключает возможность многократной нумерации вершин. Всякий раз, когда обнаруживается пронумерованная вершина, осуществляется возврат в текущую вершину поиска для продолжения обхода орграфа. Если это корневая вершина, то для нее выбирается следующая исходящая дуга и обход продолжается по этой дуге. Если это вершина, которая имеет только одну исходящую дугу, то в описании алгоритма после шага 9 необходимо включить проверку того, что следующая вершина пронумерована. Эта проверка позволяет исключить “зацикливание” алгоритма, обеспечив переход на шаг 6 в случае обнаружения пронумерованной вершины.

Анализ эффективности выполнения рекурсивного и итеративного алгоритмов для двух граничных вариантов структуры орграфов (линейного орграфа и двоичного дерева) позволил получить следующие соотношения:

$$K_1 = (2N - 1) / N \text{ — для линейного орграфа;}$$

$$K_2 = (N + 2M) / (N + 2M - 1) \text{ — для двоичного дерева,}$$

где N – общее количество вершин орграфа, M – количество корневых вершин, K – коэффициент, показывающий отношение эффективностей рекурсивного и итеративного алгоритмов.

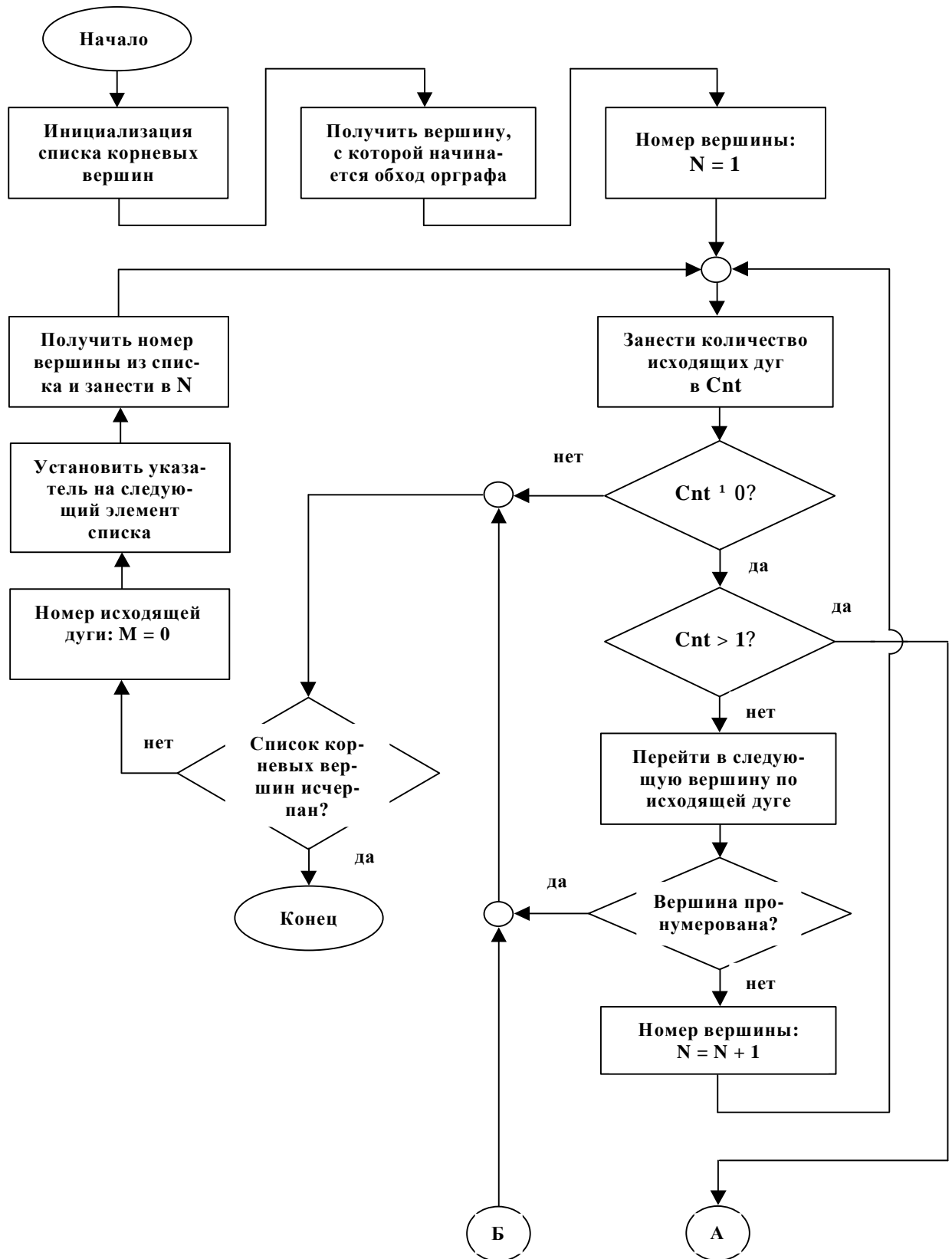


Рис. 3.1(а). Структурная схема итеративного алгоритма для обхода вершин ациклического орграфа

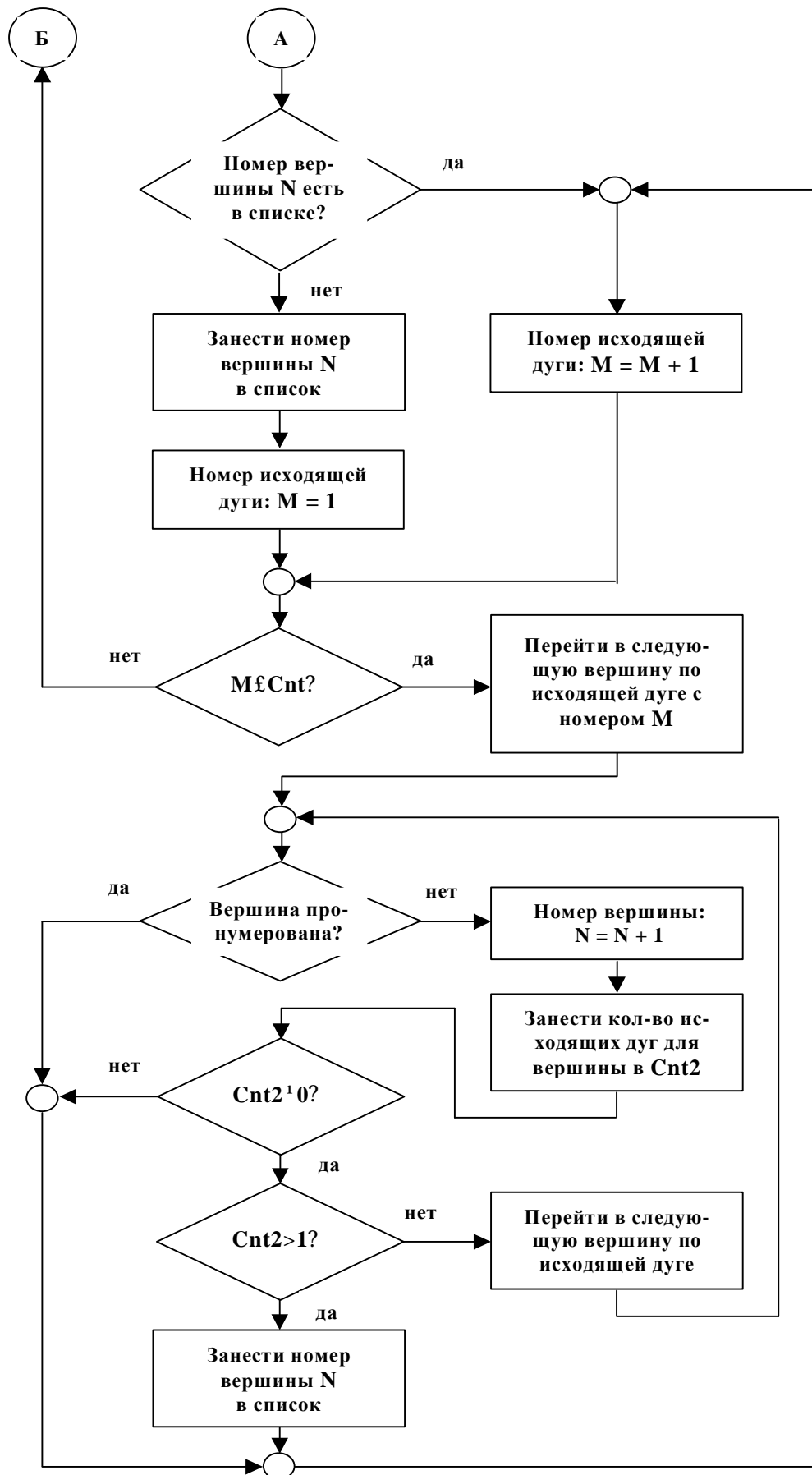


Рис. 3.1(б). Структурная схема итеративного алгоритма для обхода вершин ациклического орграфа (продолжение)

Для сравнительной оценки эффективностей выполнения указанных алгоритмов при обработке орграфов более сложной структуры, фрагментами которых являются как линейные графы, так деревья, был поставлен вычислительный эксперимент с привлечением ЭВМ. На IBM PC-совместимом компьютере в среде Турбо-Паскаль версии 6.0 разработан комплекс программ для работы с графами, включающий: программу для формирования и модификации структуры орграфа (граф-редактор), программы, реализующие рекурсивный и итеративный алгоритмы для обхода вершин ациклического орграфа соответственно.

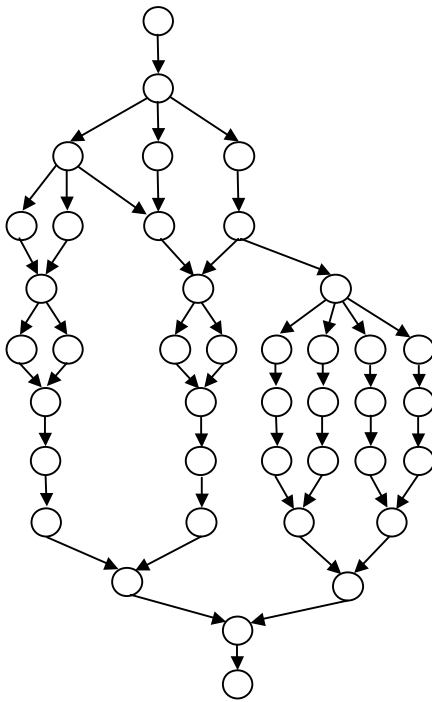


Рис. 3.2. Пример ациклического орграфа произвольной структуры

Обработка полутора десятков представлений орграфов произвольной структуры и последующие расчеты показали, что при выполнении обхода орграфа итеративный алгоритм выполняет в среднем в 1,5 раза меньше обращений к его вершинам, чем известный рекурсивный алгоритм. Например, для ациклического орграфа произвольной структуры, показанного на рис. 3.2, это отношение составляет приблизительно 1,8.

Приведенный выше итеративный алгоритм обхода вершин ациклического орграфа модифицирован применительно к задаче реализации глубокого отката в технологическом

маршруте проектирования [83]. Описание модифицированного итеративного алгоритма приведено ниже.

1. НАЧАЛО. Для вершины, соответствующей КПП, в процессе выполнения которой была обнаружена ошибка проектирования, сменить состояние “Активная” на “Приостановлена”.

2. Ввести идентификатор вершины, начиная с которой выполняется обход в орграфе, т.е. идентификатор проектной процедуры, сформировавшей некорректное проектное решение.

3. Инициализировать список идентификаторов корневых вершин (под)деревьев орграфа.
4. Прочитать запись, относящуюся к указанной вершине, из файла рабочей модели.
5. Получить информацию о состоянии вершины.
6. Если вершина имеет состояние “Локально завершена”, то переход на шаг 9.
7. Если список корневых вершин исчерпан, то КОНЕЦ.
8. Получить очередной идентификатор вершины из списка корневых вершин, установить указатель на следующий элемент и переход на шаг 4.
9. Сменить состояние “Локально завершена” у вершины на “Пассивная”.
10. Получить информацию об ответственном исполнителе данной проектной процедуры и послать ему оповещение о смене состояния и о причине этого.
11. Получить содержимое счетчика выходных ссылок для данной вершины (*OutCnt*).
12. Если *OutCnt* = 1, т.е. вершина не является корнем (под)дерева, получить выходную ссылку (идентификатор вершины) и выполнить переход на шаг 4.
13. Если идентификатор данной вершины отсутствует в списке корневых вершин, поместить идентификатор в список корневых вершин.
14. Инициализировать номер выходной ссылки: $i = 1$.
15. Если $i > OutCnt$, то переход на шаг 7.
16. Получить i -ю выходную ссылку, т.е. идентификатор i -го потомка данной вершины.
17. Прочитать запись, которая относится к вершине, указанной выходной ссылкой (идентификатор этой вершины используется в качестве ключа поиска в файле рабочей модели).
18. Получить содержимое счетчика выходных ссылок для данной вершины (*Cnt*).
19. Если *Cnt* > 1, т.е. это корневая вершина, то поместить идентификатор этой вершины в список идентификаторов корневых вершин. Иначе – переход на шаг 21.

20. Увеличить номер (индекс) выходной ссылки на 1, т.е. $i = i + 1$, и выполнить переход на шаг 15.

21. Получить информацию о состоянии вершины.

22. Если вершина имеет состояние “Пассивная” или “Приостановлена”, то переход на шаг 20.

23. Сменить состояние “Локально завершена” у вершины на “Пассивная”.

24. Получить информацию об ответственном исполнителе данной проектной процедуры и послать ему оповещение о смене состояния и о причине этого.

25. Получить выходную ссылку и выполнить переход на шаг 17.

Реализация приведенного выше модифицированного алгоритма обеспечивает поиск всех КПП, на вход которых поступили некорректные КПР, проверку и смену их рабочих состояний, рассылку соответствующих оповещений ответственным исполнителям этих КПП.

Задача реализации отката в технологическом маршруте проектирования выполняется выбором пункта “Откат в технологическом маршруте” в меню “Работа с информационной моделью”. Далее в диалоге задается уточняющая информация, необходимая для решения этой задачи (в частности, указывается идентификатор проектной процедуры, начиная с которой необходимо осуществить поиск в глубину в орграфе рабочей модели процесса проектирования).

Первоначально для реализации отката в технологическом маршруте проектирования была разработана программа *OtkatR*, реализующая известный рекурсивный алгоритм. Программа написана на языке *Bliss-32* [72, 73]. Файл выполняемого образа программы (файл ОТКАТР.EXE) имеет размер приблизительно равный 17 Кбайтам. Впоследствии была разработана программа *OtkatI*, реализующая описанный выше модифицированный итеративный алгоритм. Программа написана на языке программирования *Паскаль/MBC* (расширение стандартного Паскаля) [84, 85]. Файл выполняемого образа программы (файл ОТКАТИ.EXE) имеет размер приблизительно равный 39 Кбайтам.

Глава 4. Состав, структура и функциональные возможности мониторинной системы

Большой объем и сложность задач проектирования современных ИЭТ и МСВТ, обусловленные их высокой функциональной сложностью, с одной стороны, и многоаспектным (а также многоуровневым – в рамках каждого аспекта) характером процесса проектирования, с другой стороны, предопределяют выбор распределенной архитектуры интегрированной САПР.

В свою очередь, распределенный характер обработки проектной информации в интегрированной САПР обуславливает использование двухуровневой распределенной системы управления, представленной центральной и локальными мониторинными системами.

В данной главе описаны состав, структура и функциональные возможности, а также сходные свойства и отличительные признаки каждого из двух типов МС, реализованных в интегрированной САПР. Выполнено позиционирование в структуре центральной МС базовых компонентов, рассмотренных в предыдущих главах монографии. Рассмотрена унифицированная схема информационного взаимодействия проектирующих подсистем интегрированной САПР, обеспечиваемая локальной и центральной МС совместно с информационной системой, и реализующая основные положения концепции сквозного проектирования. Кроме того, приведена схема взаимодействия двух обслуживающих подсистем САПР – архивной и информационной систем – при передаче завершенных проектных решений в архив САПР.

4.1. Общие сведения об интегрированной САПР МСВТ

Техническая база интегрированной САПР МСВТ представлена совокупностью объединенных в локальную вычислительную сеть (ЛВС) ИГС и АРМ, основу которых составляют отечественные 32-разрядные мини- и микроЭВМ семейства “Электроника”: “Электроника 82” (УВК 8000-006, УВК 8000-007), “Электроника МС0104” и “Электроника МС0107” [86–92]. Дополнительно в составе АРМ логического проектирования могут быть использованы спецпроцессоры “Электроника МС90” и “Электроника МС91” – ускорители логического

моделирования (УЛМ), аппаратно реализующие двухпроходный событийный алгоритм моделирования [92, 93]. Технические средства и системное программное обеспечение 32-разрядных ИГС и АРМ обеспечивают многопользовательский и мультипрограммный режим работы [92].

Прототипом ЛВС для интегрированной САПР послужила локальная сеть магистральной структуры типа Ethernet. В ЛВС данного типа максимальная пропускная способность сетевого канала связи, в качестве которого используется “толстый” коаксиальный кабель (10Base5 Ethernet), составляет 10 Мбит/с. Для подключения 32-разрядных мини- и микроЭВМ к ЛВС используются сетевые контроллеры, разработанные специалистами СКТБ “Светлана” (г. Санкт-Петербург) и ОКБ при заводе “Процессор” (г. Воронеж) на основе прототипных образцов контроллеров DEUNA и DEQNA фирмы DEC [94].

Программные средства интегрированной САПР представлены рядом ППП проектирования, а именно:

прикладной интерактивный графический пакет “ИГРА”, предназначенный для подготовки информации (текстовых и графических описаний библиотечных элементов и схем на языках БИМОД, МОС-94 и SDL) для проектирующих подсистем САПР [92, 95–100];

прикладной пакет поведенческого моделирования “ПРИАМ”, предназначенный для иерархического (многоуровневого) моделирования алгоритма функционирования блоков и узлов цифровых устройств, описанных на языках АЛОС, NHDL и SDL [92, 101, 102];

прикладной пакет логического моделирования “ПРАЦИС”, предназначенный для логико-временной верификации цифровых устройств и генерации тестовых последовательностей. Существуют несколько разновидностей этого пакета (ПРАЦИС-Т, ПРАЦИС-ТМ, ПРАЦИС-100), различающиеся функциональными возможностями [92, 103–105];

прикладной пакет ускоренного моделирования аппаратуры “ПУМА”, предназначенный для трансляции исходного описания схемы во внутренний формат системы ускоренного моделирования (СУМ), формирования БД СУМ, загрузки описания в УЛМ, управления процессом логического моделирования, приема результатов моделирования из СУМ и их обработки [92, 100];

прикладные пакеты топологического проектирования: “ТЕРЕК” (для проектирования топологии коммутационных слоев БИС на БМК, проектируемых по КМОП-технологии), “АИДА” (для проектирования топологии СБИС на БМК, проектируемых по биполярной технологии), “СИАТ-ИНГРЕД” (для проектирования топологии МПП) [106–108];

пакет прикладных программ “АСКОР” для корректировки схем по результатам топологического проектирования [92];

пакет схемотехнического проектирования “ПРАНИС-К”, предназначенный для моделирования цифровых устройств на уровне физических элементов [92];

прикладной пакет подготовки конструкторской документации (КД) в соответствии с требованиями ЕСКД, обеспечивающий ввод с терминала данных для КД, автоматическую генерацию отчетных форм КД, вывод КД на экран терминала или на построочно-печатающее устройство в форматах А3 и А4 [109].

Перечисленные выше программные средства представляют основу для организации различных технологических маршрутов проектирования МСВТ. Однако, создание интегрированной САПР путем комплексирования имеющихся программных инструментальных средств, часть из которых разработана на различной концептуальной основе, т.е. без учета их совместного использования, требует решения ряда важных задач.

Первоочередной и наиболее очевидной задачей является устранение несоответствий в форматах проектных данных, т.е. задача обеспечения информационного согласования между различными инструментальными средствами. Для ее решения обычно выполняется доработка используемых ППП путем включения в их состав специальных интерфейсных программ (различных конверторов, пре- и постпроцессоров форматов данных).

Другой важной задачей, возникающей при организации сквозных маршрутов проектирования в интегрированной САПР, является автоматизация информационных обменов между проектирующими подсистемами.

Не столь очевидными, но не менее важными, являются следующие задачи: организация управления прохождением проектов по различным технологическим маршрутам проектирования, координация действий различных групп проектировщиков, занятых в работе над проектом, мониторинг текущего со-

стояния проекта [110]. Для автоматизации решения этих и других задач управления процессом проектирования в условиях мультипроектного режима работ разработана мониторинговая система – одна из обслуживающих подсистем интегрированной САПР, выполняющая функции системы управления. Ниже описаны структура мониторинговой системы и функциональные возможности каждой ее подсистемы.

4.2. Структура системы управления интегрированной САПР МСВТ

В соответствии с распределенным характером процесса проектирования в интегрированной САПР МСВТ реализована двухуровневая распределенная система управления (СУ), которая представлена центральной (ЦМС) и локальными (ЛМС) мониторинговыми системами, как показано на рис. 4.1.

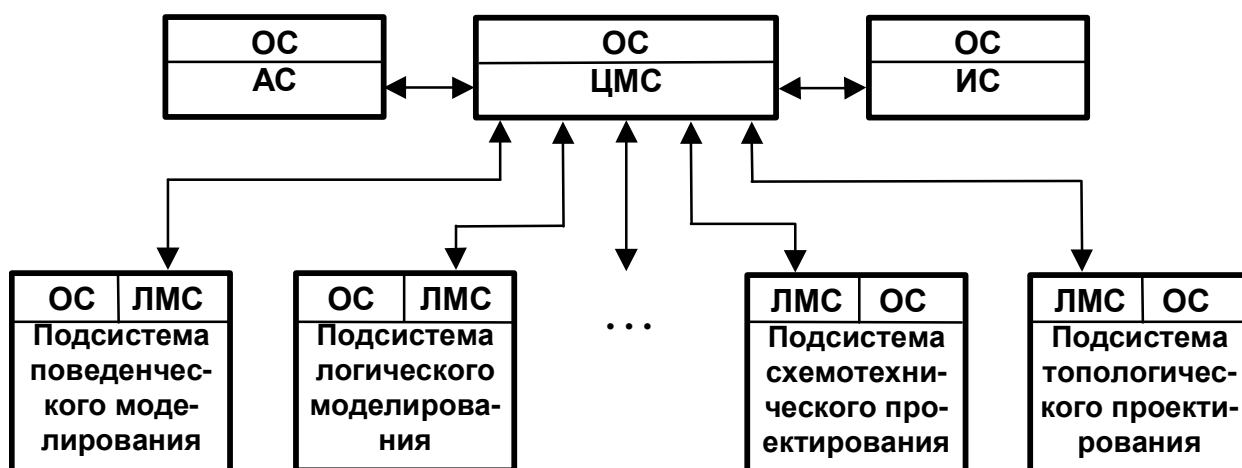


Рис. 4.1. Двухуровневая распределенная система управления интегрированной САПР

ЦМС функционирует на выделенной 32-разрядной мини-ЭВМ “Электроника 82”, связанной сетевым каналом с совокупностью 32-разрядных АРМ и ИГС. ЦМС включает в себя компоненты методического, лингвистического, программного и информационного обеспечения, реализованные в следующих подсистемах (рис. 4.2):

- генерации;
- диалога;
- регистрации;

справочной;
информационной;
управления;
контроля;
защиты;
взаимодействия;
связи.

Каждая ЛМС, функционирующая на проблемно-ориентированном АРМ или ИГС, содержит перечисленные выше подсистемы, за исключением подсистемы управления и информационной подсистемы, и управляет работой соответствующей проектирующей подсистемы.

4.3. Функциональные возможности мониторной системы

Базовые компоненты МС реализуют инвариантные процедуры управления процессом автоматизированного проектирования, обеспечивающие организацию конвейерной обработки проекта и информационного взаимодействия подсистем интегрированной САПР, мониторинг текущего состояния проекта и контроль логической целостности проекта, защиту проектной информации при ее хранении и передаче, информационную и инструментальную среды для реализации руководителем проекта функций оперативного управления, организацию диалога с пользователями МС на основе иерархической системы меню, защиту АРМ и ИГС, входящих в состав интегрированной САПР, от несанкционированных действий пользователей и ряд других функций.

Функциональное назначение каждой подсистемы МС интегрированной САПР описано ниже.



Рис. 4.2. Структурная схема центральной мониторинг системы

4.3.1. Подсистема генерации

Подсистема генерации обеспечивает выполнение следующих действий:

установку (инсталляцию) МС с дистрибутивных носителей (магнитная лента, набор дискет) на жесткий магнитный диск 32-разрядной мини- или микроЭВМ;

настройку МС на решение определенного круга задач проблемной (предметной) области;

восстановление работоспособного состояния МС при возникновении особых ситуаций.

Установка МС производится с дистрибутивных носителей, содержащих файлы МС в специальном упакованном формате – формате наборов сохранения программы BACKUP [68, 111]:

дистрибутивная магнитная лента (метка тома “CMS”) или дистрибутивный набор дискет (метка многотомного набора “CMS”) с файлами ЦМС;

дистрибутивная магнитная лента (метка тома “LMS”) или дистрибутивный набор дискет (метка многотомного набора “LMS”) с файлами ЛМС.

В ходе установки производится настройка на конкретную конфигурацию технических средств ЭВМ путем задания имени НМЛ или НГМД (для передачи контейнеров на МЛ или на дискетах соответственно – как страховочный вариант на случай выхода ЛВС из строя) и номера компьютера, однозначно идентифицирующего компьютер в составе интегрированной САПР.

В процессе установки МС создается командный файл SY\$MANAGER:MSTARTUP.COM, который обеспечивает назначения логических имен МС. Для того, чтобы этот файл выполнялся при загрузке операционной системы, команда для его запуска включается в стартовый командный файл системы SY\$MANAGER:SYSTARTUP.COM.

В результате выполнения командного файла MSTARTUP.COM в таблицу логических имен системы заносятся следующие системные имена:

MS\$CMS	Узел ЛВС, на котором размещается ЦМС.
MS\$CMSRWORK	Рабочий справочник, используемый подсистемой взаимодействия ЦМС (см. ниже).
MS\$COM	Справочник, содержащий командные файлы.

MS\$COMPUTER	Номер компьютера (от 0 до 99), идентифицирующий компьютер в интегрированной САПР.
MS\$DIR	Корневой каталог МС.
MS\$DISK	Диск, на котором размещена МС.
MS\$FLOPPY	Накопитель на гибких магнитных дисках (НГМД), используемый для обмена контейнерами на ГМД.
MS\$FMP	Справочник, содержащий файлы рабочих моделей процесса проектирования.
MS\$HLB	Справочник, содержащий библиотеки справочной информации.
MS\$IMAGE	Справочник, содержащий файлы выполнимых образов программ МС САПР.
MS\$JNL	Справочник, содержащий файлы регистрационной и учетной информации.
MS\$LIB	Справочник, содержащий библиотеки МС САПР.
MS\$RWORK	Справочник для приема контейнеров от ИС САПР.
MS\$TAPE	Накопитель на магнитной ленте (НМЛ), используемый для обмена контейнерами на МЛ.
MS\$WHERE	Идентифицирует МС как центральную (CENTR) или локальную (LOCAL).
MS\$WRKCTN	Справочник, используемый подсистемой взаимодействия МС САПР.

Кроме того, в таблицу команд системы заносятся следующие две команды:

MONSYS (*MONitor SYStem* – *мониторная система*) – команда запуска мониторной системы;

LEI (*Load and Execute Image* – *загрузить и выполнить образ*) – команда запуска аттестованных программных средств интегрированной САПР.

При установке МС в файл полномочий пользователей операционной системы (SYS\$MANAGER:SYSUAF.DAT) заносится запись о пользователе с именем MONSYS, которая в дальнейшем будет использоваться при обмене контейнерами через ЛВС.

Настройка МС на решение задач определенной предметной области выполняется администратором мониторной системы. Она заключается в выполнении аттестации инструментальных программных средств САПР, предназначенных для решения этих задач, и создании иерархической системы меню для запуска соответствующих программ. Для аттестации инструментального ПО авторами разработаны программы, выполняющие модификацию структуры заголовков файлов выполнимых образов программ, что предотвращает запуск программ обычным способом, принятым в системе (по команде RUN); запуск программ с помощью команды LEI (с предварительным восстановлением первоначальной структуры заголовка файлов образов, вывод журнала аттестованных программных средств (файл MS\$JNL:IMAGES.JNL) на экран терминала, на принтер или на то и на другое одновременно.

Восстановление работоспособного состояния МС требуется в случае нарушения целостности информации МС, которое может возникнуть в результате аппаратных сбоев или серьезных ошибок ПО. Обеспечить целостность информации МС позволяет регулярное создание ее резервных копий на магнитных носителях (магнитной ленте, дискетах или жестком магнитном диске).

4.3.2. Подсистема диалога

Подсистема диалога предназначена для организации диалога с пользователями МС и обеспечивает следующие возможности:

- построение иерархической системы меню, описывающей выполняемые в каждой подсистеме САПР работы;

- ввод дополнительных реквизитов в процессе диалога для передачи их управляющей программе и использовании при вызове соответствующих процедур;

- оперативный доступ к информации, содержащейся в справочной библиотеке (файл MS\$HLB:MSHELP.HLB), в процессе ведения диалога с пользователем.

4.3.3. Подсистема регистрации

Подсистема регистрации предназначена для выполнения следующих функций:

- ввод нового пользователя МС, т.е. создание новой записи в файле полномочий пользователей МС;

- удаление пользователя МС, т.е. удаление соответствующей записи из файла полномочий пользователей МС;

- модификация содержимого записи файла полномочий пользователей МС (обеспечивается возможность модификации всех полей записи за исключением поля учетного имени пользователя);

- вывод листинга файла полномочий пользователей МС;

- регистрация попыток несанкционированного доступа к МС;

- регистрация времени входа в МС и времени выхода из нее, вида и кодов завершения работ пользователей МС;

- регистрация информационных обменов между подсистемами (обмен информацией выполняется не напрямую между подсистемами, а через ИС САПР);

- регистрация сертифицированных программных средств интегрированной САПР.

В записи файла полномочий пользователей (файл MS\$JNL:CMSUAF.DAT – для центральной МС, MS\$JNL:LMSUAF.DAT – для локальной МС) содержатся следующие данные:

- учетное (входное) имя пользователя, длина которого не превышает 6 символов;

- номер категории пользователя, представленный целым числом от 0 до 15 (чем больше номер категории, тем меньше полномочий);

- пароль, который пользователь должен использовать при входе в МС (строка длиной не менее 3 и не более 8 символов);

- предельное количество попыток ввода неверного пароля перед блокированием входа в МС;

- фамилия, имя и отчество пользователя (строка длиной не более 50 символов);

- символьный код проектирующего подразделения (строка длиной не более 50 символов);

- код подсистемы (целое число от 0 до 32);

имя узла ЛВС, выделенного для работы пользователя.

Работу с подсистемой регистрации выполняет администратор мониторинг системы в интерактивном режиме.

4.3.4. Справочная подсистема

Справочная подсистема предназначена для выполнения следующих функций:

- вывод на экран терминала или на принтер информации о состоянии проектирования заданного функционального узла, совокупности узлов или объекта проектирования в целом (мониторинг текущего состояния проекта);

- вывод на экран терминала и на принтер содержимого справочников МС;

- вывод на экран терминала и на принтер содержимого журнальных файлов, формируемых подсистемой регистрации МС;

- реализация механизма оповещения пользователей МС (доставка системных оповещений, прием уведомлений о получении оповещений адресатами).

4.3.5. Информационная подсистема

Информационная подсистема обеспечивает прием, хранение, выдачу и сопровождение следующих информационных массивов САПР:

- рабочая модель процесса проектирования;

- файл восстановления информации после системных сбоев;

- различные журналы регистрации;

- файл полномочий пользователей МС;

- контейнеры с проектными решениями и сопроводительной информацией перед их передачей в ИС.

4.3.6. Подсистема управления

Подсистема управления предназначена для управления процессом проектирования в интегрированной САПР и обеспечивает следующие возможности:

- выполнение генерации рабочей модели процесса проектирования по исходному описанию информационной модели, представленному на ЯОМ;

выполнение оперативной корректировки рабочей модели процесса проектирования;

вывод листинга рабочей модели процесса проектирования на экран терминала и на принтер.

Оперативная корректировка рабочей модели процесса проектирования выполняется в интерактивном режиме и включает в себя следующие операции:

добавление проектной процедуры в рабочую модель процесса проектирования;

удаление проектной процедуры из рабочей модели процесса проектирования;

модификация реквизитов проектной процедуры в модели процесса проектирования (в частности, обеспечивается возможность изменения рабочего состояния проектной процедуры, т.е. перевода проектной процедуры в состояние “пассивна” или “ожидания”, активизации проектной процедуры, находящейся в состоянии ожидания, приостановки выполняющейся процедуры);

добавление ссылок на проектную процедуру в рабочей модели процесса проектирования;

удаление ссылок на проектную процедуру в рабочей модели процесса проектирования.

Кроме того, подсистема управления позволяет выполнить поиск в рабочей модели “изолированных” вершин, т.е. проектных процедур с пустой областью ссылок, и контурных путей, образованных вследствие циклических ссылок проектных процедур. Обе эти ситуации могут возникнуть вследствие некорректно выполненной модификации рабочей модели процесса проектирования.

4.3.7. Подсистема контроля

Подсистема контроля обеспечивает руководителя проекта инструментальными средствами для контроля за ходом выполнения проектных работ. Эти средства позволяют выполнять следующие функции:

поиск в рабочей модели процесса проектирования проектных процедур с просроченной датой окончания работ;

поиск в рабочей модели процесса проектирования проектных процедур с неверной входной информацией (это требуется в случае, когда обнаруживается

ошибка в переданном проектном решении и необходимо выполнить “откат” в маршруте; при этом проектные процедуры, активизированные данным проектным решением должны быть переведены в пассивное состояние или состояние ожидания);

вывод содержимого журнала учета корректировок рабочей модели процесса проектирования (файл MS\$JNL:CORJNL.IDX) на экран терминала и на принтер;

вывод содержимого журнала учета времени работы пользователей в МС (файл MS\$JNL:TIMUSR.IDX) на экран терминала и на принтер;

вывод содержимого журнала учета видов работ в МС и кодов их завершения (файл MS\$JNL:WRKCOD.IDX) на экран терминала и на принтер;

вывод содержимого журнала учета идентификаторов проектов (файл MS\$JNL:IDOPRJ.IDX) на экран терминала и на принтер;

вывод содержимого журнала учета смены состояний проектных процедур рабочей модели процесса проектирования (файл MS\$JNL:STATUS.IDX) на экран терминала и на принтер;

вывод содержимого журнала учета системных оповещений (файл MS\$JNL:SYSMSG.IDX) на экран терминала и на принтер;

вывод содержимого журнала учета контейнеров, переданных в ИС интегрированной САПР (файл MS\$JNL:CONTRS.IDX), на экран терминала и на принтер;

вывод содержимого журнала учета модифицированных контейнеров (файл MS\$JNL:MODCNTR.IDX) на экран терминала или на принтер;

вывод содержимого журнала учета удаленных контейнеров (файл MS\$JNL:DELCNTR.IDX) на экран терминала и на принтер.

4.3.8. Подсистема защиты

Подсистема защиты обеспечивает выполнение следующих функций:

защиту на системном уровне от сбоев оборудования;

защиту от несанкционированного доступа, включающую:

- 1) обеспечение доступа пользователей интегрированной САПР к различным ее подсистемам в зависимости от номера категории пользователя;

- 2) использование средств защиты, имеющихся в операционной системе;
- 3) разрешение доступа к средствам МС в соответствии с информацией файла полномочий пользователей МС;
- 4) использование методов криптографических преобразований информации;
- 5) ограничение физического доступа к программному обеспечению МС интегрированной САПР путем размещения ЦМС на отдельной ЭВМ;
- 6) замену кодов идентификации информационных массивов проектирующих подсистем интегрированной САПР на внутренние коды МС при передаче информации в ИС САПР.

4.3.9. Подсистема взаимодействия

Подсистема взаимодействия предназначена для обеспечения информационного взаимодействия между подсистемами САПР, которое выражается в передаче контейнеров с проектными решениями. Как отмечалось выше, контейнеры являются основной единицей информационного обмена в интегрированной САПР.

Для организации контейнерного обмена подсистема взаимодействия ЦМС обеспечивает следующие возможности:

- создание и редактирование файла описания проектного решения;
- передача контейнера в ИС САПР с использованием ЛВС;
- передача контейнера в ИС с использованием МЛ;
- передача контейнера в ИС с использованием ГМД;
- запись информации о передаче контейнера в журнал учета переданных контейнеров;
- замена контейнера в ИС с использованием ЛВС;
- замена контейнера в ИС контейнером, размещенным на МЛ;
- замена контейнера в ИС контейнером, размещенным на ГМД;
- запись информации о модификации контейнера в журнал учета модифицированных контейнеров;
- прием контейнера от ИС с использованием ЛВС;

- прием контейнера от ИС на МЛ;
- прием контейнера от ИС на ГМД;
- удаление контейнера из ИС;
- запись информации об удалении контейнера в журнал учета удаленных контейнеров.

Более подробно функциональные возможности подсистемы взаимодействия описаны ниже.

4.3.10. Подсистема связи

Подсистема связи обеспечивает выполнение следующих функций:

- передачу информационных массивов САПР по каналам связи и на магнитных носителях информации (магнитных лентах и магнитных дисках);
- передачу управляющей информации и запросов подсистем интегрированной САПР.

4.4. Взаимодействие проектирующих подсистем САПР

Взаимодействие проектирующих подсистем в интегрированной САПР осуществляется путем передачи информации. Эта передача выражает причинно-следственные связи между комплексными проектными процедурами технологического маршрута проектирования. По характеру зарождения, использования и передачи информация делится на две группы:

- информация, необходимая для использования только в данной комплексной проектной процедуре;
- информация, совместно используемая несколькими комплексными проектными процедурами.

Как отмечалось в Главе 1, минимальным объектом управления для мониторинговой системы является комплексная проектная процедура. В этой связи при рассмотрении информационного взаимодействия проектирующих подсистем учитывается только информация, совместно используемая разными комплексными проектными процедурами.

Информационное взаимодействие проектирующих подсистем, реализующих функции поведенческого и функционально-логического моделирования,

генерации тестовых последовательностей и топологического проектирования в интегрированной САПР МСВТ (см. выше), основано на передаче файлов, содержащих проектные решения и пользовательские библиотеки элементов. Учитывая высокую сложность и разнородность подсистем интегрированной САПР, а также противоречивость требований к информационному обеспечению, в качестве основной единицы информационного обмена между подсистемами выбран файл специального вида, называемый контейнером.

Мониторная система, как основной механизм, через который происходит информационное взаимодействие подсистем интегрированной САПР, должна удовлетворять следующим требованиям:

1. Передача проектных решений из одной проектирующей подсистемы в другую должна осуществляться только через мониторную систему, которая обязана фиксировать следующие данные, характеризующие передачу:

имя пользователя, передавшего файл;

дату и время передачи файла;

номер версии файла;

идентификационную информацию, характеризующую файл.

2. Вне зависимости от типов и содержимого файлов, порождаемых проектирующими подсистемами, передача должна осуществляться стационарным образом.

3. Программные средства интегрированной САПР с точки зрения МС должны рассматриваться как набор аттестованных данных.

Концепция технического обеспечения, принятая при построении интегрированной САПР МСВТ, предусматривает соединение АРМ и ИГС, реализующих решение предметных задач проектирования, с мини-ЭВМ верхнего уровня, содержащей ЦМС, линиями связи. Апробированы следующие виды межмашинных соединений:

1) асинхронная линия связи, реализованная с помощью интерфейса “Стык 2” (RS-232);

2) канал параллельной передачи данных, реализованный с помощью интерфейсов прямого доступа к памяти СК-19/СК26;

3) скоростной канал связи, реализованный с помощью средств локальной сети Ethernet.

Первые два вида связи обеспечивают скорость передачи данных от нескольких десятков до нескольких сотен килобайт в секунду, что недостаточно для реализации информационных обменов между подсистемами современных САПР. Кроме того, эти виды связи не обеспечивают надежной защиты от ошибок, возникающих в результате различных помех в каналах связи при передаче данных. Наконец, эти виды связи обеспечивают соединение типа “точка-точка”, т.е. коммуникационный канал рассчитан только на две ЭВМ. В силу указанных недостатков двух первых видов соединений для межмашинной связи выбрана локальная вычислительная сеть, хотя и более дорогостоящая, но обеспечивающая достаточно высокую скорость (максимально 10 Мбайт/с) и высокую надежность передачи данных.

4.5. Информационная система интегрированной САПР

Информационная система (ИС) – это обслуживающая подсистема, обеспечивающая информационное взаимодействие прочих подсистем, входящих в состав интегрированной САПР МСВТ. Информационное взаимодействие подсистем, выражающееся в унифицированном обмене информацией, осуществляется в рамках единых организационных и методологических требований:

обмен информацией выполняется не напрямую между подсистемами интегрированной САПР, а через ИС;

инициатором информационного обмена выступает проектирующая или обслуживающая подсистема интегрированной САПР, которая адресует свой запрос к ИС через МС;

единицей обмена является информационный контейнер, имеющий строго определенную структуру, представленную описательной и содержательной частями;

для упорядочения доступа к контейнеру организуется очередь заявок, которая обслуживается в соответствии с дисциплиной FIFO (*first in, first out* – первым пришел, первым вышел), т.е. обслуживание выполняется в порядке поступления запросов от подсистем;

изменение информации в содержательной части контейнера разрешается только с санкции руководителя проекта;

уведомление об изменении в содержательной части контейнера автоматически рассылается всем потребителям информации данного контейнера, т.е. ответственным исполнителям всех проектных процедур, информационно-зависимых от проектной процедуры, проектное решение которой подверглось корректировке.

4.5.1. Понятие контейнера

Контейнер – это унифицированная информационная структура, состоящая из двух неоднородных частей: описательной части контейнера и собственно проектных данных.

Описательная часть контейнера. Описательная часть контейнера, так называемый файл описания проектного решения (ФОПР), условно делится на три части:

- описательная часть, формируемая автором проектного решения;
- описательная часть, формируемая мониторной системой;
- описательная часть, формируемая информационной системой.

Краткое описание каждой из этих частей представлено ниже.

Описательная часть, формируемая автором. При формировании контейнера с проектным решением его автор (обычно это ответственный исполнитель комплексной проектной процедуры) должен занести в контейнер информацию, однозначно идентифицирующую этот контейнер и обеспечивающую учет его движения в интегрированной САПР. Эта информация содержит следующие сведения:

- идентификатор ЭВМ;
- идентификатор проектирующей подсистемы САПР;
- идентификатор проекта;
- идентификатор комплексной проектной процедуры;
- идентификатор автора (имя ответственного исполнителя);
- идентификатор функционального узла в проекте;
- идентификатор типа проектного решения;
- тип носителя, назначенного для хранения контейнера (магнитный диск, магнитная лента и т.д.);
- дата формирования контейнера;

спецификации файлов, содержащих собственно проектные данные; резервные поля (для обеспечения возможности развития системы).

Перечисленная выше информация вводится автором проектного решения (контейнера) с помощью интерактивного редактора ФОПР, входящего в состав сервисных средств МС. Формирование и/или редактирование ФОПР выполняется с помощью специальных экранных форм и ряда простых команд редактирования, позволяющих выполнять позиционирование курсора в полях ввода экранной формы, ввод и удаление (забой) символов, завершение работы с редактором ФОПР.

Описательная часть, формируемая МС. Описательная часть, формируемая МС интегрированной САПР, включает в себя следующие элементы:

данные паспортизации (идентификатор контейнера, контрольные суммы записей ФОПР, файлов данных и контейнера в целом);

резервные поля (для обеспечения возможности развития системы).

Описательная часть, формируемая ИС. Описательная часть, формируемая ИС интегрированной САПР в момент приема контейнера, а также на протяжении всего времени хранения его в хранилище контейнеров ИС, включает в себя следующие элементы:

дата приема контейнера в ИС;

ссылки на записи в журнальных файлах учета выдачи контейнеров и учета изменений, выполненных в контейнерах;

резервные поля (для обеспечения возможности функционального расширения системы).

Часть контейнера, содержащая собственно проектные данные. Эта часть контейнера образована файлами данных, представляющими проектное решение. Внутренний формат проектных данных в общем случае специфичен для каждой проектирующей подсистемы интегрированной САПР. Любая из проектирующих подсистем, принимающая контейнер для использования содержащихся в нем проектных данных, должна обеспечить преобразование этих данных в свой внутренний формат.

Физический формат контейнера. Физически информационный контейнер в интегрированной САПР МСВТ представляет собой файл в формате

набора сохранения утилиты BACKUP операционных систем МВС и МОС-32М [111].

4.5.2. Информационный фонд ИС

В интегрированной САПР создается и используется множество информационных объектов. ИС распознает и обрабатывает лишь часть из них. Все обрабатываемые ИС информационные объекты делятся на две категории:

глобальные объекты, являющиеся общими для всех проектов (списки и библиотеки);

локальные объекты (описание проекта и проектные решения, выработанные в рамках данного проекта).

Глобальные данные и их подготовка к использованию в ИС. Глобальными данными ИС являются списки и библиотеки. К спискам в ИС интегрированной САПР относятся:

список подсистем, содержащий идентификаторы и развернутые названия подсистем интегрированной САПР, зарегистрированных в ИС как потенциальные потребители ее информационных ресурсов;

список типов проектных решений, содержащий идентификаторы и развернутые названия типов проектных решений, которые могут быть выработаны проектными процедурами в проектирующих подсистемах интегрированной САПР;

список проектов, содержащий идентификаторы и развернутые названия проектов, которым ИС обеспечивает информационную поддержку.

Подготовку списков для ИС выполняют пользователи интегрированной САПР. Формирование списков на магнитном носителе выполняет администратор ИС (после их утверждения руководителем САПР) с помощью соответствующего программного инструментария, входящего в состав ИС.

К библиотекам в ИС интегрированной САПР относятся:

библиотека типовых проектных решений, содержащая описания всех принятых типовых проектных решений (ТПР) и аттестованных библиотечных элементов (АБЭ);

библиотека аттестованных программных средств (АПС), содержащая описания всех программных средств, реализующих КПП или проектные процедуры и проектные операции в рамках каждой КПП.

Формирование библиотек, а также включение в их состав новых ТПР, АБЭ и АПС, выполняется только с санкции руководителя САПР. Включение нового объекта в соответствующую библиотеку выполняется в два этапа. На первом этапе в ИС передается лишь описание объекта и администратор ИС регистрирует потенциальный библиотечный объект в соответствующем журнальном файле. На втором этапе подсистема, формирующая библиотечный объект, используя запрос к МС, фактически передает этот объект в ИС. Таким образом, в библиотеку могут быть занесены только предварительно зарегистрированные объекты.

Локальные данные и их подготовка к использованию в ИС. К локальным данным относятся:

описание проекта;

проектные решения, выработанные в рамках данного проекта.

Описание проекта создается группой разработчиков, выполняющих работу над проектом, под руководством руководителя проекта. ИС обеспечивает язык описания проекта (ЯОП), позволяющий описать иерархическую структуру объекта проектирования (ОП), перечень идентификаторов типов проектных решений для каждого структурного компонента ОП, перечень подсистем интегрированной САПР, имеющих доступ к проектным решениям для каждого компонента ОП, перечень идентификаторов подсистем, требующих уведомления об изменении проектного решения для каждого компонента ОП.

Количество и внутренний формат файлов, содержащих проектное решение, сформированное КПП, определяется проектирующей подсистемой, реализующей эту процедуру. ИС обрабатывает каждый из этих файлов как единый информационный объект, упаковывая его, наряду с другими файлами, в информационный контейнер (.SAV-файл – набор сохранения утилиты BACKUP).

4.5.3. Взаимодействие информационной и мониторной систем

Одной из основных функций МС интегрированной САПР является обеспечение взаимодействия между проектными процедурами, осуществляемого путем передачи проектных решений. Согласно концепции сквозного проектирования, принятой при построении интегрированной САПР МСВТ, непосредственные информационные обмены между проектирующими подсистемами не допускаются. Все информационные обмены между подсистемами контролируются ЦМС. Единицей информационных обменов является контейнер. Схема передачи контейнера с КПП в хранилище ИС: КПП → ЛМС → ЦМС → ИС. Схема приема контейнера из хранилища ИС противоположна: КПП ← ЛМС ← ЦМС ← ИС.

Со стороны МС информационные обмены поддерживаются подсистемой взаимодействия, которая организует проектное решение в виде контейнера. Для организации обмена контейнерами пользователю предлагаются следующие виды работ, отраженные в меню “Подсистема взаимодействия”:

- редактор файла описания проектного решения (ФОПР);
- передача контейнера в ИС с использованием ЛВС;
- передача контейнера в ИС на МЛ;
- передача контейнера в ИС на ГМД;
- замена контейнера в ИС с использованием ЛВС;
- замена контейнера в ИС контейнером на МЛ;
- замена контейнера в ИС контейнером на ГМД;
- прием контейнера от ИС с использованием ЛВС;
- прием контейнера от ИС на МЛ;
- удаление контейнера из ИС.

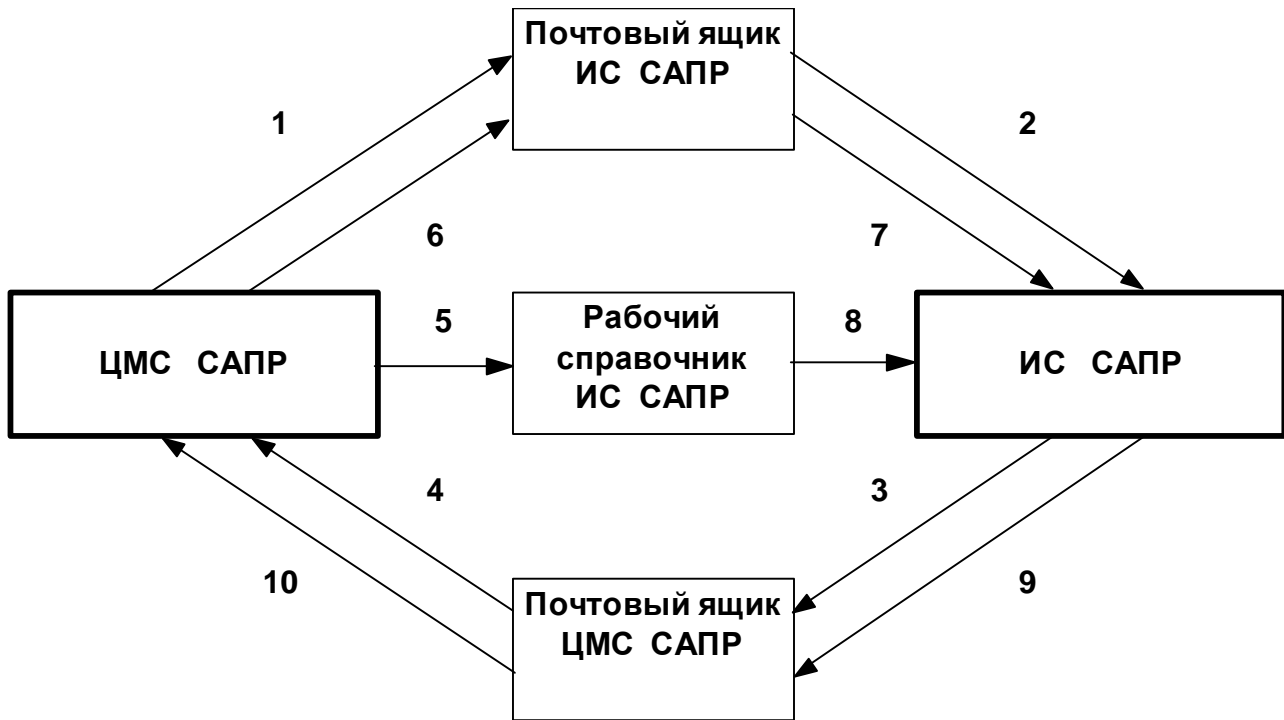
Ниже более подробно рассмотрены виды работ, связанные с передачей, удалением и заменой контейнеров в ИС интегрированной САПР, а также с приемом контейнеров от ИС.

Передача контейнера в информационную систему. Этот вид работы обеспечивает возможность формирования контейнера и передачи его в ИС интегрированной САПР МСВТ. После выбора пункта меню, соответствующего этому виду работы, пользователь должен ввести спецификацию ФОПР для передаваемого контейнера в ответ на запрос мониторной системы.

Затем, если работа выполняется в среде ЦМС, осуществляется проверка корректности контейнера по модели процесса проектирования. Критерием корректности контейнера является активное состояние проектной процедуры-отправителя контейнера. В случае корректности контейнера из файлов, перечисленных в ФОПР, и из самого ФОПР формируется контейнер. При этом содержимое файлов подвергается криптопреобразованию, т.е. шифруется. При формировании контейнера генерируется его уникальный идентификатор, который однозначно определяет контейнер. После успешного формирования контейнера его идентификатор отображается на экране видеотерминала. В дальнейшем этот идентификатор используется при выполнении работ по замене, приему и удалению данного контейнера.

Дальнейшие действия по передаче контейнера зависят от месторасположения МС, в которой выполняется данная работа. Если это ЦМС, то сформированный контейнер передается информационной системе, после чего последняя выполняет его тестирование. В случае успешного тестирования информационная система принимает контейнер и размещает его в своей базе данных. На этом работа по передаче контейнера завершается. Если же информационная система по каким-либо причинам не может принять контейнер, то выводится диагностическое сообщение. Укрупненная схема, описывающая последовательность операций по передаче контейнера с проектным решением из ЦМС в ИС интегрированной САПР, приведена на рис. 4.3.

Успешная передача контейнера в информационную систему регистрируется в журнале учета переданных контейнеров (файл MS\$JNL:CONTRS.IDX), и всем пользователям с 0-й категорией (ответственным исполнителям), работающим в подсистемах, в которых должны выполняться процедуры-получатели контейнера, рассылаются системные оповещения. Процедура-отправитель контейнера переводится в состояние “завершена локально”. Для процедур-получателей в модель процесса проектирования приходит уведомление о передаче контейнера. Если проектная процедура получила все необходимые ей контейнеры, то она переводится в состояние ожидания.



1. ЦМС посылает запрос (блок параметров) на обмен контейнером в почтовый ящик ИС.
2. ИС считывает запрос из своего почтового ящика.
3. ИС посылает ответ о корректности запроса (блока параметров) и своей готовности принять контейнер в почтовый ящик ЦМС.
4. ЦМС считывает ответ ИС из своего почтового ящика.
5. ЦМС передает контейнер с проектным решением в рабочий справочник ИС.
6. ЦМС посылает сообщение в почтовый ящик ИС о передаче контейнера в рабочий справочник ИС.
7. ИС считывает сообщение о передаче контейнера из своего почтового ящика.
8. ИС считывает контейнер из своего рабочего справочника в оперативное хранилище контейнеров.
9. ИС посылает ответ об успешном размещении контейнера в хранилище в почтовый ящик ЦМС.
10. ЦМС считывает ответ переданный ИС из своего почтового ящика.

Рис. 4.3. Схема передачи контейнера в ИС интегрированной САПР

Если работа по передаче контейнера выполняется в среде ЛМС, то контейнер будет передан по локальной сети в ЦМС, где он будет подвергнут проверке и, если она окажется успешной, передан в информационную систему, которая произведет обработку контейнера, как описано выше.

Если же передать контейнер по сети окажется невозможно (например, из-за проблем с техническими средствами ЛВС), контейнер может быть записан на магнитную ленту. В этом случае передачу контейнера в информационную систему необходимо осуществить, выполнив в ЦМС работу “Передача контейнера в информационную систему на МЛ”. Этот вид работы позволяет передать в ИС

контейнер, сформированный в ЛМС и записанный на магнитную ленту. Этот вид работы доступен только в ЦМС.

После выбора работы из меню пользователь должен сообщить ЦМС идентификатор передаваемого контейнера и установить магнитную ленту с контейнером на устройство НМЛ, определенное ЦМС. ЦМС выполнит чтение контейнера с магнитной ленты и осуществит ряд действий, аналогичных перечисленным выше.

Прием контейнера от информационной системы. Эта работа предоставляет проектной процедуре возможность получить контейнер, переданный для нее другой процедурой в информационную систему САПР.

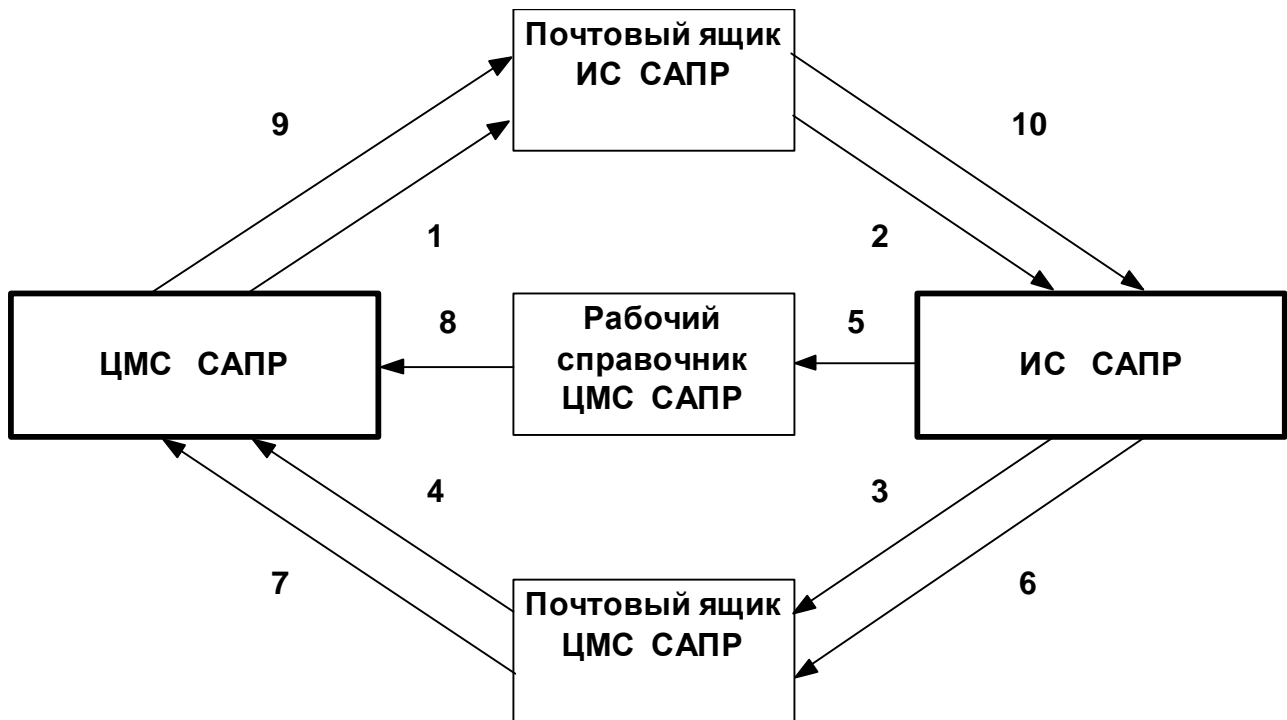
После выбора работы из меню пользователь должен сообщить мониторинг системе идентификатор контейнера, а также идентификаторы подсистемы, проекта, узла ОП и типа проектного решения, которые должны совпадать с содержимым соответствующих полей ФОПР принимаемого контейнера. Введенная информация передается в информационную систему. Если эта информация удовлетворяет требованиям ИС, то контейнер передается мониторинг системе САПР.

Укрупненная схема, описывающая последовательность операций по приему контейнера с проектным решением из ИС в ЦМС интегрированной САПР, приведена на рис. 4.4.

Дальнейшие действия зависят от того, где выполняется работа: в среде ЦМС или в среде ЛМС. Если эта работа выполняется в ЦМС, то ЦМС запрашивает номер компьютера в сети, на котором расположена подсистема, принимающая контейнер. Затем ЦМС проверяет целостность контейнера. Если целостность контейнера нарушена, то пользователь, запросивший контейнер, оповещается об этом и процесс обработки контейнера прекращается.

Если же проверка целостности контейнера дала положительный результат, то последовательность дальнейшей обработки зависит от расположения подсистемы-получателя контейнера. Если эта подсистема расположена на том же компьютере, что и ЦМС, то запрашивается спецификация справочника пользователя, в который должен быть передан контейнер и в который записываются файлы данных, извлеченные из контейнера.

Если работа по приему контейнера выполняется в среде ЛМС, то ЦМС считает, что подсистема, запросившая контейнер, находится на том же компьютере, что и ЛМС, и передает контейнер по сети. Если передача оказывается успешной, то проводится проверка контейнера и, если она дает положительный результат, ЛМС запрашивает спецификацию справочника и передает контейнер в указанный справочник. Если же передача по сети завершается неудачей, то контейнер может быть передан с помощью магнитной ленты или дискет.



1. ЦМС посылает запрос (блок параметров) на прием контейнера в почтовый ящик ИС.
2. ИС считывает запрос из своего почтового ящика.
3. ИС посылает ответ о корректности запроса (блока параметров) и своей готовности передать контейнер в почтовый ящик ЦМС.
4. ЦМС считывает ответ ИС из своего почтового ящика.
5. ИС передает контейнер с проектным решением в рабочий справочник ЦМС.
6. ИС посылает сообщение в почтовый ящик ЦМС о передаче контейнера в рабочий справочник ЦМС.
7. ЦМС считывает сообщение о передаче контейнера в свой рабочий справочник из своего почтового ящика.
8. ЦМС считывает контейнер из своего рабочего справочника.
9. ЦМС посылает ответ об успешном приеме контейнера в почтовый ящик ИС.
10. ИС считывает ответ переданный ЦМС из своего почтового ящика.

Рис. 4.4. Схема приема контейнера из ИС интегрированной САПР

Удаление контейнера из информационной системы. Эта работа позволяет удалить контейнер из информационной системы. Выполнение ее возможно только в среде ЦМС.

После выбора работы из меню пользователь должен сообщить ЦМС идентификатор удаляемого контейнера, а также идентификаторы проекта, узла ОП, подсистемы и типа проектного решения, которые должны совпадать с содержимым соответствующих полей ФОПР удаляемого контейнера. Информационная система проверяет введенную информацию и, если эта информация удовлетворяет ее требованиям, удаляет контейнер. Во все подсистемы, которым этот контейнер был предназначен, рассылаются системные оповещения об удалении контейнера. Факт удаления контейнера регистрируется в журнале учета удаленных контейнеров (файл MS\$JNL:DELCNTR.IDX).

Укрупненная схема, описывающая последовательность операций по удалению контейнера с проектным решением из информационного фонда ИС по запросу ЦМС интегрированной САПР, приведена на рис. 4.5.

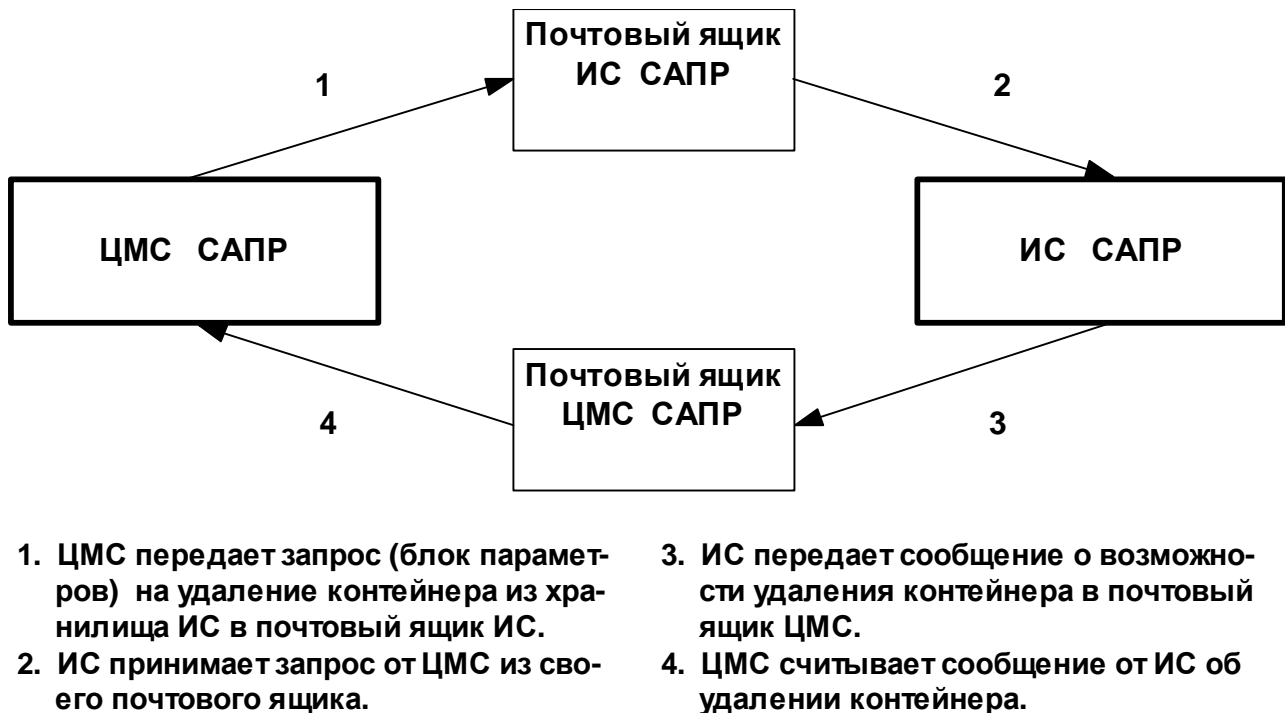


Рис. 4.5. Схема удаления контейнера из ИС интегрированной САПР

Замена контейнера в информационной системе. Этот вид работы позволяет поместить в базу данных информационной системы новую рабочую версию контейнера. После выбора работы из меню пользователь должен

сообщить МС спецификацию ФОПР для новой версии контейнера и идентификатор контейнера.

Затем, если работа выполняется в среде ЦМС, осуществляется проверка корректности контейнера по модели процесса проектирования. В случае корректности контейнера из файлов, перечисленных в ФОПР, и из самого ФОПР формируется контейнер, имеющий заданный идентификатор.

Дальнейшие действия по замене контейнера зависят от месторасположения МС, в среде которой выполняется работа. Если это ЦМС, то контейнер передается информационной системе, после чего подвергается тестированию. В случае успешного тестирования информационная система принимает контейнер и заменяет в своей базе контейнер с указанным идентификатором на вновь поступивший контейнер. Если же информационная система по каким-либо причинам не может произвести замену контейнера, выводится соответствующее диагностическое сообщение.

Успешная замена контейнера в информационной системе регистрируется в журнале учета модифицированных контейнеров (файл MS\$JNL:MODCNTR.IDX). При этом всем пользователям с нулевой категорией, работающим в подсистемах, выполняющих процедуры-получатели контейнера, выполняется рассылка системных оповещений. Процедура-отправитель контейнера переводится в состояние “завершена локально”.

Схема замены контейнера с проектным решением в БД ИС интегрированной САПР может представить как последовательное выполнение схем удаления и передачи контейнера в БД ИС.

Если работа по замене контейнера выполняется в среде ЛМС, то контейнер будет передан по локальной сети в ЦМС, где он будет подвергнут проверке и, если она окажется успешной, передан в информационную систему, которая произведет обработку контейнера, как описано выше.

Если же передать контейнер по сети невозможно, его можно записать на магнитную ленту. В этом случае замену контейнера в информационной системе необходимо осуществить, выполнив в среде ЦМС работу “Замена контейнера в информационной системе на контейнер с магнитной ленты”.

Эта работа позволяет заменить контейнер, содержащийся в информационной системе, на контейнер, сформированный в ЛМС и записанный там на магнитную ленту. Эта работа доступна только в среде ЦМС.

Пользователь должен сообщить ЦМС идентификатор контейнера и установить том магнитной ленты в устройство НМЛ, указанное ЦМС. ЦМС выполнит чтение контейнера с магнитной ленты и осуществит ряд действий по замене контейнера, аналогичных описанным выше.

4.5.4. Взаимодействие информационной и архивной систем

Взаимодействие ИС с АС заключается в передаче завершенных проектных решений в архив интегрированной САПР. Эта передача предусматривает следующий порядок действий:

1. Руководитель проекта извещает руководителя САПР о необходимости передачи законченного проектного решения в архив интегрированной САПР.

2. Руководитель САПР отдает распоряжение администратору АС о необходимости приема контейнеров с законченными проектными решениями в архив интегрированной САПР.

3. АС выполняет запрос к ЦМС на передачу соответствующих контейнеров с проектными решениями.

4. ЦМС выполняет анализ рабочей модели процесса проектирования с целью проверки логической целостности проекта. Проверяется, что все проектные процедуры модели имеют состояние “локально завершена”, т.е. контейнеры с их проектными решениями находятся в оперативном хранилище ИС САПР. Затем ЦМС переадресует полученный от АС запрос к ИС.

5. ИС передает копию каждого указанного контейнера (.SAV-файла) в МС, которая, в свою очередь, отправляет его в АС интегрированной САПР. Если в запросе дополнительно было задано удаление контейнера из оперативного хранилища, ИС удаляет контейнер (соответствующий .SAV-файл) и снимает его с учета.

6. АС интегрированной САПР, получив контейнер, помещает его в архив и выполняет его каталогизацию (постановку на учет).

7. АС отправляет ЦМС уведомление о том, что каждый из запрошенных контейнеров с проектными решениями успешно помещен в архив.

8. ЦМС, получив уведомление от АС, выполняет перевод всех проектных процедур рабочей модели процесса проектирования в состояние “глобально завершена”, что означает окончательное завершение работы по данному проекту, и удаляет запись, соответствующую данному проекту, из журнала учета идентификаторов проекта (файл MS\$IDOPRJ.IDX).

Авторами исследовались вопросы, связанные с разработкой модельного представления архивной информации и элементов архивной подсистемы САПР на базе перспективного запоминающего устройства терабитной емкости (ТЗУ), использующего последние достижения нанотехнологии [112, 113]. Эти исследования выполнялись в рамках хоздоговорной НИР “Нива-БЦУ” (“Разработка и изготовление экспериментального образца блока цифрового управления терабитным ЗУ”), их результаты использовались в экспериментальных работах по созданию ТЗУ в ГАНИИ “Дельта” (г. Москва).

4.6. Аттестация инструментальных программных средств

Распределенная вычислительная среда, реализованная интегрированной САПР МСВТ, представляет собой совокупность объединенных в локальную сеть функциональных подсистем. Ужесточение требований к качеству вырабатываемых с их помощью проектных решений приводит к необходимости использования только проверенных программных инструментальных средств. Среда обработки имеет динамический характер, когда взамен устаревший программных средств вводят в действие вновь разработанные эффективные и функционально более полные программы.

Прежде чем программа сможет использоваться в составе инструментального программного обеспечения интегрированной САПР МСВТ, она должна быть апробирована и аттестована. Процедура аттестации предполагает выполнение структурной перестройки заголовка выполняемого образа программы, запись о проведенной операции аттестации в журнальный файл MS\$JNL:IMAGES.JNL, формирование контейнера с аттестованной программой и передача его в хранилище контейнеров информационной системы САПР. Использование программы на проблемно-ориентированном АРМ предполагает

прием соответствующего контейнера из оперативного хранилища, проверку целостности и достоверности его содержимого, распаковку содержимого и настройку программы и таблицы команд пользователя.

В начале файла образа программы находится заголовок, который описывает образ и его секции [114]. Заголовок образа программы имеет сложную структуру (рис. 4.6). В начале заголовка находится часть заголовка фиксированного размера, которая содержит такую важную информацию, как размер заголовка, смещения до следующих частей заголовка, идентификационные номера и код типа образа, требуемая маска привилегий и т.д. Эта часть заголовка имеет два зарезервированных поля, каждое размером 32 бита (длинное слово) и смещениями 20 и 36 относительно начала заголовка. Эти поля можно использовать для хранения сигнатур (характерных цепочек бит), идентифицирующих аттестованные программы.

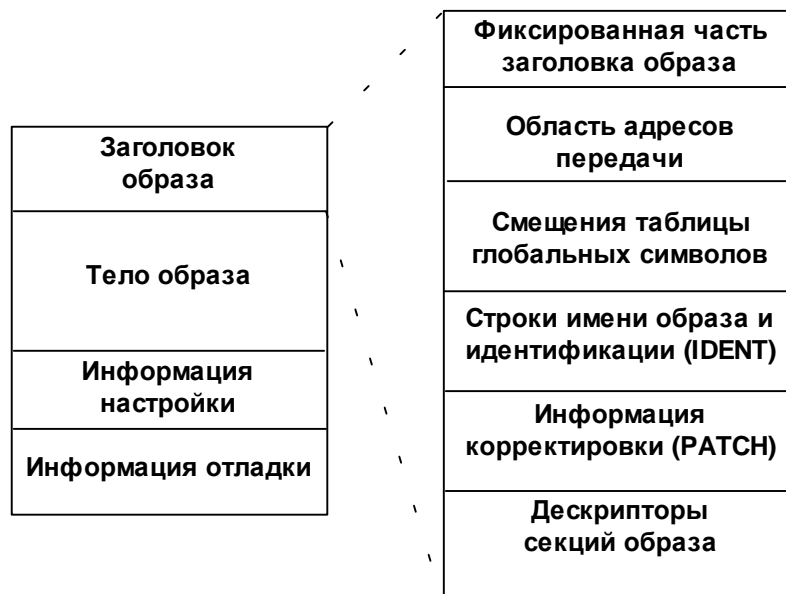


Рис. 4.6. Укрупненная структура заголовка образа программы

Для выполнения структурной перестройки фиксированной части заголовка программного образа и формирования идентифицирующих сигнатур используется программа *ModifyHeader* (файл MODHDR.EXE). После изменения заголовка программного образа запуск его обычным способом (по команде *RUN* или с помощью системной сервисной процедуры *SYS\$SPAWN* становится невозможным).

Запуск аттестованной программы в среде мониторной системы выполняется по команде *LEI*, которая предварительно восстанавливает первоначальную структуру заголовка образа с помощью программы *RestoreHeader* (файл *RESTHDR.EXE*). Восстановлению образа предшествует проверка полей сигнатур, позволяющая выявить программы, не прошедшие аттестацию, которые не могут быть запущены в среде мониторной системы.

С другой стороны, аттестованные программы без предварительного восстановления заголовка не могут быть запущены в среде операционной системы. Таким образом, процедура аттестации обеспечивает определенную дисциплину использования инструментальных программных средств, предотвращая 1) запуск в среде мониторной системы не аттестованных программных средств и 2) запуск аттестованных программных средств вне мониторной системы.

Программы *ModifyHeader* и *RestoreHeader* написаны на языке *Bliss-32* [72, 73]. Файлы выполнимых образов этих программ имеют размеры приблизительно равные 17 и 12 Кбайтам.

Просмотр и печать содержимого журнала аттестованных программных средств выполняется с помощью программы *Typer* (*TYPEDR.EXE*). Эта программа написана на языке *Фортран/МВС* [74, 75]. Размер программы приблизительно равен 23 Кбайтам.

Заключение

В ходе работы по созданию мониторинговой системы интегрированной САПР МСВТ авторами получены следующие основные научно-практические результаты:

Выполнена формализация процесса проектирования и предложена его концептуальная модель в виде асинхронного последовательно-параллельного конвейера, отличающаяся от ранее использованной введением понятий комплексной проектной процедуры, комплексного проектного решения и откатов различной глубины.

Предложена унифицированная схема системного интерфейса между функциональными подсистемами, основанная на использовании понятия информационных структур контейнерного типа и обеспечивающая поддержку стратегии комплексной обработки информации в интегрированной распределенной среде.

Предложена концепция распределенной мониторинговой системы с логической звездообразной структурой, обеспечивающая возможность контроля информационных обменов между функциональными подсистемами, образующими интегрированную распределенную среду обработки.

Разработан итеративный алгоритм обхода вершин произвольного ориентированного графа, обеспечивающий эффективную программную реализацию процедуры управления типа “глубокий откат” в технологическом маршруте обработки.

Разработано математическое и программное обеспечение поддержки динамических свойств среды распределенной обработки данных, основанное на механизмах модификации структуры заголовка программного образа модулей функциональных подсистем и гарантирующее использование “легальных” программных средств для реализации комплексной обработки данных.

На основе предложенной концептуальной модели процесса проектирования разработана обобщенная информационная модель, отражающая процесс обработки в следующих проекциях: структура проекта, структура технологического маршрута, календарный план-график работ над проектом. Для описания и

актуализации введенной информационной модели разработано лингвистическое, математическое и программное обеспечение мониторинной системы.

Разработанные базовые компоненты МС используются в составе математического, программного, лингвистического и информационного обеспечения интегрированной САПР МСВТ в ОАО “ОКБ Процессор” и на заводе “Процессор” (г. Воронеж).

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Энкарначчо Ж., Шлехтендаль Э. Автоматизированное проектирование. Основные понятия и архитектура систем: Пер. с англ. – М.: Радио и связь, 1986. – 288 с.
2. Липп Х.М. Методические аспекты логического синтеза: Пер. с англ. // ТИИЭР, т. 71, №1, 1981. – С. 110-121.
3. Ниссен К. Методология и средства иерархического проектирования СБИС: Пер. с англ. // ТИИЭР, т. 71, №1, 1981. – С. 81-94.
4. Харин В.Н., Стариков А.В. Информационные аспекты проектирования высоконадежных функциональных систем // Оптимизация и моделирование в автоматизированных системах: Межвуз. сб. научн. тр., Воронеж: ВГТУ, 1995. – С. 68-70.
5. Харин В.Н., Стариков А.В. Обобщенная информационная модель проекта высоконадежных систем особой сложности // Методы и средства оценки и повышения надежности приборов, устройств и систем: Тез. докл. междунар. науч.-технич. конф., Пенза, 1996. – С. 132-134.
6. Автоматизация схемотехнического проектирования: Учебн. пособие для вузов / Ильин В.Н., Фролкин В.Т., Бутко А.И., Камнева Н.Ю., Тихомирова Е.М.; Под ред. Ильина В.Н. – М.: Радио и связь, 1987. – 368 с.
7. Интеграция данных в САПР БИС. Направления практической реализации / Беляков Ю.Н., Руденко А.А., Топузов И.Г., Егоров Ю.Б. – М.: Радио и связь, 1990. – 160 с.
8. Гантер Р. Методы управления проектированием программного обеспечения: Пер. с англ. – М.: Мир, 1981. – 392 с.
9. Коллинз Г., Блей Дж. Структурные методы разработки систем: от стратегического планирования до тестирования. Пер. с англ. – М.: Финансы и статистика, 1986. – 264 с.
10. Малиньяк Л. Дальнейшее расширение функциональных возможностей САПР // Электроника, 1991, № 11-12. – С. 15-23.
11. Ганн Л. Какой критерий станет главным при оценке САИТ в 1990-е годы? // Электроника, 1990, № 2. – С. 9-19.

12. Бунза Дж., Хоффман Г., Томпсон Э. Основные направления развития автоматизации проектирования в 1990-х годах // Электроника, 1990, № 2. – С. 39-47.
13. Маклауд Дж. Новая методология проектирования требует создания более эффективных средств САПР // Электроника, 1990, № 8. – С. 14-20.
14. Уэбер С. Пакет программ синтеза схем на основе описаний на языке VHDL // Электроника, 1990, № 8. – С. 23-26.
15. Малиньяк Л. Начало широкомасштабного внедрения средств синтеза схем в системы автоматизированного проектирования // Электроника, 1992, № 1-2. – С. 98-106.
16. LOGIC DESIGN AND SIMULATION / E. Horbst (Editor). – Elsevier Science Publishers B.V. (North-Holland), 1986. – 256 с.
17. Клейнрок М. Теория массового обслуживания: Пер. с англ. – М.: Машиностроение, 1979. – 432 с.
18. Фуксман А.Л. Технологические аспекты создания программных систем. – М.: Статистика, 1979. – 184 с.
19. Системы автоматизированного проектирования в радиоэлектронике: Справочник / Авдеев Е.В., Еремин А.Т., Норенков И.П., Песков М.И.; Под ред. Норенкова И.П. – М.: Радио и связь, 1986. – 368 с.
20. Корячко В.П., Курейчик В.М., Норенков И.П. Теоретические основы САПР: Учеб. для вузов. – М.: Энергоатомиздат, 1987. – 400 с.
21. Норенков И.П., Маничев В.Б. Основы теории и проектирования САПР: Учеб. для вузов по спец. “Вычислительные маш., компл., сист. и сети”. – М.: Высш. шк., 1990. – 335 с.
22. Харин В.Н. Базовое программное обеспечение типовых САПР изделий микроэлектроники. – Электронная промышленность, 1987, № 5. – С. 58-60.
23. Ганн Л. Инструментальные средства автоматизации проектирования, обеспечивающие параллельную работу над проектами // Электроника, 1990, № 7. – С. 58-61.
24. Belkhale K., Brouwer R., Banerjee P. Task scheduling for exploiting parallelism and hierarchy in VLSI CAD algorithms // IEEE Trans. Comput. – Aid. Des. Integr. Circuits and Syst., 1993, № 5. – С. 557-567.

25. Обеспечение параллельного выполнения проектных работ в интегрированной САПР сквозного проектирования СБИС / Харин В.Н., Стариков А.В., Удовик А.П., Горохов А.В. – Электронная промышленность, 1994, № 4-5. – С. 94-96.
26. Харин В.Н., Стариков А.В. Инструментальные средства распараллеливания работ при проектировании СБИС // Оптимизация и моделирование в автоматизированных системах: Межвуз. сб. научн. тр., Воронеж: ВГТУ, 1994. – С. 52-54.
27. Олбрайт Д. Подготовка к внедрению стандартов на инфраструктуры в области САПР-электроники // Электроника, 1990, № 2. – С. 47-53.
28. Electronic CAD Frameworks / D.S. Harrison, A.R. Newton, R.L. Spickelmier, T.J. Barnes // Proceedings of the IEEE, 1990, Vol. 78, № 2. – С. 393-417.
29. Системная среда САПР СБИС / А.Л. Стемпковский, В.А. Шепелев, А.В. Власов. – М.: Наука, 1994. – 251 с.
30. Малиньяк Л. Управление данными проектирования средствами инфраструктуры САПР // Электроника, 1991, № 11-12. – С. 23-31.
31. Ван Тайл Ш. Современные консорциумы – что это такое? // Электроника, 1995, № 2. – С. 75-81.
32. Месарович М., Такахара Я. Общая теория систем: математические основы / Пер. с англ. – М.: Мир, 1978. – 311 с.
33. Острейковский В.А. Теория систем: Учеб. для вузов по спец. “Автом. сист. обр. информ. и упр.”. – М.: Высш. шк., 1997. – 240 с.
34. ГОСТ 22487–77 ПРОЕКТИРОВАНИЕ АВТОМАТИЗИРОВАННОЕ. Термины и определения – М.: Изд-во стандартов, 1978. – 11 с.
35. ГОСТ 23501.13–81 САПР. Мониторная система. Общие требования – М.: Изд-во стандартов, 1981. – 6 с.
36. Блохин В.Н., Демин А.В., Немолочнов О.Ф. Формализация процесса проектирования в ИСАПР // Автоматизированное проектирование в радиоэлектронике и приборостроении: Сб. научн. тр. – Санкт-Петербург. гос. электротехн. ун-т, СПб, 1994. – С. 11-14.
37. Хомский Н. Формальные свойства грамматик: Пер. с англ. / Кибернетический сборник, новая серия, вып. 2. – М.: Мир, 1966. – С. 121-280.

38. Рейуорд-Смит В. Дж. Теория формальных языков. Вводный курс: Пер. с англ. – М.: Радио и связь, 1988. – 128 с.
39. Алгоритмический язык Алгол-60 / Под ред. П. Наура: Пер. с англ. – М.: Мир, 1965. – 71 с.
40. Вирт Н. Программирование на языке Модула-2: Пер. с англ. – М.: Мир, 1987. – 224 с.
41. Вирт Н. Алгоритмы и структуры данных: Пер. с англ. – М.: Мир, 1989. – 360 с.
42. Йенсен К., Вирт Н. Паскаль. Руководство для пользователя: Пер. с англ. – М.: Финансы и статистика, 1989. – 255 с.
43. Харин В.Н., Стариков А.В. Язык описания технологического маршрута проектирования в интегрированной САПР СБИС // Оптимизация и моделирование в автоматизированных системах: Межвуз. сб. научн. тр., Воронеж: ВГТУ, 1994. – С. 45-51.
44. Семантика языков программирования: Сб. статей / Пер. с англ. под ред. В.М. Курочкина. – М.: Мир, 1980. – 394 с.
45. Тетельбаум А.Я. Иерархический подход к проектированию сверхбольших интегральных схем // Электронная техника. – Сер. 10, Микроэлектронные устройства. – 1981. – Вып. 6 (30). – С. 29-32.
46. Автоматизированное проектирование СБИС на базовых кристаллах // Петренко А.И., Лошаков В.Н., Тетельбаум А.Я., Шрамченко Б.Л. – М.: Радио и связь, 1988. – 160 с.
47. Харин В.Н., Стариков А.В. Язык описания информационной модели проекта в интегрированной САПР СБИС // Оптимизация и моделирование в автоматизированных системах: Сб. научн. тр., Воронеж, ВГТУ, 1994. – С. 112-115.
48. Пратт Т. Языки программирования: разработка и реализация / Пер. с англ. – М.: Мир, 1979. – 574 с.
49. Грис Д. Конструирование компиляторов для цифровых вычислительных машин: Пер. с англ. – М.: Мир, 1975. – 544 с.
50. Льюис Ф., Розенкранц Д., Стирнз Р. Теоретические основы проектирования компиляторов: Пер. с англ. – М.: Мир, 1979. – 654 с.

51. Маккиман У., Хорнинг Дж., Уортман Д. Генератор компиляторов: Пер. с англ. – М.: Статистика, 1980. – 526 с.
52. Хантер Р. Проектирование и конструирование компиляторов: Пер. с англ. – М.: Финансы и статистика, 1984. – 232 с.
53. Waite W., Goos G. Compiler Construction. – Springer-Verlag New York Inc., 1984. – 446 p.
54. 589.7545471.00051 – 01 35 01. Многофункциональная виртуальная операционная система. Макроассемблер. Описание языка.
55. Остапенко Г.П., Толмачева Н.А., Горский В.Е. Макроассемблер для СМ1700. – М.: Финансы и статистика, 1990. – 239 с.
56. Баазе С. Ассемблер мини-ЭВМ VAX-11: Пер. с англ. – М.: Финансы и статистика, 1988. – 413 с.
57. Сибеста Р. Структурное программирование на языке ассемблера ЭВМ VAX-11: Пер. с англ. – М.: Мир, 1988. – 536 с.
58. Кэпс Ч., Стаффорд Р. VAX: Программирование на языке ассемблера и архитектура / Пер. с англ. – М.: Радио и связь, 1991. – 416 с.
59. Лин В. PDP-11 и VAX-11. Архитектура ЭВМ и программирование на языке ассемблера: Пер. с англ. – М.: Радио и связь, 1989. – 320 с.
60. Малые ЭВМ высокой производительности. Архитектура и программирование / Г. П. Васильев, Г. А. Егоров, В.С. Зонис и др.; Под ред. Н. Л. Прохорова. – М.: Радио и связь, 1990. – 256 с.
61. 589.7545471.00051– 01 33 01. Многофункциональная виртуальная операционная система. Макроассемблер. Руководство программиста.
62. 589.7545471.00051– 01 33 08. Многофункциональная виртуальная операционная система. Редактор связей. Руководство программиста.
63. 589.7545471.00051– 01 33 02. Многофункциональная виртуальная операционная система. Системные сервисные процедуры. Руководство программиста.
64. 589.7545471.00051– 01 33 12. Многофункциональная виртуальная операционная система. Библиотека общих процедур. Руководство программиста.
65. 589.7545471.00051– 01 31 03. Многофункциональная виртуальная операционная система. Система управления записями. Описание применения.

66. 589.7545471.00051– 01 33 04. Многофункциональная виртуальная операционная система. Система управления записями. Руководство программиста.
67. 589.754547.00161– 01 13 01. Мониторная система САПР. Описание программы. – Воронеж, 1989. – Л. 30.
68. 589.754547.00161– 01 32 01. Мониторная система САПР. Руководство системного программиста. – Воронеж, 1989. – Л. 13.
69. 589.754547.00161– 01 33 01. Мониторная система САПР. Руководство программиста. – Воронеж, 1989. – Л. 77.
70. 589.754547.00161–01 34 01. Мониторная система САПР. Руководство оператора. – Воронеж, 1989. – Л. 65.
71. 589.754547.00161– 01 51 01. Мониторная система САПР. Программа и методика испытаний. – Воронеж, 1989. – Л. 11.
72. VAX-11 BLISS-32 Language Guide.
73. VAX-11 BLISS-32 User's Guide.
74. 589.7545471.00051– 01 35 02. Многофункциональная виртуальная операционная система. Фортран/МВС. Описание языка.
75. 589.7545471.00051– 01 33 03. Многофункциональная виртуальная операционная система. Фортран/МВС. Руководство программиста.
76. Ахо А., Хопкрофт Дж., Ульман Дж. Построение и анализ вычислительных алгоритмов: Пер. с англ. – М.: Мир, 1979. – 536 с.
77. Гудман С., Хидетниemi С. Введение в разработку и анализ алгоритмов: Пер. с англ. – М.: Мир, 1981. – 368 с.
78. Свами М., Тхуласираман К. Графы, сети и алгоритмы: Пер. с англ. – М.: Мир, 1984. – 455 с.
79. Евстигнеев В.А. Применение теории графов в программировании. / Под ред. А. П. Ершова. – М.: Наука. Гл. ред. физ.-мат. лит., 1985. – 352 с.
80. Стариков А.В. Итеративный алгоритм для обхода произвольного орграфа // Современные проблемы информатизации: Тез. докл. IV междунар. электронной научн. конф., Воронеж: ВГПУ, 1999. – С. 58.
81. Стариков А.В. Итеративный алгоритм для обхода вершин в ациклическом орграфе // Межвуз. сб. научн. тр. “Математическое моделирование, компьютерная оптимизация технологий, параметров оборудования и систем управления лесного комплекса”, Воронеж: ВГЛТА, 1999. – С. 220-223.

82. Tarjan R.E. Depth first search and linear graph algorithms. – SIAM J. Computing, 1972, 1, № 2, P. 146-160.
83. Глотов М.В., Завгородний С.В., Стариков А.В. Об одной реализации итеративного алгоритма для обхода вершин произвольного орграфа // Межвуз. сб. научн. тр. “Математическое моделирование, компьютерная оптимизация технологий, параметров оборудования и систем управления лесного комплекса”, Воронеж: ВГЛТА, 1999. – С. 224-227.
84. 589.7545471.00051 – 01 35 04. Многофункциональная виртуальная операционная система. Паскаль/МВС. Описание языка.
85. 589.7545471.00051 – 01 33 03. Многофункциональная виртуальная операционная система. Паскаль/МВС. Руководство программиста.
86. Интерактивные графические системы нового поколения для САПР изделий электронной техники / Власов А.М., Дыбой В.А., Межов В.Е., Плотников В.В., Харин В.Н. // Автометрия, 1986, № 5. – С. 3-8.
87. Мини- и микроЭВМ семейства “Электроника” / Толстых Б.Л., Талов И.Л., Цывинский В.Г., Межов В.Е., Плотников В.В., Бондарович Г.Г. – М.: Радио и связь, 1987. – 294 с.
88. АРМ и ИГС нового поколения для САПР изделий микроэлектроники / Власов А.М., Плотников В.В., Харин В.Н., Чекмарев В.П. // Электронная промышленность, 1987, № 5. – С. 56-58.
89. Проектирование САПР и АРМ изделий электронной и вычислительной техники: Учеб. пособие / Межов В.Е., Питолин В.М., Плотников В.В., Харин В.Н. – Воронеж: ВПИ, 1989. – 101 с.
90. Техническое обеспечение САПР: Учеб. пособие / Дыбой В.А., Межов В.Е., Питолин В.М., Проценко И.Г. – Воронеж: ВПИ, 1990. – 92 с.
91. Унифицированные программно-технические комплексы для САПР ИЭТ и СВТ / Лопатин В.С., Харин В.Н., Межов В.Е., Горохов А.В., Чевычелов Ю.А. // Электронная промышленность, № 4-5, 1994. – С. 90-96.
92. Системы автоматизации проектирования и информационные управляющие комплексы на базе супермикроЭВМ / Лопатин В.С., Межов В.Е., Чевычелов Ю.А., Сергеев Б.С., Баранников Н.И. – Воронеж: ВГТУ, 1996. – 254 с.

93. Система ускоренного проектирования БИС / Левов Ю.А., Межов В.Е., Чевычелов Ю.А., Кононыхина Н.А. // Электронная промышленность, 1994, № 4-5. – С. 96-98.
94. Networks. DIGITAL Ethernet Products and Services Catalog (ED-25484-42). – 145 p.
95. Межов В.Е., Чевычелов Ю.А., Кононыхина Н.А. Подсистема “ИГРА” – графическая поддержка человеко-машинной системы проектирования аппаратуры / Эффективность, качество, надежность систем “человек – техника”: Тез. докл. IX симпоз. – Воронеж, 28-30 ноября, 1990. – С. 108-109.
96. Кононыхина Н.А., Межов А.В., Рындин А.А. Интерактивная графическая подсистема формирования данных в унифицированных системах автоматизированного проектирования типа “КУЛОН” / Новые информационные технологии в науке, образовании и бизнесе: Тез. докл. Междунар. конф. и шк. молодых ученых и специалистов “САПР-92”. – Воронеж, 4-13 мая, 1992. – С. 122-123.
97. Кононыхина Н.А., Межов А.В., Рындин А.А. Перспективные графические средства как основа высокой эффективности и бездефектности проектирования электронных схем / Новые информационные технологии в науке, образовании и бизнесе: Тез. докл. Междунар. конф. и шк. молодых ученых и специалистов “САПР-92”. – Воронеж, 4-13 мая, 1992. – С. 124-125.
98. Пакет “ИГРА” – интерактивная графическая среда процесса сквозного проектирования цифровой аппаратуры / Кононыхина Н.А., Лапшина М.А., Межов А.В., Рындин А.А. // Оптимизация и моделирование в автоматизированных системах: Межвуз. сб. научн. тр. – Воронеж: ВПИ, 1992. – С. 137-142.
99. Кононыхина Н.А., Чевычелов Ю.А. Графические средства системы ускоренного моделирования / Оптимизация и моделирование в автоматизированных системах: Межвуз. сб. научн. тр. – Воронеж: ВГТУ, 1994. – С. 39-44.
100. Интерактивные графические средства поддержки проектирования МЭА: Учеб. пособие / Межов В.Е., Питолин В.М., Чевычелов Ю.А., Кононыхина Н.А. – Воронеж: ВГТУ, 1994. – 105 с.

101. Лопатин В.С., Медведкова И.Е., Межов В.Е. Реализация иерархического моделирования в пакете ПРИАМ // Автометрия, 1990, № 5. – С. 117-119.
102. Львович Я.Е., Медведкова И.Е., Межов В.Е. Языки, методы и алгоритмы многоуровневого поведенческого моделирования: Учеб. пособие – Воронеж: ВПИ, 1992. – 109 с.
103. Интерактивная система логического моделирования цифровых схем / Власов А.М., Дыбой В.А., Межов В.Е., Плотников В.В., Чевычелов Ю.А. // Автометрия, 1986, № 5. – С. 9-12.
104. Лобов И.Е., Межов В.Е., Чевычелов Ю.А.. Логическое моделирование и генерация тестов цифровых схем в системе “Кулон” / Актуальные проблемы создания интеллектуальных САПР РЭА и СБИС: Тез. докл. Всесоюз. шк.-семина. молодых ученых и специалистов. – Гурзуф, 11-19 сентября, 1989. – С. 161-167.
105. Дыбой В.А., Межов В.Е., Рындин А.А. Автоматизация функционально-логического проектирования микроэлектронных устройств и аппаратуры на мини-ЭВМ: Учеб. пособие – Воронеж: ВПИ, 1990. – 78 с.
106. Горохов А.В., Скляр В.В., Устинов В.С. Пакет проектирования топологии коммутационных слоев матричных БИС / Разработка и оптимизация САПР и ГАП изделий электронной техники на базе высокопроизводительных мини- и микроЭВМ: Тез. докл. Всесоюз. шк.-семина. молодых ученых и специалистов. – Гурзуф, 11-19 сентября, 1989. – С. 96.
107. Седов Н.Н., Горохов А.В., Харин В.Н. Системный подход к задачам топологического проектирования БИС / Эффективность, качество, надежность систем “человек - техника”: Тез. докл. IX симпози. – Воронеж, 28-30 ноября, 1990. – С. 120-121.
108. Горохов А.В., Кирсанова Ю.И., Харин В.Н. Структурная декомпозиция топологии МАБИС / Методы искусственного интеллекта в САПР: Тез. докл. всесоюз. шк.-семина. молодых ученых. – Гурзуф, 5-13 мая, 1990. – С. 68.
109. Каталог средств вычислительной техники, программного обеспечения и услуг, предоставляемых заводом “Процессор”. – Воронеж, 1990. – 58 с.
110. Стариков А.В., Щекалев Ю.В., Харин В.Н. Динамическая модель управления распределенным процессом проектирования // Сб. тр. II междунар. на-

- учно-практич. конф. “Информационные технологии в моделировании и управлении”. – Санкт-Петербург: СПбГТУ, 2000. – С. 340-342.
111. 589.7545471.00051 – 01 34 03 Многофункциональная виртуальная система. Программы обслуживания. Руководство оператора.
112. Стариков А.В., Харин В.Н. Информационная модель архивного представления проектной информации в интегрированных САПР СБИС / САПР-95: Информационные технологии в науке, образовании, медицине и бизнесе: Тез. докл. XXII междунар. конф. и шк. по автоматизированному проектированию, Украина, Ялта-Гурзуф, 1995. – С. 268-269.
113. Информационная модель, поиск и распознавание информации в ПЗУ терабитной емкости / Борисенков Д.В., Истлентьев И.С., Стариков А.В., Харин В.Н. // Оптимизация и моделирование в автоматизированных системах: Межвуз. сб. научн. трудов, Воронеж: ВГТУ, 1995. – С. 71-75.
114. Kenah L., Goldenberg R., Bate S. VAX/VMS Internals and Data Structures. – Digital Press, 1988.

Содержание

	ВВЕДЕНИЕ	3
Глава 1.	ИНФОРМАЦИОННАЯ МОДЕЛЬ ПРОЦЕССА ПРОЕКТИРОВАНИЯ	6
1.1.	Обзор моделей процесса проектирования	7
1.1.1.	Концептуальная модель процесса проектирования	7
1.1.2.	Особенности среды проектирования	12
1.1.3.	Сетевая модель процесса проектирования	14
1.1.4.	Методологии проектирования	17
1.1.5.	Конвейер проектирования	19
1.1.6.	Организация параллельного проектирования	22
1.1.7.	Концепция сквозного проектирования в интегрированной САПР	24
1.1.8.	Инфраструктуры САПР	25
1.2.	Управление проектом и процессом проектирования в САПР	28
1.2.1.	Общие понятия и задачи управления	29
1.2.2.	Задачи управления процессом проектирования	31
1.2.3.	Формализация процесса проектирования	35
1.3.	Информационная модель процесса проектирования	40
Глава 2.	ЯЗЫК ОПИСАНИЯ И ПРОГРАММНЫЕ СРЕДСТВА ПОДДЕРЖКИ ИНФОРМАЦИОННОЙ МОДЕЛИ ПРОЦЕССА ПРОЕКТИРОВАНИЯ	44
2.1.	Язык описания информационной модели процесса проектирования	45
2.1.1.	Синтаксическая нотация	45
2.1.2.	Синтаксис ЯОМ	46
2.1.3.	Семантика ЯОМ	48
2.2.	Схема создания информационной модели процесса проектирования	51
2.3.	Программные средства поддержки модели процесса проектирования	53
2.3.1.	Генератор рабочей модели процесса проектирования.	53

2.3.2.	Программы оперативной корректировки рабочей модели	64
Глава 3.	ОСНОВНЫЕ ЗАДАЧИ ПОДДЕРЖКИ ОПЕРАТИВНОГО УПРАВЛЕНИЯ ПРОЕКТОМ, СПОСОБЫ И СРЕДСТВА ИХ РЕШЕНИЯ	70
3.1.	Мониторинг текущего состояния проекта.	70
3.2.	Координация действий проектировщиков.	74
3.3.	Контроль логической целостности проекта	79
3.4.	Перепланирование и временная увязка проектных работ . .	80
3.5.	Реализация отката в технологическом маршруте	82
Глава 4.	СОСТАВ, СТРУКТУРА И ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖ- НОСТИ МОНИТОРНОЙ СИСТЕМЫ	89
4.1.	Общие сведения об интегрированной САПР ИЭТ-МСВТ . .	89
4.2.	Структура управления интегрированной САПР.	92
4.3.	Функциональные возможности мониторной системы	93
4.3.1.	Подсистема генерации	95
4.3.2.	Подсистема диалога	97
4.3.3.	Подсистема регистрации	97
4.3.4.	Справочная подсистема	99
4.3.5.	Информационная подсистема	99
4.3.6.	Подсистема управления	99
4.3.7.	Подсистема контроля	100
4.3.8.	Подсистема защиты	101
4.3.9.	Подсистема взаимодействия	102
4.3.10.	Подсистема связи	103
4.4.	Взаимодействие проектирующих подсистем САПР	103
4.5.	Информационная подсистема интегрированной САПР	105
4.5.1.	Понятие контейнера	106
4.5.2.	Информационный фонд ИС	108
4.5.3.	Взаимодействие информационной и мониторной систем . . .	110
4.5.4.	Взаимодействие информационной и архивной систем	117
4.6.	Аттестация инструментальных программных средств	118
	ЗАКЛЮЧЕНИЕ.	121
	БИБЛИОГРАФИЧЕСКИЙ СПИСОК	123

Стариков Александр Вениаминович
Харин Валерий Николаевич

Управление сложными проектами в интегрированных САПР