

## **7. АНАЛИЗ И СИНТЕЗ**

До сих пор мы рассматривали автоматизацию управления в узком смысле, поэтому в сферу нашего внимания попадали процессы и объекты, вовлеченные в сферу компьютерных технологий. Но все это время предприятие продолжало работать, и наряду с автоматизированными комплексами существовали и функционировали традиционные механизмы управления. Общий результат достигался совместными усилиями автоматизированных и неавтоматизированных рабочих модулей, и параллельно с развитием компьютерных технологий развивалась большая технология. Теперь мы намерены вернуться к самому началу и рассмотреть эволюцию АСУП именно с этой точки зрения, то есть в контексте развития всей системы управления.

### **7.1 Эволюция системы**

Выделим основные вопросы и ключевые точки, на которые следует обратить особое внимание при обзоре поколений АСУП.

#### **Повторение пройденного**

Старые учебники, отражающие мнение первых разработчиков, говорят о том, что АСУП существует и развивается по своим собственным законам, которых вполне достаточно для понимания ее анатомии и физиологии. На самом деле это далеко не так. К моменту появления первых компьютеров, на предприятии уже существовали развитые механизмы управления, соответствующие технологиям высоких уровней. Не так уж важно, что они назывались по-другому и были ориентированы на человека, а не на компьютер. Скажем, ГОСТы на конструкторские обозначения или бухгалтерские счета существовали задолго до автоматизации.

Следовательно, при всей кажущейся самостоятельности АСУП, она двигалась по утопанной дорожке, ибо разработчики общались с пользователями, уже прошедшими системную эволюцию и успешно применяющими технологии высокого уровня. Пусть эти технологии не были доведены до глубины, пусть они не были формализованы, но все равно их существование давало разработчикам гораздо больше, чем полная пустота и отсут-

ствии каких-либо идей.

Открытия, которые затем появились в информатике – это лишь углубленная и формализованная трактовка прототипов, это освоение, окультуривание докомпьютерных методов, а отнюдь не оригинальные, впервые полученные результаты. Разработчики выступили в роли высококвалифицированных аранжировщиков, но не авторов.

#### Наполнение пирамиды

Итак, будущая высокоразвитая АСУП зарождалась в недрах старой системы управления и «воспитывалась» ею, то есть перенимала ее методы и решения. Новая система рассматривалась как средство автоматизации простых задач, а затем, по мере успешного продвижения

вперед, работники управления перешли к автоматизации более сложных механизмов управления.

Старая система управления представляла собой пирамиду, в которой далеко не все работники использовали технологии высоких уровней (рис. 43). Мы уже выяснили, что рядовой исполнитель выполняет вполне конкретные задачи, и от него не требуется понимания их глубокой и динамической связи с другими процессами. Даже если бы он и обладал таким пониманием, оно осталось бы невостребованным, поскольку управление задачами осу-

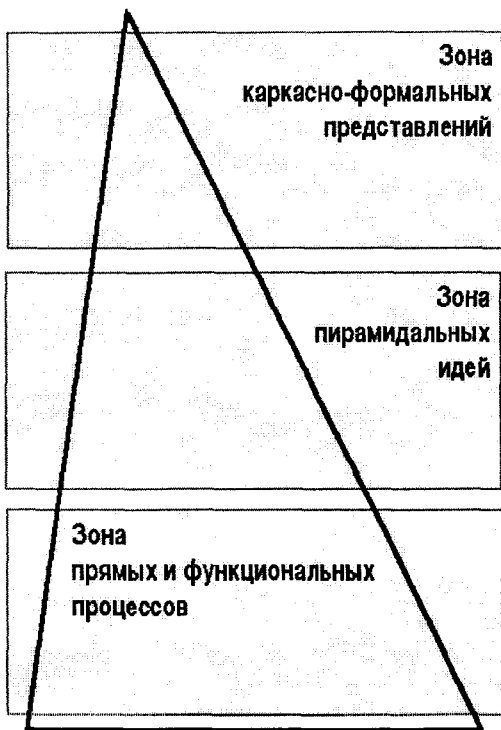


Рис. 43. Расположение технологических зон

ществляется на верхних уровнях пирамиды. Следовательно, каждому уровню административной пирамиды присуща своя технология.

Но было бы ошибкой считать, что базовая технология конкретного работника определяется тем уровнем, которого он достиг в административной иерархии. Это взаимосвязанные вещи, поскольку опыт и знания способствуют карьере, а их отсутствие – понижению в должности, но соответствие всегда относительно. В итоге, индивидуальная платформа опирается на базовую технологию, и затрагивает смежные – как вверх, так и вниз.

### **Формирование АСУП**

Задача автоматизации формулируется пользователями, находящимися на разных технологических уровнях, а в силу многочисленности работников нижнего уровня лучше всего слышны прямые и функциональные требования. Если учесть, что программист сам по себе имеет какой-то технологический базис, а также использует языки программирования определенного технологического толка, становится понятно то многообразие методов, тот технологический хаос, который наблюдался в развитии АСУП.

Создалась целая серия вариантов, генерируемая в зависимости от того, на каком технологическом уровне стоит «заказчик», на какой уровень он ориентирует «исполнителя», и на каком уровне тот действительно предполагает находиться. Скажем, разработчик, исполняющий объектные представления, справедливо считает функциональную позицию пользователя недостаточно глубокой. Но те же самые объектные представления не дают ему возможности увидеть каркас старой системы и, тем более, элементы формального синтеза.

До сих пор мы считаем количество высвобожденных работников, а это пережитки прямой технологии. До сих пор мы заказываем системы «под ключ», хотя такие представления не дотягивают даже до объектных. Но в процессе развития техники и компьютерных технологий понимание компьютерной пирамиды, развивающейся в недрах пирамиды административной, постепенно становится все более и более определенным. Жизнь показывает, что далеко не все стратегические подходы к авто-

матизации являются эффективными и просто жизнеспособными.

### **Поколения АСУП**

Очередное поколение компьютеров создавало предпосылки для применения все более сложных технологий, а они, в свою очередь, стимулировали качественный рост технической и инструментальной базы. В результате, на логику технологического развития повлияли особенности эволюции техники, и реальный путь АСУП получился более сложным и зигзагообразным, чем в теоретических рассуждениях.

Мы будем называть поколением АСУП устойчивую ситуацию в развитии автоматизированных систем, когда существует определенный уровень развития техники, когда старая система управления выставляет определенные задачи на автоматизацию, когда разработчики находятся на определенном уровне развития, и известная часть механизмов развития входит в понятие автоматизированной системы.

На первых порах ограничителем поколения являлась техника, не позволяющая осуществлять смелые замыслы разработчиков. На последних этапах техника усилилась, и теперь сдерживающим фактором становится методическое отставание разработчиков (которых, впрочем, нельзя винить за то, что они в течение 10 последних лет не прошли путь, который был освоен в промышленности за столетия).

В каждом поколении имеются лучшие образцы, квалифицированные разработчики, грамотные пользователи. Они обозначают верхнюю границу поколения, достижимую в рамках организационно-технических возможностей. Все пространство под ними заполнено другими, менее удачными, разработками и примерами неэффективного использования неплохих систем. Это естественно, и причины такого положения лежат за рамками информационных технологий и информационных поколений. Частично их также можно объяснить бурным ростом АСУП, за которым не успевает массовое обучение и пропаганда передовых методов.

## 7.2 Вычислительные центры

К первому поколению АСУП мы отнесем целый класс технических средств и технологий, связанных с централизованной обработкой информации в пакетном режиме.

### Характеристика системы

Работники управления увидели в первых компьютерах средство, с помощью которого можно облегчить выполнение имеющихся задач. Каждый работник попытался сделать заявку на автоматизацию, но возможности компьютеров первого поколения были не столь велики, чтобы реализовать полный спектр заявок. Время подготовки данных и решения задач на вычислительном центре составляло от нескольких часов до нескольких дней. Поэтому в АСУП были автоматизированы устоявшиеся задачи нормативно-справочного и расчетного характера, не связанные с оперативной обработкой данных.

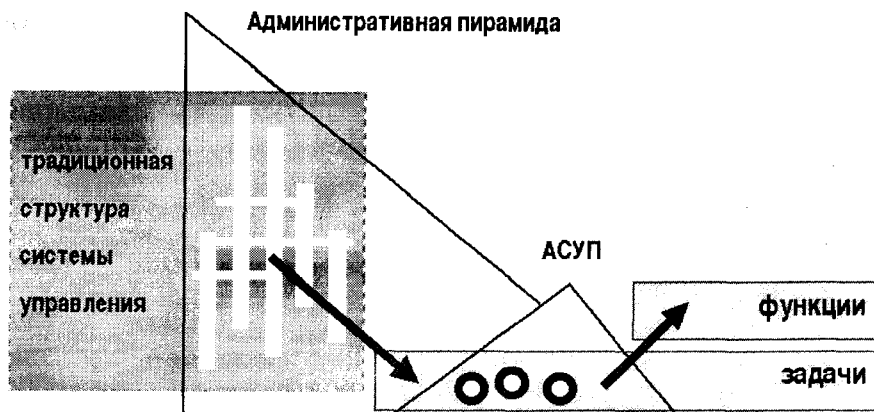


Рис. 44. Поколение вычислительных центров

С точки зрения большой технологии, первые компьютерные системы еще не были системами как таковыми. Они представляли собой множество отдельных задач, привязанных к традиционному каркасу. Разработчикам не потребовалось долго искать актуальные задачи и заниматься их компоновкой, поскольку в старой системе управления существовал перечень решаемых задач и структура их взаимодействия (рис. 44).

Из-за ограниченных возможностей техники первые системы не дотянулись до массового решения быстро развивающихся оперативных задач, что и позволило им сохранять логическую стройность – стройность аналитической системы верхов. Отношения с пользователями были организованы таким образом, что аналитико-синтетические процессы оказались за пределами компьютерной системы и по-прежнему осуществлялись силами работников предприятия. Отдел АСУ не занимался развитием единой платформы, он получал готовое решение от администрации и корректировал систему, воспринимая ее как постоянную конструкцию.

Общая характеристика первого поколения АСУП заключается в том, что его исходной базой стали развитые аналитические представления о структуре (эту работу проделала традиционная система), а главным методом стал стихийный синтез, характерный для прямой технологии. Базовая структура была принята как данность, а ее отголоски до сих пор живут в АСУП в виде стереотипных представлений.

#### **Традиции**

На предприятии хорошо прижились автоматизированные справочники продукции, материалов, технологические нормативы и многое другое. В дальнейшем они широко использовались в виде распечаток, а также как входные файлы для персональных компьютеров. Они стали фундаментом системы, на котором существуют представления о подсистемах и комплексах. Конкретный перечень каркасных понятий и подсистем далек от идеала, но их появление подготовило почву для следующей волны.

Вместе с тем, в сфере АСУП возникли первые традиции, воспринимаемые разработчиками как глобальные законы. В дальнейшем они проявились в определенной инертности мышления, что сыграло свою положительную, но еще в большей степени отрицательную роль. Именно в первом поколении зародились традиции тотального анализа, неспешной разработки и построения жестких, неразвивающихся комплексов, то есть те традиции, которые потом пришлось ломать, и влияние которых до сих пор ощутимо.

## **Результат**

Суммарная технология управления в эпоху первого поколения АСУП выглядела как все та же старая система, часть функций которой осуществлялась при поддержке вычислительного центра. АСУП общалась с предприятием через посредника – административную систему управления. Развитие полностью находилось в компетенции старой системы.

По мере роста объема задач возникли проблемы адаптации к изменениям, которые со временем выросли в глобальную проблему и привели к настоящему кризису. Увеличение объема программ привело к нелинейному росту количества связей между ними, которые необходимо было согласовывать, поэтому на определенном этапе АСУП достигли своего потолка, и дальше развиваться не смогли. Начались поиски новых информационных технологий.

### **Переход к функциям**

Как только в системе появились автоматизированные задачи, начались проблемы чисто компьютерного характера, связанные с адаптацией к меняющейся обстановке. Попытки обратиться за решением к старой системе ни к чему не привели, поскольку опыт работы с компьютером в ней отсутствовал.

Начался поиск решения, проводившийся силами АСУП методом стихийного синтеза. При этом рядом находилась традиционная система, в которой существовал своеобразный функциональный уровень. Несмотря на то, что традиционные решения не могли быть взяты за основу, они послужили прототипом, «подсказавшим идею», и в АСУП появились функции.

На последних стадиях существования вычислительных центров мы видим появление функционального подхода, сущностей, систем управления базами данных (СУБД), а также других атрибутов функциональной технологии. Результатом стихийного синтеза задач стало осознание функциональной схемы системы. В процессе эволюции выжили только те задачи, которые в какой-то степени соответствовали ей. После того, как этот результат был осознан, он лег в основу построения АСУП. Начался этап функциональной технологии.

## **Резюме**

Следовательно, первое поколение АСУП – это административная система, в которой компьютеры использовались в основном для решения устойчивых задач, вытекающих из старой аналитической схемы. Вместе с тем, внутри компьютерной системы методические представления находились на уровне прямой синтетической технологии.

В этом своеобразном сочетании кроется причина многих проблем АСУП первого поколения. И если говорить о достоинствах, то они определяются каркасным содержанием, пришедшим из старой системы. До тех пор, пока компьютерная система не сформировала свои механизмы развития, старая система играла роль своеобразного манежа, оберегающего АСУП от наиболее грубых ошибок.

### **7.3 Персональные компьютеры**

Персональный компьютер приблизился к пользователю и создал технические условия для решения оперативных задач, что породило вторую волну автоматизации предприятия.

#### **Динамические задачи**

Персональный компьютер появился в условиях, когда на вершине АСУП уже сформировалось представление о функциях, а внизу существовала масса нерешенных актуальных задач. В рамках первого поколения АСУП пользователь не имел возможности реализовать все свои заявки, поскольку они ограничивались возможностями техники. Персональные компьютеры сняли это ограничение и способствовали автоматизации самых разных задач и процессов. Компьютерная волна прокатилась практически по всей системе управления.

Новое техническое средство сыграло роль бомбы, взорвавшей здание, с таким трудом возведенное в предыдущие десятилетия. Вместо использования технологии, выращенной в рамках АСУП и приблизившейся к функциональным представлениям, разработчики снова перешли к прямой технологии. Вместо единой системы на предприятии начали стремительно развиваться индивидуальные платформы бюро и отделов. Пользователи организовали спрос на локальные задачи, появилось



предложение со стороны программистов, которые не заботились о логике развития предприятия и необходимости автоматизировать единую платформу.

Начало было прекрасным. От формализации структур и функций элементов быстро перешли к созданию объединенных конструкций, но чем дальше, тем труднее решались вопросы взаимодействия. Постепенно они стали главными, и с ростом размеров локальных автоматизированных пирамид дальнейшее их развитие стало практически невозможным из-за технико-экономической нецелесообразности.

### **Результат**

Программы, разрабатываемые для персональных компьютеров, отражали прежде всего точку зрения руководителя бюро и рядового исполнителя. Мы знаем, что даже в развитой системе управления они стоят на функциональных позициях и не поднимаются выше. Естественно поэтому, что программы для ПК в подавляющем большинстве строились на прямой и функциональной технологии. Четко просматривается итерационный характер разработки, отсутствие единой документации (а нередко отсутствие вообще какой-либо документации). Низкоквалифицированные разработчики вообще развивали программы методами стихийного поиска.

С появлением персональных компьютеров пользователь получил свободу действий, а вместе с ней и ответственность за принимаемые решения. Разработка комплексов для отдельных рабочих мест привела к тому, что часть работников получила автоматизированное средство, отражающее ее представления. Комплексы стали развиваться, постепенно захватывая все новое пространство, распространяя на него свои идеи, но тут столкнулись с сопротивлением соседей, которые делали то же самое. Волны преобразований, зародившиеся сразу во многих точках, натолкнулись друг на друга и образовали заторы.

### **Стыковка с большой машиной**

В целом ряде случаев удалось «дотянуться» до старой системы, функционирующей на вычислительном центре, что создало новый импульс для ее использования и модификации (рис. 45). Но такое взаимодействие потребо-

вало от постановщиков изворотливости, во много раз большей, чем обычная гибкость руководителя среднего звена [19].

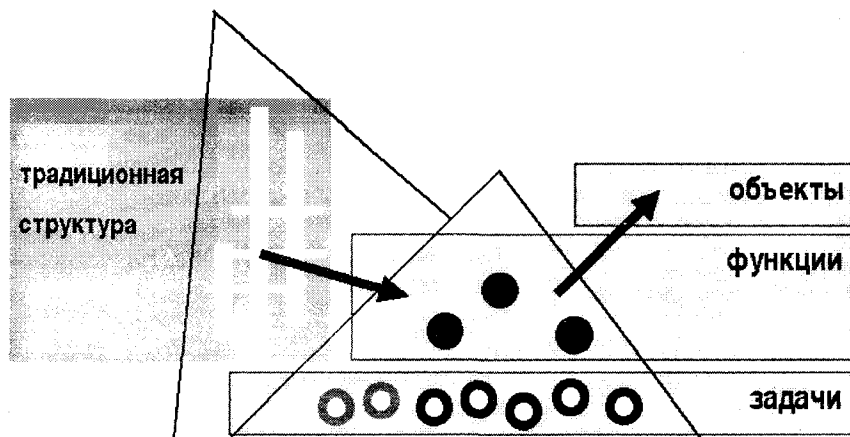


Рис. 45. Поколение персональных компьютеров

Однако именно стыковка систем первого и второго поколения позволила обнаружить, что старая схема, при всех ее недостатках, обеспечивает взаимодействие подсистем гораздо лучше, чем логически обоснованная функциональная модель. Система персональных компьютеров сгруппировалась вокруг больших машин. Административная система повторилась в уменьшенном виде в компьютерном отделе, где возникли бюро и группы, соответствующие подразделениям предприятия и обслуживающие их.

#### **Суммарная технология**

Появление персональных компьютеров не было случайностью. Синтез простых задач в недрах большой машины привел к функциональной схеме, от которой начался аналитический центробежный процесс. Система потребовала расчленения единой платформы на частные платформы, приближенные к пользователю. Сначала появились модули, затем выносные терминалы и, наконец, персональные компьютеры.

Функциональная схема во многом похожа на административный каркас, но не повторяет его. Это слабая, но

чисто компьютерная аналитика, отличающаяся от старого каркаса. Заметим, что в классических работах по функциональному анализу разбиение на подсистемы не совпадает с административным разбиением. На практике, однако, четкого различия не было, и в функциональную структуру перекочевали многие представления старой схемы управления.

Персональные компьютеры наполнили аналитическую функциональную модель синтетическим содержанием. Огромная масса задач подверглась формализации. Там, где уже существовал аналитический каркас, персональные комплексы использовали его, и в целом не нарушали принятой структуры системы.

#### **Развитие системы**

С этого момента функции развития системы начали переноситься из подразделений в отдел АСУ. Программные комплексы не подлежат изменению силами пользователя, поэтому в подразделениях работали старые механизмы развития, а в отделе АСУ начали складываться новые. Выполнение рабочих процедур существенно упростилось, но зато усложнился механизм корректировки и развития системы. Существенный недостаток технологии второго поколения – использование двух информационных моделей, то есть старого каркаса в административной системе и функциональной модели в отделе АСУ.

Но главной проблемой второго поколения стала нестыковка данных при взаимодействии функциональных модулей. Функциональный подход расчленил систему таким образом, что потерялись горизонтальные связи, нарушилась общность в развитии единых понятий. Каждое из них присутствовало на функциональных площадках, но в виде копии, никак не связанной с другими копиями.

#### **Переход к объектам**

Совершенно естественным выходом из создавшейся ситуации стала разработка новой технологии. Для стыковки функций начали использовать объектные представления. Объект стал основным элементом системы, и это проявилось в том, что независимые функции получили статус процедур, привязанных к конкретным объектам.

Пытаясь состыковать различные представления пользователей, разработчики переместили свое внимание с функций на сущности, и начали их активное развитие. Появились методы оптимизации документопотоков, создания эффективных баз данных и приведения структуры информации к третьей нормальной форме [1, 14]. В конце этого периода возникли объектно-ориентированные языки и методы объектного проектирования.

#### **7.4 Серверные системы**

К третьему крупному поколению АСУП мы отнесем системы, построенные на базе технологии «клиент-сервер» и ее многочисленных вариаций.

##### **Новая техническая база**

Персональные компьютеры представляют собой идеальное средство для реализации функций, но переход к объектам создал определенные проблемы. Дело в том, что объект является общесистемным понятием и не входит в компетенцию какого-либо подразделения или руководителя. С объектом работают несколько функций, несколько пользователей. Общесистемные интересы привели к появлению общесистемного средства ведения объектов – к серверу.

К моменту появления технологии «клиент-сервер» большие машины еще продолжали использоваться в качестве центров справочного характера, вокруг которых группировались персональные компьютеры. Поэтому может показаться, что «клиент-сервер» – это новая техническая база, возникшая путем объединения мощной центральной машины и персональных компьютеров на местах. На самом деле картина несколько тоньше.

Первое поколение АСУП, в котором главенствовали каркасные представления старой системы, получило статус устаревшего. Если возникала необходимость перевода информации с большой машины непосредственно на сервер, то ее структура при этом кардинально перестраивалась. Сервер ни в коем случае не является повторением большой машины, поскольку он создавался для согласования функций, а не для хранения переменных и массивов.

На практике отделы АСУ в организационном плане еще не были готовы к появлению такого мощного сред-

ства, поэтому в его освоении на первом этапе мы видим две традиционные волны. Одна идет сверху, как пережиток прямой технологии, и в ней четко прослеживается попытка сохранения единой жесткой схемы под руководством главного проектировщика. Другая – снизу, когда рабочая станция решает проблемы подразделения, а сервер используется для обмена информацией или для хранения наиболее очевидных и постоянных справочников.

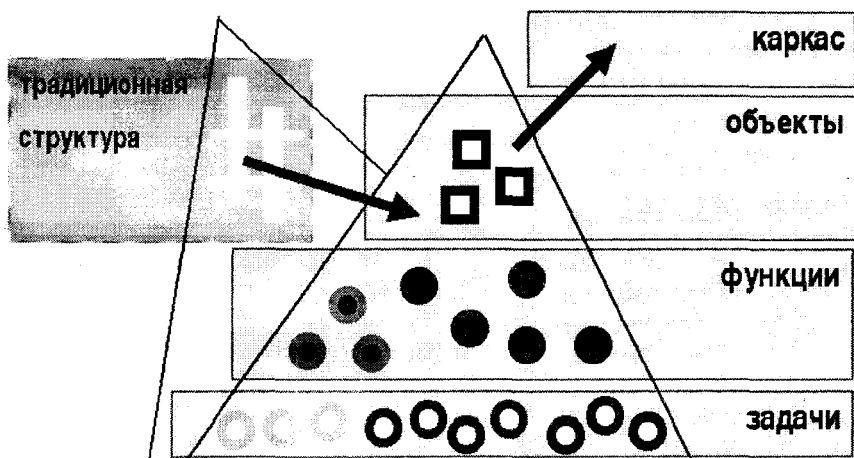
#### **Решаемые задачи**

Разработка программ для пользователей во многом сохранила технологию, сформированную во времена автономных персональных компьютеров. Сохранились фазы обследования, моделирования, проектирования, основанные на выяснении структуры данных и алгоритмов, которые используются либо на одном рабочем месте, либо на нескольких рабочих местах. Сервер, который был призван навести объектный порядок в системе, начал с копирования функционального беспорядка, а потом уже стало поздно что-то менять.

Как видим, отсутствие теории управления предприятием заставило разработчиков копировать докомпьютерную схему управления, построенную в расчете на человека. На уровне прямой и функциональной технологии новая схема еще похожа на старую, но на среднем уровне пирамиды, в зоне столкновения аналитических и синтетических потоков, к нестыковкам анализа и синтеза добавились нестыковки компьютерных систем различных поколений, а позже – нестыковки технологий. Проблема административной системы не просто повторилась, она еще и умножилась сама на себя (рис. 46).

#### **Новая пирамида**

Пытаясь объединить каркас старой системы и функциональные модули, разработчик создает модель, которая наиболее полно отражает представления среднего слоя управления. Он пытается представить всю систему в виде взаимоувязанных модулей, построенных на объектах и вытекающих из общей макросхемы, и по этой причине его модель в верхней части сводится к единой вершине. По крайней мере, разработчику хотелось бы свести ее к пирамиде.



**Рис. 46. Поколение серверов**

Исходная пирамида управления имела проблему – ее аналитика, так ярко выраженная на верхних уровнях, постепенно вырождалась на среднем уровне и не доходила до основания пирамиды. Ее синтетика, успешно работающая на уровне первичных модулей, не достигала верхушки пирамиды. Искусственная модель объектной технологии опирается на понятия среднего уровня, поэтому синтетические и аналитические потоки смогли распространиться и захватить всю пирамиду. Но при этом возникла неувязка – такая модель является искусственной, созданной разработчиком, а пользователь продолжает использовать свою, естественную модель.

#### **Две технологии**

Программный комплекс функционально вливается в административную систему управления, но полной его передачи пользователю все равно не происходит. Комплекс требует сопровождения, которое пользователь обеспечить не в силах. Созданная разработчиком модель не входит в общую понятийную платформу управления, а используется только аналитиками и программистами. При необходимости внести коррективы в схему управления задача формулируется пользователем, затем она переводится на язык аналитиков, интерпретируется в плоскости информационной модели, где и решается. Ре-

зультат проходит обратный путь и усилиями пользователя привязывается к общей понятийной платформе, после чего устраняются все обнаруженные ошибки. Таким образом, существующая технология предполагает решение трех задач – перевод задачи на язык программиста, решение задачи в плоскости информационной модели и обратный перевод.

Возник парадокс, который известен каждому, но осознан лишь немногими. Пытаясь формализовать схему управления предприятием, аналитики начали решительную борьбу с неформализованными процедурами управления, для чего и создали пространство информационных моделей. Но сам переход в это пространство и из него оказался неформализованным. Для того, чтобы решить реальную проблему управления, аналитики каждый раз планируют свои действия по модификации системы, то есть придумывают алгоритмы на ходу. В системе остались неформализованные процессы, они лишь сместились из сферы рабочих процедур в сферу развития.

Модель постепенно обрастает частными решениями, и наступает момент, когда теряется ее маневренность, и самой системе грозит гибель. АСУП превращается в монстра, который невозможно ликвидировать, ибо многие рабочие процедуры уже не выполняются в ручном варианте. Но и эксплуатация становится проблематичной, поскольку в системе существует слишком много компромиссных решений. Требуется коренная чистка, и ее периодически проводят, разрабатывая заново всю систему или значительную ее часть.

#### **Переход к каркасу**

Создание мощных и эффективных объектов идет успешно до тех пор, пока не встает вопрос об их объединении в один большой объект «предприятие» или, если это невозможно, в макросхему, где виден перечень основных объектов и схема их взаимодействия. Мы уже исследовали этот вопрос подробно и выяснили, что с появлением объектов начинается период синтеза, то есть постепенного согласования отдельных объектов в рамках макросхемы, называемой каркасом.

## **7.5 Новые поколения**

Попробуем увидеть в технике и архитектуре современных систем зачатки новой организации АСУП, которые пока еще не обособились в самостоятельное поколение.

### **Новая техника**

Трудно прогнозировать развитие технических средств, но понимание дальнейших технологических шагов позволяет сделать осторожные предположения следующего характера.

Во-первых, централизованное хранение информации имеет свои плюсы и минусы. Плюс заключается в быстром доступе и возможности реорганизации структуры. Минус – в громадных объемах, которыми необходимо управлять. В развитых системах существуют многосерверные конструкции. Это следующая волна рассредоточения, возникающая подчас вынужденно, и причину здесь следует искать в объеме данных.

Нетрудно заметить, что в случае разнесения системы на несколько серверов лучшим признаком декомпозиции являются представления о рабочих площадках, вытекающие из каркасной технологии. Но, по логике вещей, после рассредоточения на этапе каркаса должно начаться объединение в рамках формальной технологии. С этой точки зрения интересным является направление однородных вычислительных систем, в котором общая задача решается множеством небольших вычислительных устройств, объединенных в коллектив вычислителей [7].

Учитывая сказанное, выскажем предположение, что техническая конфигурация автоматизированной системы постепенно будет приближаться к структуре, имеющей нечто общее с человеческим организмом. Сегодня в ней есть два уровня – сервер и рабочая станция. Со временем их должно стать больше, и на каждом уровне появятся специализированные «органы».

### **Упорядочение**

Синтез объектов протекает довольно безболезненно, если его проводит один разработчик. Его макросхема может быть удачной или нет, но одно бесспорно – разработчик четко представляет себе перечень объектов и



взаимоотношения между ними. Этого нельзя сказать о группе разработчиков, в которой имеется несколько локальных списков объектов.

Некоторый порядок все же удастся соблюсти, поскольку в старой системе имеются понятия, являющиеся прототипами объектов. Есть, например, структура цеха и справочник продукции. Но старый каркас не может заменить макросхемы, поскольку объекты в нем недостаточно формализованы, а список их не установлен. В результате возникают общие представления в главных вопросах, но целое множество расхождений в деталях.

В информационной системе появляется своя макросхема, являющаяся своего рода каркасом, в которой делается попытка закрепить общие представления об используемых объектах. Беда только в том, что макросхема не содержит всей необходимой информации, и очень быстро стареет. Проблемой является и актуализация макросхемы. Выход может быть только один – возможность непосредственного подключения к спецификации объектов. Для этого требуется ввести общий список, механизмы согласования, механизмы обновления информации. Попытка сделать это в лоб быстро приводит к профанации идеи и прекращается.

#### **Проблема порядка**

Собственно, проблема заключается в том, чтобы постоянно контролировать список объектов, знать автора разработки или лицо, обладающее полной компетенцией на модификацию структуры (чтобы было ясно, где находится первичная информация). Кроме того, нужно знать, где расположена эталонная версия объекта, чтобы иметь возможность подключаться к нему или копировать его в свои модули.

Совершенно очевидно, что проблема носит динамический характер. Если бы речь шла об уже готовых и сданных в эксплуатацию объектах, привести в порядок их список было бы не так сложно. Но если речь идет о разработке новых объектов, то здесь ситуация меняется каждый день, а самое главное – ведущий разработчик не спешит с обозначением схемы объекта, справедливо полагая, что это ограничит ему свободу маневра. Если сегодня показать предварительную схему, то завтра выяс-

нится, что на нее уже сориентировались партнеры, и отыграть назад будет непросто. Отсюда разработчик предпочитает придерживать оперативную информацию.

Как следствие, возникают дублирующие друг друга разработки. Необходимость в конкретном объекте возникает сразу у нескольких разработчиков, и каждый из них начинает прорабатывать свою версию, не оповещая коллег о принимаемых решениях. Если бы в результате появлялись абсолютно одинаковые схемы, то тогда мы просто теряли бы силы на параллельную разработку. Но реальный результат еще хуже – появляется целое множество объектов, очень похожих, но не идентичных, и теперь их нужно собирать в единое целое.

### Выход

Проблемы такого рода будут продолжаться до тех пор, пока компьютерная пирамида не завершит строительство своей верхушки, то есть не дойдет до каркаса и типовых наборов элементов. Только в этом случае информационная структура получит четкое описание, а проблемы развития сместятся в какую-либо другую плоскость (рис. 47). Мы не будем рассматривать этот

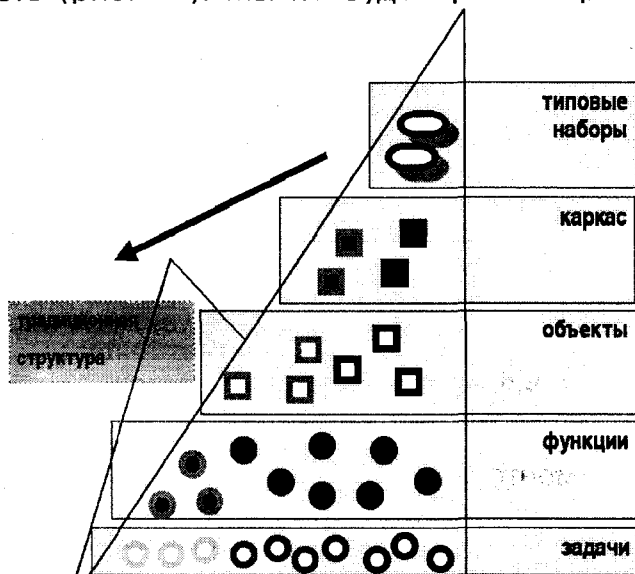


Рис. 47. Новые поколения

процесс подробно, чтобы не уйти от исследования настоящего к проектированию будущего.

### **Две вершины**

Нет сомнения в том, что компьютерная пирамида со временем вырастает в более мощную конструкцию, чем старая система управления. Если на первых порах она использует результаты и представления административной системы, то в развитой АСУП сама административная система опирается на информационные понятия и средства. Где-то на уровне объектной технологии происходит «смена лидера», и происходит она не без проблем.

Проблемы возникают как у старой, так и у новой системы. Для разработчиков, которые привыкли обращаться к пользователям и сдавать результаты своей работы пользователям, трудности связаны с психологическим барьером, который необходимо преодолеть. Необходимо взять ответственность на себя, и уже не выполнять пожелания пользователя, а вести его за собой. Не углубляясь в детали этого процесса, отметим, что здесь имеется много эмоционального.

Пользователь также проходит через психологический барьер. Оценив преимущества новой системы, он тем не менее чувствует, что ситуация выходит из-под контроля, и теперь не он руководит системой, а система руководит им. Правильнее было бы сказать, что система руководила им всегда, но компьютеры выявили и наглядно показали пользователю, где и как она это делает. Пользователь действительно оказался под властью, но не разработчика, а системы, системных законов. Разработчик сыграл роль посредника.

Так или иначе, но «смена лидера» проходит болезненно для всей системы, и на этапе объектной технологии наблюдается торможение, кризис, брожения умов, авантурные решения и многие другие атрибуты двоевластия. И чем быстрее будет внедрен каркас новой системы, тем лучше, не говоря уже об использовании типовых наборов элементов.

### **7.6 Развитие системы**

Подводя итог развитию технологий, мы замечаем определенную регулярность в развитии аналитико-синтетических процессов.

### Технологический цикл

На уровне каждой технологии мы начинаем с некоторого базового понятия, сформированного в результате применения предыдущих технологий. Это может быть элемент, функция, объект. Базовые понятия позволяют построить систему, но при этом возникает какая-либо сопутствующая проблема, и в рамках технологии предлагается новое понятие, решающее эту проблему. В дальнейшем выясняется, что решение одной проблемы приводит к появлению другой, и цикл повторяется.

Например, старая система вычленила задачи для решения на компьютере. Возникла проблема согласованного решения задач, которая привела к идее функций и функциональной структуры. Затем анализ функциональной модели привел к понятию объектов. Необходимость согласованного развития объектов – к каркасу, а его анализ – к типовым наборам элементов.

### Чередование

Бросается в глаза чередование синтеза и анализа как основных технологических методов (рис. 48). И в этом нет ничего удивительного, так как очередная технологическая мысль, идущая от интересов системы сверху вниз, или от интересов элементов снизу вверх, должна была дойти до конца, и при этом не могла развернуться на 180 градусов.

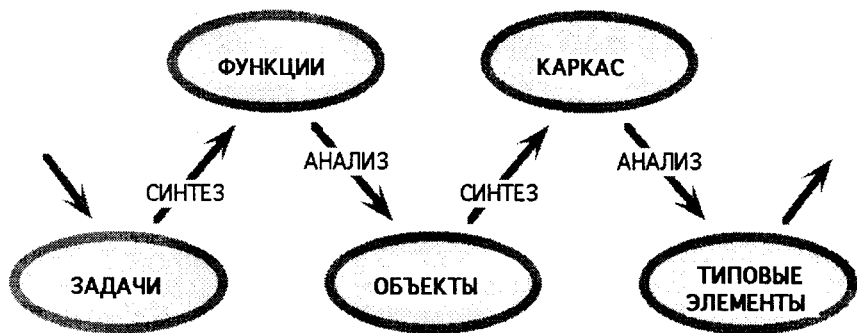


Рис. 48. Аналитико-синтетическая канва развития

Если мы применили аналитическую технологию и получили новые представления, то проведение нового цикла анализа должно привести к уже имеющемуся резуль-

тату (если перед этим работа была выполнена качественно). Что же касается синтетического метода, то новые понятия еще не подвергались рассмотрению с этой точки зрения, и здесь заложены определенные резервы.

Чередование синтеза и анализа напоминает колебания маятника и показывает их взаимодействие. Процесс анализа заканчивается появлением нового типа элементов, поэтому синтез не является процедурой возврата к центральной схеме. Это совершенно новый и равноправный этап, заканчивающийся получением конструкций нового типа.

### **Прогресс**

Пока неясно, в какой степени полученный результат может быть общим, но в результирующей схеме прослеживается три этапа синтеза, и каждый раз речь идет об элементах определенного типа.

Сначала синтезу подвергаются задачи, а результатом являются функциональные модули. Речь идет о функциях, о действии. Затем синтезируются структуры информации, то есть объекты. На третьем витке речь идет уже об исполнительных элементах, о физической реализации. Впрочем, третья фаза относится уже к новому витку эволюции.

С другой стороны, анализ осуществляет переход от одной плоскости к другой. Начиная от функций, он приводит к структуре данных. Начиная от каркаса, то есть от данных, он доводит дело до реализации, то есть до функций.

Просматриваются две плоскости – функций и данных. Оптимизация в одной из них осуществляется синтетическим путем. Появляются новые элементы, которые путем анализа порождают изменения в другой плоскости.

### **Практические отклонения**

Выше мы описывали пропорциональный процесс развития, в котором события развивались на основе логики. Но существуют внешние соображения, которые также накладывают отпечаток на развитие системы. Если говорить о предприятии, то это может быть финансовый кризис, недостаток квалифицированных кадров, субъ-ективная позиция руководства и многое другое.

Поэтому описанный выше процесс на практике развивается с вариациями, которых, в принципе, не так много.

Дело в том, что в развитии системы всегда участвуют два процесса, поддерживающие и подкрепляющие друг друга. Синтетические процессы оказывают давление на аналитическую структуру, если даже она сама не пытается развиваться. Содержание изменяет форму, ломает ее путем давления изнутри. С другой стороны, форма также изменяется, чем стимулирует развитие содержания, создает для него благоприятные или неблагоприятные условия.

### **Вертикальная картина**

До сих пор мы предполагали, что понимание кризиса и желание выйти из него возникает во всей пирамиде управления одновременно. Так бывает, но бывает и иначе, когда одна часть работников активно стремится к реорганизации, а другая по каким-то причинам остается пассивной. В этом случае возникает вертикальный дисбаланс в структуре. Если первым за реорганизацию берется директор, то верхушка единой платформы становится активной. Это усиливает каркас системы и стимулирует процессы развития. Пирамида как бы растет вверх, центр тяжести приподнимается, теряется устойчивость, возникает опасность «опрокидывания».

В случае, если реорганизация начинается снизу, в индивидуальных платформах возрождается логическая схема, появляются альтернативные предложения по выходу из кризиса, которые не находят быстрого воплощения из-за пассивности руководства. Тем не менее, в результате инициативы снизу аналитические процессы активизируются, и при усилении кризиса эффективные решения, хотя и с опозданием, поступают на реализацию. В этом случае пирамида управления как бы расплывается в горизонтальном направлении, что повышает ее устойчивость, но существует опасность, что при наличии более рослых конкурентов систему могут «затоптать».

Активность среднего слоя пирамиды приводит к децентрализации и распаду системы на составляющие, если только в среднем слое имеется несколько лидеров, примерно равных по силе. Если же лидер один, и у него нет желания «делать карьеру» и становиться во главе предприятия, он может остаться в прежнем статусе, оказывая влияние во всех направлениях – вверх, вниз и по

среднему слою. Такая ситуация часто приводит к появлению теневых структур, то есть к появлению у пирамиды дополнительной «верхушки», растущей из ее середины.

### **7.7 Автоматизированные рабочие места**

Постепенный перенос механизмов управления в компьютерную систему изменяет роль человека в системе управления.

#### **Традиционная подготовка**

Мы уже говорили о том, что работники управления попадают в старую систему после соответствующих учебных заведений. Присовокупляя сюда среднюю школу и обучение в дошкольном возрасте, мы видим длительный и сложный процесс обучения, предназначенный для подготовки работника управления.

Нетрудно представить, что по всей программе обучения здесь и там разбросаны понятия, формирующие представление о старой структуре управления, о способе решения производственных задач, о принципах подчиненности, о механизмах решения проблем, и так далее. Совершенно очевидно, что до появления компьютеров прошло немало времени, и все эти традиционные представления успели глубоко проникнуть в систему подготовки кадров.

#### **Разработка**

Мышление разработчика отличается от мышления рядового управленца. Разработчик должен понимать суть объектного и каркасного подхода, видеть аналитико-синтетические потоки. Несмотря на то, что эти представления довольно сложны, проблем с обучением разработчика не так уж много, поскольку он получает эти знания как основные, а система подготовки разработчиков сконцентрирована и доступна для контроля и модификации.

Разумеется, и у разработчиков возникают проблемы, связанные с недостаточно высоким уровнем понимания своих задач и профессиональной подготовкой, но это характерно для любой системы образования и для любого учащегося. Это, скорее, вопрос прилежания, чем вопрос оптимальности программы обучения. Одним словом, обучение немногочисленной профессиональной группы разработчиков не является серьезной проблемой.

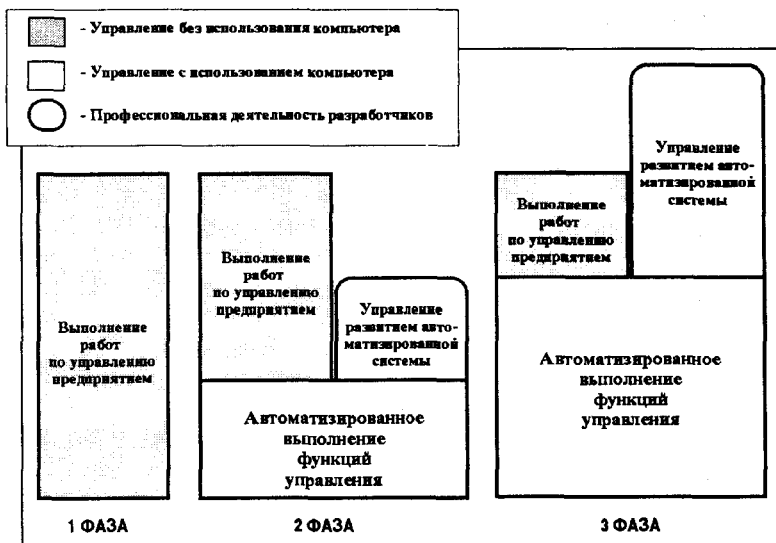


Рис. 49. Смена задач у работников управления

### Пользователи

По мере внедрения компьютерных комплексов пользователь не только пересаживается за компьютер, но и меняет понятийную базу. От документов переходит к файлам, от задач – к функциям. В отличие от профессионала, его задача гораздо скромнее. Ему не нужно искать новых решений, достаточно эксплуатировать уже готовые комплексы. Тем не менее, эта эксплуатация связана с принципиально новыми представлениями, и их необходимо изучить.

Возникает необходимость корректировки глобальной системы обучения, причем таким образом, чтобы новый работник был готов правильно действовать в рамках компьютерной системы. Дополнение программы обучения новыми сведениями необходимо, не подлежит сомнению, и в принципе также не представляет трудности. Проблема заключается в исключении из системы обучения докомпьютерных представлений. Учитывая массовый переход пользователей в автоматизированное пространство управления, эта проблема становится весьма актуальной (рис. 49).



### **Уровни управления**

Не меньшей проблемой является универсальность человека как элемента управления. Каждый молодой специалист проходит свой путь эволюции, и далеко не все доходят до верхнего уровня. В старой системе управления работники распределены по уровням, соответствующим их назначению. Исполнитель работает на уровне элементов, руководитель отдела – на уровне конструкции, директор – на уровне всей системы.

К сожалению, все они собраны из одного и того же материала. Директор остается рабочим и иногда мыслит категориями элемента, а рабочий обладает интеллектом, позволяющим ему подниматься над локальными интересами. Дело осложняется еще и тем, что каждый из них является гражданином и мыслит за страну, а также выполняет функции элемента в семье. Эта деятельность оказывает влияние на общее мышление, в результате создается мешанина методов.

Компьютер идеально настраивается на определенный уровень, правда, он требует дополнительных механизмов для адаптации и настройки. Из компьютера можно сделать как элемент, так и конструкцию, а можно заставить его работать на интересы системы – все зависит от программирования. Более того, в компьютере возможно разместить совершенно различные математические конструкции, то есть смоделировать элемент, конструкцию и систему по различным правилам, и они никогда не станут подменять друг друга.

Это свойство оказывается самым заманчивым, но оно оборачивается целым рядом трудностей. Во-первых, формализация должна быть на высоте. Во-вторых, должна быть решена проблема отделения порождающих механизмов от исполнительных, что позволит реализовать механизм адаптации.

Выход может быть найден в комплексировании человека и компьютера, то есть в создании такого элемента, в котором часть функций возложена на человека, а другая часть – на компьютер. Более того, если мы собрались делать различные модели для элемента, конструкции и системы, нам следует собирать человеко-машинные комплексы по различным правилам. В принципе, уже

сейчас это заметно на существовании спецпрограмм для пользователей и программ для разработчиков. Появились и специальные программы для руководителей.

#### **Комбинированная система**

Следовательно, нам нужно еще раз пройти весь эволюционный процесс, но теперь с точки зрения человеко-машинных комплексов, развивавшихся от простых технологий к сложным. Именно они постепенно выросли в современную АСУП и развиваются дальше.

Эволюция АСУП требует корректировки обучения будущих разработчиков и пользователей буквально со школьной скамьи, и обучения не составлению программных кодов, а системных законов. В частности, основам каркасной и формальной технологии. Этот процесс, представляющийся сегодня далеким и фантастическим, осуществится в самые короткие сроки и повсеместно, как только появятся результаты внедрения АСУП новых поколений, как только конкуренция покажет, что предприятие должно либо перестроиться, либо умереть.

#### **Современные технологии**

Как видим, мы не отвергаем достижений современных информационных технологий. С единственной поправкой – они возникают не сами по себе, а как стихийные проявления процесса эволюции, много раз повторившегося в прошлом в других системах. Эти процессы привязаны к проблемам развития единой понятийной платформы.

Информатика генерирует новые технологии, которые стартуют от моделей и других информационных атрибутов. Эти технологии начинают работать в сфере производства и в основном гибнут. Выживают только те, которые соответствуют последовательному продвижению от прямой к формальной технологии.

Теперь мы также можем легко объяснить, почему технология, которая удачно сработала на трех предприятиях, приводит к самым плачевным результатам на четвертом. Причина в том, что современные технологии оставляют за скобками каркас старой системы. Они используют его, они развиваются под его управлением, но эти факты в самой технологии не находят отражения.

Если в одном случае мы имеем развитый и эффективный каркас, технология приводит к определенным ре-

зультатам. Повторив те же действия в условиях другого каркаса, мы не получаем приемлемых результатов. Следовательно, современные технологии выполняют только часть задачи, а другую часть по-прежнему решает старая система управления. И от того, как она сработает, во многом зависит успех всего мероприятия.

Мы предлагаем более простой путь. Вместо того, чтобы производить множество технологий «задом наперед», осознать происходящее и начать работать правильно – от предприятия, от единой платформы. Тогда все технологии, различающиеся деталями, будут иметь больше шансов на выживание. Собственно, специалистам необходимо заняться изучением вопросов, которые до сих пор оставались за рамками научного и инженерного круга и относились к «организации работ».