

## Модели функционирования экономики

УДК 338.26

ОРГАНИЗАЦИЯ ДАННЫХ В МОДЕЛЯХ ПРОЦЕССА  
ВЫПОЛНЕНИЯ ПЛАНА

С.М.Лавлинский

При создании машинных моделей экономических систем, ориентированных на научные цели, разработчики, как правило, основное внимание уделяют программной реализации самой модели, упрощая процедуры ввода-вывода информации и схему управления процессом вычислений. Разработка моделей, предназначенных для использования в процессе управления реальными экономическими объектами, требует более тщательного исследования системных вопросов, связанных с информационным обеспечением, реализацией диалоговых процедур, адаптацией и сопровождением программного комплекса. Опыт создания подобных систем (см. [1,2]) показывает, что наиболее трудоемким делом оказывается выбор рациональной структуры базы данных и разработка эффективной системы управления ею.

Целью настоящей работы является описание способа организации данных в моделях специального класса – моделях процесса выполнения плана [3], позволяющего построить машинную систему, пригодную для практического использования плановыми работниками при оценке принимаемых решений. К настоящему времени создано уже несколько модификаций базовой модели выполнения плана: отраслевого [4], регионального [5] и народнохозяйственного [6] уровней, при программной реализации которых удалось единообразно решить проблему организации данных<sup>1)</sup>.

<sup>1)</sup> Системы реализованы на языке Фортран-IV в ДЭС ЕС ЭЕМ.

Опыт практического использования разработанных машинных систем показал эффективность применения двух баз данных - вспомогательной и основной. Структура основной базы данных, предназначенной для оперативного обеспечения расчетов, непосредственно настроена на рекуррентную вычислительную схему моделей реализации плана и обеспечивает высокий уровень быстродействия системы. Вспомогательная база данных снабжена специальной системой управления, обеспечивающей удобство работы при первоначальной подготовке информации, и по мере наполнения является источником информации при формировании основной базы данных.

Конструктивные особенности использованного способа организации данных излагаются в последующих параграфах и могут быть применены при решении вопросов информационного обеспечения левых моделей с рекуррентной вычислительной схемой.

#### §1. Принципиальная схема модели процесса выполнения плана

В основу модели процесса выполнения плана положено представление о функционировании экономической системы (отрасли, региона, народного хозяйства) как совокупности динамически взаимосвязанных "элементарных" подпроцессов, отражающих различные стороны хозяйственной деятельности. Отдельные блоки модели отражают как функциональный разрез воспроизводственного процесса (производство, материально-техническое снабжение, капитальное строительство и т.д.), так и организационный - имитируют динамический процесс взаимодействия (по дням, декадам, месяцам) хозяйственных звеньев, преследующих в своих текущих решениях различные цели.

Использованный в модели способ формализации динамики взаимосвязей отдельных элементов системы основан на построении дискретного аналога непрерывного процесса функционирования в виде операторного уравнения, описывающего переход из одного состояния в другое:

$$Z^t = Q(Z^{t-1}, R^t, P^t), \quad t=1, \dots, T. \quad (I)$$

Здесь  $Z^t$  - вектор состояния на конец декады  $t$ , отдельные компоненты которого означают, например, объемы производства продукции и ее поставок потребителям для каждого предприятия системы, объемы выполненных работ различных видов на строительных

объектах и другие показатели, характеризующие воспроизводственный процесс;

$P^t$  - плановые задания для предприятий на декаду  $t$  ;

$R^t$  - прогноз ресурсной обеспеченности.

Оператор перехода  $Q$  формализуется в виде системы имитационных алгоритмов, описывающих процессы функционирования отдельных хозяйственных звеньев и их взаимодействие (материальные потоки в системе), а также различные процедуры принятия решений. Основная идея предлагаемого метода моделирования заключается в задании начального состояния  $Z^0$ , сценария развития (планов  $P^t$  и прогноза ресурсной обеспеченности  $R^t$ ) и последующем вычислении траектории развития системы  $\{Z^t\}_{t=1}^T$  на основе рекуррентных соотношений  $Q$ .

## §2. Структура основной базы данных

Использование рекуррентной вычислительной схемы в моделях реализации плана предъявляет весьма жесткие требования к быстродействию системы. В отличие от балансовых и эконометрических моделей при вычислении траектории развития экономической системы на основе соотношений (I) приходится последовательно и многократно обрабатывать многомерные информационные массивы, большую часть которых необходимо располагать на внешних носителях<sup>1)</sup>. В этих условиях структура базы данных должна выбираться с учетом особенностей вычислительной схемы и обеспечивать высокий уровень быстродействия программного комплекса.

Естественным путем уменьшения быстродействия машинной модели является минимизация суммарного времени обращения в процессе счета к внешним устройствам, содержащим значительную часть необходимой информации. Опыт разработки программного комплекса показал, что для оперативного информационного обеспечения вычислительного процесса наиболее целесообразно сгруппировать информационные массивы на магнитном диске по каждому предприятию в отдельную логическую запись, которую в дальнейшем для удобства будем называть рабочим файлом.

1) Использование детализированной номенклатуры в моделях реализации плана приводит к значительно большему по сравнению с известными экономико-математическими моделями совокупному объему перерабатываемой информации. Первостепенное значение в этих условиях приобретает проблема экономии оперативной памяти ЭВМ.

В состав рабочего файла для предприятия входят матрица производственных способов и другие сведения технологического характера, планы выпуска и поставок, вектор состояния предприятия; компоненты вектора состояния - объемы трудовых и материальных ресурсов, фонд полезного рабочего времени оборудования на данный момент, а также результаты производственной деятельности предприятия (объемы выпусков и осуществленных поставок). Рабочий файл строительного объекта состоит из паспортов объекта, фиксирующего его текущее состояние.

Рабочий файл содержит исчерпывающую информацию о предприятии и представляет собой последовательную запись фиксированной структуры на магнитном диске, начало которой совмещено с первым элементом зоны числового файла. Такое пространственное расположение рабочего файла на диске делает его файлом прямого доступа, время обращения к которому минимально. Номера зон, начиная с которых располагаются рабочие файлы предприятий и объектов, задаются извне модели и определяются экспертно на основе оценки величины рабочих файлов.

В машинной модели все вычислительные алгоритмы "настроены" на структуру рабочих файлов. При такой организации процесс функционирования моделируемой системы на уровне отдельной декады заключается в последовательной обработке рабочих файлов, поскольку участие предприятия в воспроизводственном процессе сводится к изменению компонент вектора состояния (элементов рабочего файла). При этом использование рабочих файлов позволяет минимизировать как число обращений к внешней памяти, так и время каждого обращения, а в оперативной памяти постоянно хранится лишь общесистемная информация.

Таким образом, оперативная база данных, используемая непосредственно в процессе расчетов, состоит из рабочих файлов, а система управления ею строится на основе операторов Фортрана READ, WRITE и обеспечивает высокий уровень быстродействия машинной модели процесса выполнения плана.

Каким образом может быть сформирована такая оперативная база данных? Ответ на этот вопрос неутрачен, поскольку рабочие файлы имеют весьма сложную структуру, настроенную на обработку имитационными алгоритмами, и мало приспособлены для того, чтобы их формировать "вручную" и использовать в качестве исходных информационных массивов системы. Существенным ограничением,

накладываемым на способ формирования оперативной базы данных, является также требование простоты контроля полноты и достоверности исходной информации. Наиболее целесообразно с этой точки зрения осуществлять контроль над сравнительно небольшими информационными массивами, несущими функциональную нагрузку, с фиксированной структурой и известной областью значений.

Учет всех этих требований привел к тому, что процесс подготовки информации для работы модели разбивается на два этапа. Первоначально заполняется вспомогательная база данных (БД1), в которой информация хранится произвольное время и используется лишь при окончательном формировании основной оперативной базы данных (БД2). На втором этапе из информационных массивов БД1 формируется БД2 - "склеиваются" рабочие файлы, используемые имитационными алгоритмами. Параллельно осуществляется программный контроль полноты и достоверности исходной информации.

### §3. Вспомогательная база данных

Автономный программный комплекс, реализующий систему управления БД1, выполняет определенный набор операций и непосредственно "настроен" на нужды модели функционирования. Так, в качестве базового объекта в БД1 принят относительно небольшой информационный массив, содержащий сведения определенного характера об отдельном предприятии, объекте (например, планы производства, планы снабжения, паспорт строительного объекта и т.д.). Программный комплекс спроектирован и реализован таким образом, что БД1 заполняется по мере подготовки информации порциями, содержащими произвольное число элементарных информационных массивов, и не требует одновременного ввода всего объема информации, необходимой для работы модели. Это создает существенные удобства при подготовке информационной базы, особенно удобные при частичной корректировке группы массивов, а также в условиях, когда информация различного характера поступает из разных источников. Процесс подготовки информации облегчается также предоставленной пользователю возможностью в любой момент времени распечатать на дисплее или АЦПУ как содержимое отдельных, уже занесенных в БД1 массивов, так и общий перечень имеющейся к данному моменту информации.

В БД1 применен следующий способ хранения информации. В чис-

ловом файле на магнитном диске отводится область, в которой располагаются все массивы БДІ. Все информационные массивы записываются последовательно друг за другом, без каких-либо промежутков. Такой способ хранения продиктован как необходимостью работы с большим совокупным объемом информации, так и большим числом элементарных массивов, не позволяющим каждый из них представить в виде файла прямого доступа.

Фактическое расположение имеющихся в БДІ элементарных массивов фиксируется в каталоге банка KATAL, записанном на диске - двумерной матрице размерностью  $(\bar{N} \times 5)$ , где  $\bar{N}$  - число массивов, для которых зарезервировано место в БДІ. Первые два элемента строки KATAL( $i_0, 1$ ) и KATAL( $i_0, 2$ ) фиксируют 4-символьный код и числовой индекс записанного массива с внутренним локальным номером  $i_0$ . Например, план поставок 4-го предприятия идентифицируется в БДІ как (PLPO, 4). Подобная структура (функциональная нагрузка, принадлежность) оказывается весьма удобной и достаточной для идентификации информации, используемой в модели.

Элементы KATAL( $i_0, 3$ ), KATAL( $i_0, 4$ ) определяют "координаты" массива на магнитном диске: первый указывает номер зоны числового файла, а второй - номер элемента этой зоны, начиная с которого зависит массив. Параметр KATAL( $i_0, 5$ ) соответствует числу элементов в массиве.

Принципиальная схема программного комплекса представлена на рис. I. Выделены следующие возможные режимы работы с БДІ:

- инициализация,
- запись массивов в БДІ,
- печать массивов из БДІ,
- печать каталога,
- расширение БДІ.

**Инициализация.** Начальный этап работы с БДІ: каталог заполняется нулями и записывается в отведенное на магнитном диске место (с первого элемента зоны  $NZ_1$ ). Следом за каталогом записываются номера зон  $NZ_2$  и  $NZ_3$ , фиксирующие отведенное на диске для БДІ место, и параметр NMAS - количество массивов, записанных к настоящему моменту в БДІ.

**Запись.** В программном модуле "ЗАПИСЬ" с магнитного диска считывается последняя версия каталога и последовательно вводятся идентификаторы массива, количество его элементов и сам массив (при этом указывается тип данных - целый или веществен-

ный - и используется соответствующий формат). После этого управление передается модулю LWRITE, осуществляющему запись в БД введенного массива.

Последовательно анализируя строки каталога, программа LWRITE проверяет, нет ли в БД массивов с аналогичными идентификаторами. Если нет, то массив фиксируется в каталоге и записывается следом за массивом, бывшим до этого последним, передвигая там самым метку конца записи МК в БД. Если оставшегося места не хватает для записи очередного массива, то на дисплее выдается соответствующее сообщение системы и никаких изменений ни в каталоге, ни в БД не происходит.

Если же в БД находится аналог введенного массива и количество элементов в новой и старой версиях совпадает, то массив записывается на старое место без каких-либо изменений в каталоге.

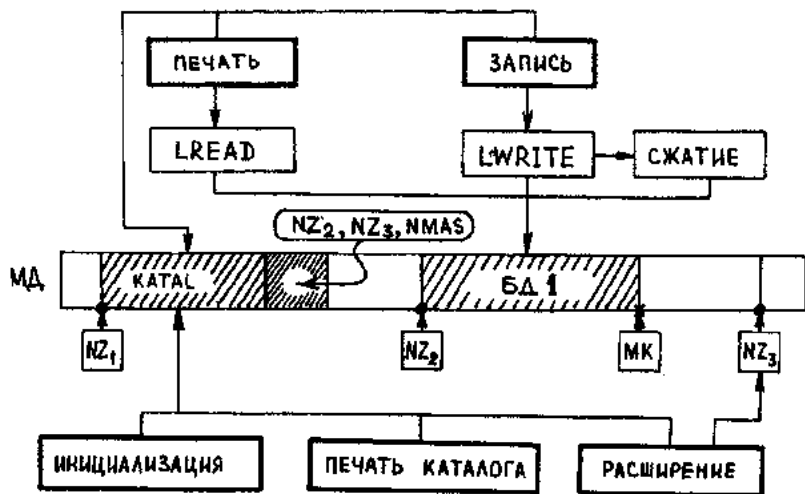


Рис. 1. Структура программного комплекса, реализующего систему управления БД

Значительно более сложная ситуация возникает, когда в БДІ имеется аналог введенного массива, но с другим числом элементов. В этом случае управление передается модулю "СЖАТИЕ", удаляющему старую версию и сдвигаящему записи остальных массивов способом, изображенным на рис.2 (с соответствующей корректировкой каталога). По завершению работы модуля "СЖАТИЕ" новая версия массива записывается в конец (начиная с метки МК).

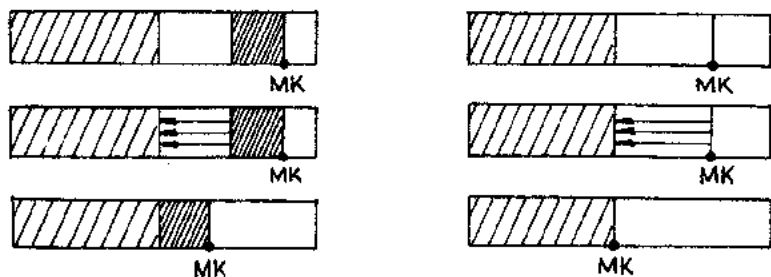


Рис. 2. Процедура сжатия

В процессе работы модуля "ЗАПИСЬ" может быть введено и записано в БДІ произвольное число информационных массивов, программа завершает работу при введении символического идентификатора вида "жжжж".

**П е ч а т ь .** При работе программного модуля "ПЕЧАТЬ" с магнитного диска считывается каталог, затем вводятся идентификаторы распечатываемого массива и указатель формата, который должен использоваться при печати элементов массива.

Модуль LREAD осуществляет поиск по каталогу массива в БДІ с заданными идентификаторами. Если такого массива в БДІ нет, то об этом выдается сообщение. Если же массив в каталоге найден, то программа находит его в БДІ, считывает и распечатывает по требуемому формату.

Работа программы завершается при введении символического идентификатора вида "жжжж".

**П е ч а т ь к а т а л о г а .** Программа считывает с магнитного диска и распечатывает последнюю версию каталога - матрицу KATAL.

**Р а с ш и р е н и е БДІ.** В случае, когда БДІ заполнен настолько, что не удастся записать очередной массив (после метки



конца МК), предусмотрена возможность расширения БДІ. При этом увеличивается на заданную величину номер зоны, ограничивающей БДІ справа ( $NZ_3$  на рис.1), и записывается на диске после каталога и параметра  $NZ_2$ .

В процессе работы с БДІ пользователь оперирует набором управляющих модулей, каждый из которых выполняет одну из описанных выше операций. Первоначально оценивается общий объем информации и планируется месторасположение на магнитном диске БДІ и каталога (номера зон  $NZ_1, NZ_2, NZ_3$ ), после чего используется режим "ИНИЦИАЛИЗАЦИЯ" (с входными параметрами  $NZ_1, NZ_2, NZ_3$ ). Результат работы — готовая к заполнению информационными массивами база данных.

Заполнение БДІ осуществляется в режиме "ЗАПИСЬ", при этом информация может вводиться как с дисплея, так и с магнитной ленты. В целях контроля содержимое занесенных в БДІ массивов может быть распечатано. Общий перечень информационных массивов, имеющихся к текущему моменту в БДІ, и их характеристики пользователь может получить в режиме "ПЕЧАТЬ КАТАЛОГА". Процесс ввода продолжается, пока вся информация, необходимая для работы системы, не будет записана.

#### §4. Формирование основной базы данных

Хотя во вспомогательной базе данных БДІ и содержится вся необходимая для модели информация, использование ее в процессе расчетов в качестве информационной базы нецелесообразно. Дело в том, что при оперативном использовании БДІ нам пришлось бы считывать и записывать по отдельности каждый из элементарных массивов БДІ, а это значительно увеличивает как общее число обращений к внешним устройствам, так и время каждого обращения: даже для такой сравнительно простой операции, как чтение, в БДІ каждый раз необходимо программно находить нужный массив, считывать занятые им зоны и лишь после этого заполнять рабочий массив, передаваемый в вызывающую программу.

Процесс формирования основной базы данных организован следующим образом. Предварительно производится оценка величины рабочих файлов и на ее основе распределяется свободное место на диске — для каждого рабочего файла  $i$  фиксируется номер зоны  $NZFILE_i$ , начиная с которой он будет располагаться. Затем работает программа "ФОРМИРОВАНИЕ", выполняющая следующие операции.

Для каждого предприятия (объекта)  $i$  заранее известны состав и структура рабочего файла – последовательность (  $AAAA_i, BBBB_i, CCCC_i, DDDD_i$  ) элементарных массивов, меняющаяся, вообще говоря, в зависимости от типа предприятия. Для каждого из массивов в программе имеются соответствующие блоки контроля, настроенные на структуру, разброс, тип данных и т.д. В ходе работы программа "ФОРМИРОВАНИЕ" последовательно для каждого предприятия считывает из БД1 элементарные массивы, из которых формируется рабочий файл, производит специализированный контроль на достоверность, в заданной последовательности "склеивает" рабочий файл и записывает его в отведенное место в БД2. В случае, если нужный массив в БД1 не найден или в нем обнаружены какие-либо ошибки, выдается соответствующее сообщение. Если же в БД1 была сосредоточена достоверная исчерпывающая информация, то процесс формирования рабочих файлов завершается, и можно приступить к расчетам.

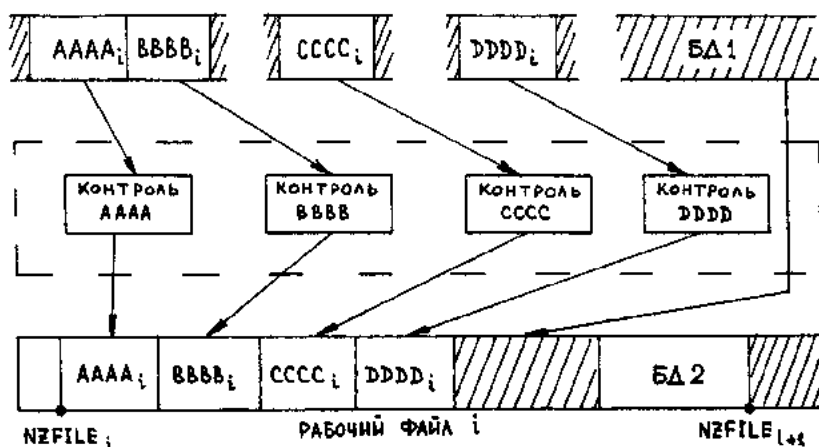


Рис. 3. Схема создания рабочего файла в программе "ФОРМИРОВАНИЕ"

## §5. Адаптация машинной модели в рамках АСУ

Рассмотрим некоторые проблемы информационного обеспечения, возникающие при практическом использовании модели реализации плана в рамках АСУ различного уровня.

Мы описали вариант информационного обеспечения машинной модели, в котором модельная база данных не состыкована программно с реальными источниками информации и заполняется с дисплея или предварительно подготовленными на перфокартах или магнитной ленте массивами. С этим можно как-то мириться при практическом использовании модели на стадии предплановых исследований и в процессе планирования. Решение задач оперативного управления требует постоянного обновления исходной информации в системе, что практически трудно реализуемо в рамках описанного варианта информационного обеспечения. Поэтому эффективное использование машинной модели в реальных АСУ невозможно без предварительной адаптации имеющегося программного обеспечения.

Всякая АСУ содержит ряд подсистем: перспективное планирование, годовое планирование, оперативное управление, учет, финансы и т.д. Данные, используемые разными подсистемами, существенно различаются по степени агрегации, а по экономическому содержанию весьма схожи друг с другом. Это сведения о технологии производства, запасах ресурсов, планах и т.д. Поэтому если мы хотим добиться согласованности перспективных, годовых и оперативных планов, естественно требовать использования единой информационной базы.

Если такая база (общесистемная база данных) имеется в АСУ, то адаптация машинной модели связана с некоторыми программистскими затратами: необходимы дополнительные программы, "перекачивающие" требующуюся информацию из общесистемной БД в модельную БД. Однако опыт, накопленный к настоящему времени в области создания единых баз данных АСУ, явно недостаточен. Пока большинство АСУ представляют собой наборы отдельных задач, решаемых в пакетном режиме и согласованных друг с другом лишь в содержательном и методологическом планах.

В этих условиях встраивание модели процесса выполнения плана в состав АСУ сопровождается анализом имеющихся подсистем с целью выявления возможных источников информации. Практический опыт показывает, что значительная часть необходимой информации содер-

жится в учетных задачах. Этот источник предпочтителен еще и потому, что в большинстве АСУ именно здесь имеется наибольший программный и информационный задел.

Чтобы состыковать источники информации и модельную БД, приходится разрабатывать дополнительные программы, обеспечивающие "перекатку" необходимых массивов. Оставшаяся часть исходной информации модели функционирования загружается в модельную БД вышеописанным способом.

#### ЛИТЕРАТУРА

1. АНЦЫЗ С.М., МАКАРОВ В.Л., МАРШАК В.Д., ФЕДЕЛОВ В.Ф. Математическое обеспечение перспективного отраслевого планирования. - Новосибирск: Наука, 1979.
2. БОБКОВ И.М. Автоматизированные системы управления и их адаптация. - Новосибирск: Наука, 1978.
3. ПЕРМИНОВ С.Б. Имитационное моделирование процессов управления в экономике. - Новосибирск: Наука, 1981.
4. ЛАВЛИНСКИЙ С.М., ПЕРМИНОВ С.Б. Матричная имитационная модель процесса выполнения отраслевого плана. - Оптимизация, 1979, вып. 23(40), с.110-127.
5. ЛАВЛИНСКИЙ С.М., ПЕВНИЦКИЙ А.И., ПЕРМИНОВ С.Б. Система моделей функционирования экономики региона. - Новосибирск, 1982. - 50 с. (Препринт/ИМ СО АН СССР: № 21).
6. ЛАВЛИНСКИЙ С.М., ПЕВНИЦКИЙ А.И., ПЕРМИНОВ С.Б. Система имитационных моделей взаимодействия хозяйственных звеньев в процессе реализации годового народнохозяйственного плана. - Оптимизация, 1981, вып. 27(44), с.96-115.

Поступила в ред.-изд. отдел  
04.10.85 г.