

СООБЩЕНИЯ
ОБЪЕДИНЕННОГО
ИНСТИТУТА
ЯДЕРНЫХ
ИССЛЕДОВАНИЙ
ДУБНА

P10-85-688

А.М.Ершов, Т.А.Ершова, П.П.Сычев,
Г.Н.Тентюкова

СТРУКТУРА БАЗЫ ДАННЫХ
ДЛЯ НЕКОТОРОГО КЛАССА ЗАДАЧ АСУ

1985

При разработке информационного и программного обеспечения автоматизированной системы управления (АСУ) огромную роль играет выбор структуры базы данных. Этот выбор в значительной мере определяет характеристики проектируемой системы: ее эффективность, удобство в использовании, возможность развития и усовершенствования.

База данных как информационная модель объекта управления

Информационная модель экономико-организационного объекта (такого как предприятие или НИИ) практически может быть представлена совокупностью взаимосвязанных показателей^{1/1}. При создании АСУ эта модель должна быть перенесена в ЭЕМ. Совокупность информационных массивов, отражающих информационную модель объекта управления, будем называть базой данных.

Структура базы данных определяет состав и соотношения между информационными массивами, а также средства организации, хранения и доступа к информации. В общем случае база данных включает:

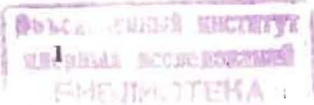
- нормативно-справочные данные, составляющие информационный базис системы;
- текущие сведения о состоянии управляемого объекта (массивы оперативного учета);
- накапливаемые учетные и архивные сведения, отражающие развитие объекта за длительный период;
- оперативную информацию, отражающую текущие изменения объекта управления.

В работе рассматривается класс задач обработки данных в АСУ организационного типа, характеризующийся следующими особенностями:

- относительно велик объем оперативной информации за отчетный период;
- источником сведений об изменении состояния объекта управления является человек, а их носителем - документ. Например, в системах кадрового учета - приказ о приеме на работу, приказ о переводе на должность и т.п., в системе учета материальных ресурсов - платежный документ об оплате полученных материальных ценностей и т.д.

Структура массивов нормативно-справочной информации, массивов оперативного учета и накопительных массивов определяется с учетом конкретной специфики разрабатываемой системы. Рассмотрим подробнее структуру массивов оперативной информации, поступающей из первичных документов, отражающих изменения объекта управления.

Традиционный способ решения задач обработки данных рассматриваемого класса состоит в том, что на основе информации первичных документов строятся массивы изменений состояния объекта управления за



отчетный период. Например, для системы кадрового учета – массив изменения должностей и т.п., для учета материальных ценностей – массив поступления материалов от поставщиков и т.д.

Отметим некоторые недостатки этой схемы. Во-первых, первичные документы часто являются многоцелевыми. Например, платежный документ используется как для учета материальных ценностей, так и для системы финансово-расчетных операций. При этом перечень необходимых реквизитов документа, вообще говоря, не совпадает для различных систем. Это вынуждает либо использовать вторичный документ, выписываемый на основе первичного и содержащий информацию, требуемую только для данной системы, либо многократно вводить первичные документы для их включения в различные массивы изменений. При наличии большого числа массивов изменений система становится громоздкой и неустойчивой при эксплуатации. Во-вторых, часто возникают расхождения между учетными данными ЭВМ и фактическим состоянием объекта. Для определения размеров и причин такого расхождения необходимо установить, все ли требуемые документы полностью обработаны и не были ли какие-либо из них обработаны дважды. В данной схеме сделать это либо сложно, либо вообще невозможно.

Рассмотрим альтернативный вариант, когда в базе данных хранятся все первичные документы за отчетный период с полным набором своих реквизитов. Все изменения в массивах оперативного учета, а также все сводные отчеты и накопительные массивы формируются непосредственно на основе хранящихся первичных документов.

Единый массив первичных документов

Предлагаемая структура базы данных приведена на рис.1. Ее основой является единый массив первичных документов (в дальнейшем – просто массив документов), содержащий все первичные документы за отчетный период. В состав базы данных включен также системный справочник, который наряду с другими требуемыми данными содержит словарь реквизитов и описания первичных документов.

Словарь реквизитов содержит сведения о всех возможных реквизитах первичных документов. Для каждого реквизита указаны его синтаксические параметры: тип – числовой или символьный, максимальная размерность. Кроме того, даны название реквизита, его обозначение, возможность семантической проверки. Словарь реквизитов может рассматриваться и реализовываться как часть словаря данных современных систем управления базами данных (СУБД).

Описание документа определяет структуру первичного документа. Вообще говоря, первичный документ может иметь достаточно сложную

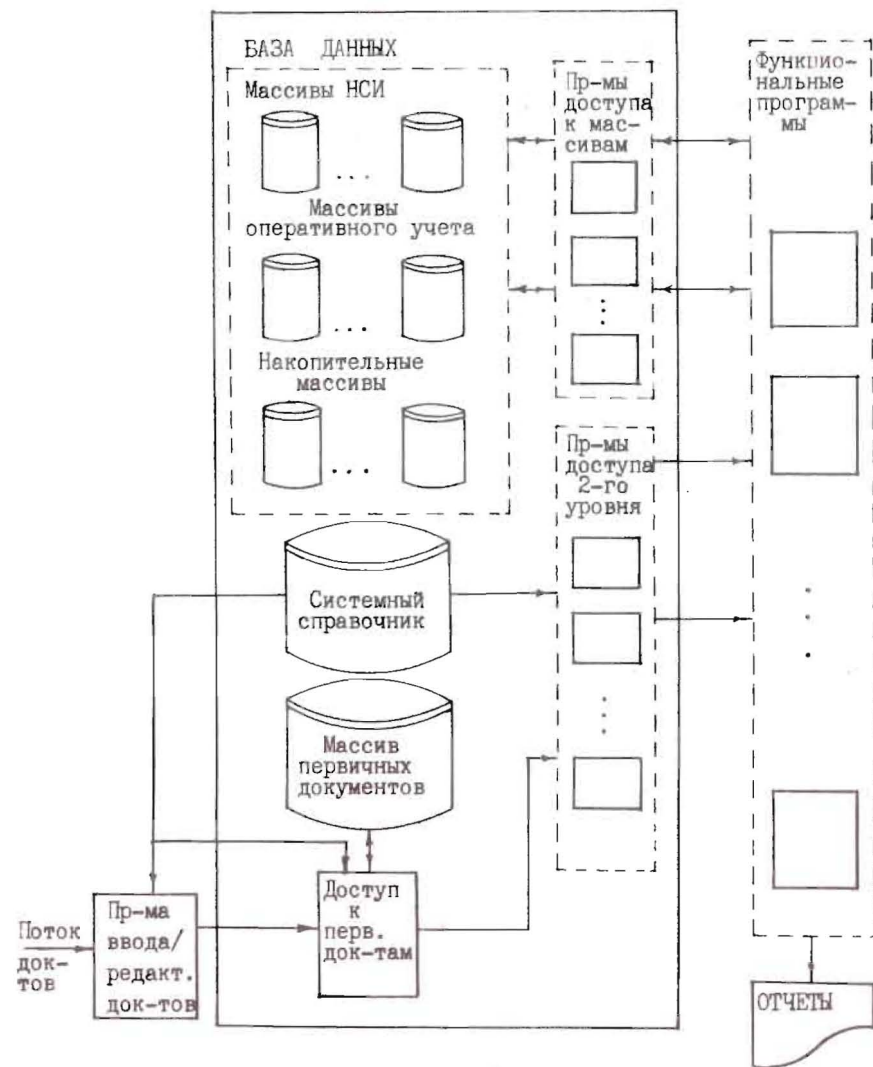


Рис.1. НСИ – нормативно-справочная информация.

структуру. Однако опыт авторов показал, что для большинства практических реализаций приемлемой оказывается простая двухуровневая структура. В этом случае документ содержит заголовочную часть ("шапку") и переменное число строк!

Для каждого семейства однотипных документов строится описание, которое фиксирует:

- код (тип) документа и его название;
- состав и порядок следования реквизитов заголовочной части документа;
- состав и порядок следования реквизитов строки документа.

Использование описаний документов дает возможность ввода всех первичных документов с помощью единой программы ввода/редактирования документов. На основе описаний документов программа производит полный синтаксический контроль вводимых документов, а также обеспечивает возможность редактирования или полного удаления ранее введенных документов^{2/}.

Семантическая обработка документов, т.е. отражение содержащейся в них информации в массивах оперативного учета, формирование накопительных массивов и выдача итоговых отчетов, производится соответствующими функциональными программами. Однако для систем реального времени информация первичного документа должна быть немедленно отражена в массивах оперативного учета. Эти функции могут быть возложены на ту же единую программу ввода/редактирования/обработки документов. При редактировании уже записанного документа в этом случае должны производиться соответствующие изменения в массивах оперативного учета.

Доступ к первичным документам

Рассматриваемая схема предъявляет жесткие требования к ее реализации для обеспечения эффективного доступа к большому числу разнотипных документов. Большинство функциональных программ обрабатывают первичные документы последовательно, но для ряда задач, в первую очередь для редактирования, необходим прямой доступ к конкретному документу.

Полная запись документа содержит (см. рис.2):

- фиксированную часть, содержащую тип документа, ключевые поля и, возможно, служебные реквизиты, такие как число строк в документе, флаг наличия синтаксических ошибок в документе, флаги семантической обработки и т.п.;
 - переменную часть, содержащую "шапку" и строки документа.
- Разбиение этой части на отдельные реквизиты производится с помощью описания данного типа документа.

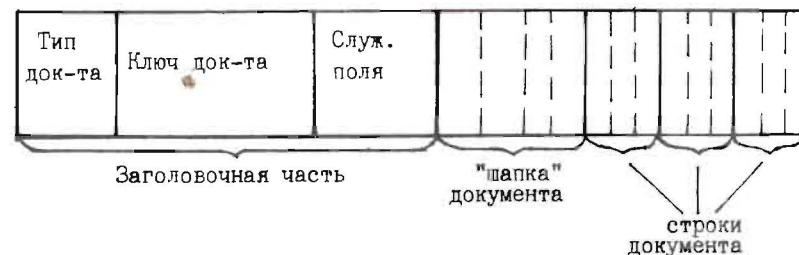


Рис. 2

Ключом документа является совокупность значений конкретных ключевых реквизитов. Номера ключевых реквизитов вводятся как часть описания документа. В качестве них могут быть использованы код, место выписки документа, его номер и т.п.

Полный доступ к документам обеспечивается специальной программой доступа. При этом возможны следующие операции:

- чтение по полному ключу, т.е. типу документа и значению его ключевых реквизитов;
- последовательное чтение;
- запись, перезапись или удаление конкретного документа (производится по полному ключу).

Возможные реализации схемы данных массива документов в СУБД ОКА^{4/} приведены на рис.3. В первом случае корневой сегмент(КЕYDOC)

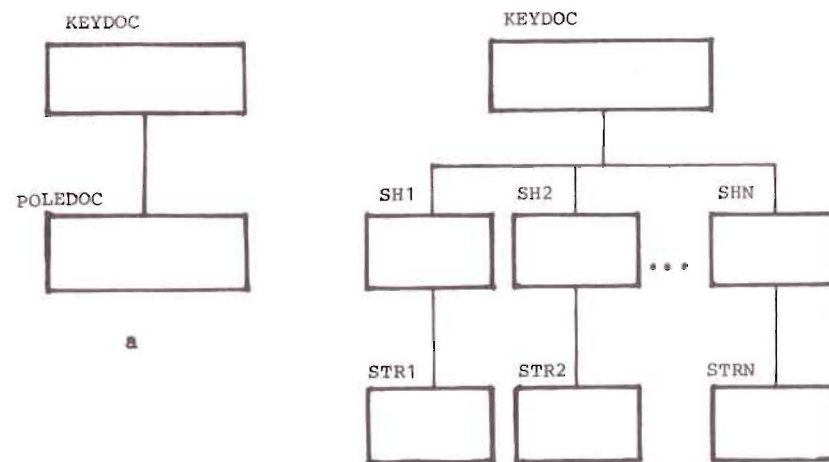


Рис.3

б

содержит фиксированную часть записи, переменная часть состоит из необходимого числа реализаций подчиненного сегмента (PОВЕДОС). Таким образом, реализуется запись переменной длины. Во втором случае для каждого типа документа включается свой сегмент 2-го уровня (SH1, SH2, ...), содержащий "шапку" документа и сегмент 3-го уровня (STR1, STR2, ...), соответствующий строке документа. Описание документа в этом случае является частью описания данных (DBD) в СУБД ОКА.

Каждая из приведенных схем имеет свои достоинства и недостатки. На практике авторы использовали первый вариант (рис.3а).

Использование рассмотренного способа доступа к массиву документов не может быть рекомендовано для функциональных программ, т.к. работа требует постоянной динамической привязки к описанию документов. Однако это основной способ для программ, обеспечивающих ввод, редактирование, обработку и распечатку всех первичных документов.

Программы доступа второго уровня

Пусть $A = (\alpha_1, \dots, \alpha_m)$ - полный словарь возможных реквизитов первичных документов. Тогда каждому типу документов соответствует отношение

$$R = R(\beta_1, \dots, \beta_n), \quad \beta_i \in A,$$

где атрибут β_i - i -й реквизит документа. Кортежу этого отношения соответствует "документострока", т.е. полный набор реквизитов документа, причем значения реквизитов "шапки" документа повторяются для каждой новой строки документа. Таким образом, с точки зрения отношения R неважно, принадлежит ли конкретный реквизит β "шапке" или строке документа.

Полная реляционная схема массива первичных документов задается следующим образом:

$$R_i = R_i(\alpha_{i1}, \dots, \alpha_{in_i}), \quad i=1, \dots, N, \quad \alpha_{ij} \in A,$$

для массива, содержащего N различных типов документов.

Информационные требования функциональных программ к массиву первичных документов задаются в виде набора абстрактных файлов (АФ) второго уровня^{3/}. Абстрактным файлом второго уровня назовем отношение $\hat{R} = \hat{R}(\beta_1, \dots, \beta_m)$, где каждое β_i определяется следующим выражением:

$$\beta_i = \beta_i^1 \vee \beta_i^2 \vee \dots \vee \beta_i^{i'}, \quad i' \geq 1, \quad \beta_i^j \in A \quad \text{для } 1 \leq i < i^N,$$

$$\beta_i^{i'} \in A \cup \{\omega\}.$$

Смысл указанной формулы таков, что в качестве i -го атрибута отношения \hat{R} должен быть выбран один из перечисленных реквизитов в порядке предпочтения. Выражение может заканчиваться специальным символом ω , что соответствует "пустому", т.е. необязательному для данного АФ реквизиту.

Для каждого АФ \hat{R} может задаваться также множество типов документов $\hat{I} = \{i\}$, $i \in N$, для которых он определен. Если \hat{I} не задано, оно считается совпадающим с полной схемой: $\hat{I} = \{1, 2, \dots, N\}$.

Документ i -го типа удовлетворяет требованиям АФ \hat{R} , если:

- а) $i \in \hat{I}$ и
 б) для $\forall j \in \{1, \dots, m\}$ либо существует $\alpha_{ik} \in \{\alpha_{i1}, \dots, \alpha_{in_i}\}$, входящий в список $\beta = \beta_j^1 \vee \dots \vee \beta_j^{j'}$, либо этот список заканчивается специальным символом ω .

Множество типов документов, удовлетворяющих требованиям \hat{R} , обозначим через $\hat{I}_1 \subset \hat{I}$. Отношение \hat{R} строится следующим образом:

$$\hat{R} = \bigcup_{i \in \hat{I}_1} \hat{R}_i,$$

где \hat{R}_i - соответствующая проекция первичных документов i -го типа R_i . Для отсутствующих в документе необязательных реквизитов АФ соответствующим атрибутам присваиваются "пустые" значения, т.е. нулевые для числовых и пробелы для символьных реквизитов.

Каждый АФ реализуется специальной программой доступа. Типовое обращение к ней выглядит следующим образом (в синтаксисе Кобола):

```
CALL 'NNNN' USING REC-NNNN, FLAG,
```

- где NNNN - имя АФ второго уровня;
 REC-NNNN - запись фиксированной структуры, куда передается сформированная очередная документострока (кортеж отношения NNNN);
 FLAG - код возврата, нулевое значение которого означает успешное завершение операции; значение, отличное от нуля, сигнализирует либо об окончании требуемых данных в массиве документов, либо о возникновении ошибочных ситуаций.

Программа доступа второго уровня получает информацию из первичных документов при помощи ранее рассмотренного полного доступа к массиву документов.

Практическая реализация

Указанная структура базы данных использована авторами при разработке программного обеспечения ряда подсистем АСУ ОИЯИ: подсисте-

мы учета материальных ценностей (УМЦ ОИЯИ), подсистемы расчета потребностей Опытного производства ОИЯИ для выпуска электронных блоков (РП ОП ОИЯИ), подсистемы учета основных фондов (УОФ ОИЯИ).

Практическая реализация показала ряд важных достоинств:

- возможность в процессе эксплуатации систем добавить новые типы документов или изменить описание существующих типов документов. Все функциональные программы при этом сохраняют полную работоспособность и не требуют изменений или перетрансляции. Если необходимо, они автоматически начинают обрабатывать информацию новых типов первичных документов;

- возможность в случае необходимости распечатать "сомнительный" документ именно в том виде, в каком он существует в массиве документов. Так как описание документа отражает исходную ("бумажную") структуру первичного документа, такая распечатка не требует каких-либо пояснений для пользователя системы.

Описанная модель позволила автоматизировать процесс написания программ доступа второго уровня. Программа-генератор на основе описания требуемого АФ формирует текст соответствующей программы доступа на языке программирования Кобол.

Функциональные программы, так же как и программы доступа второго уровня всех указанных систем, являются СУБД-независимыми. Все взаимодействия с базой данных они производят через программные интерфейсы и не содержат явно каких-либо операторов языка манипулирования данными конкретной СУБД.

При реализации подсистемы учета основных фондов за основу был взят готовый пакет прикладных программ. Пакет обеспечивал решение всех функциональных задач, информация первичных документов для него должна была готовиться отдельно в виде набора файлов (последовательных наборов данных ОС ЕС). Использование рассмотренной методики в этом случае позволило обеспечить единый ввод всех требуемых первичных документов. С помощью соответствующих программ доступа второго уровня появилась возможность генерировать требуемые входные файлы для функциональных программ пакета.

ЛИТЕРАТУРА

1. Сняк В.С. и др. Автоматизированные системы управления и руководитель. "Финансы и статистика", М., 1983.
2. Ершов А.М. и др. ОИЯИ, Р10-85-686, Дубна, 1985.
3. Говорун Н.Н. и др. ОИЯИ, IO-85-229, Дубна, 1985.
4. Андон Ф.И. и др. Управляющие системы и машины, 1977, №2, с.32.

Рукопись поступила в издательский отдел
20 сентября 1985 года.

Ершов А.М. и др.

P10-85-688

Структура базы данных для некоторого класса задач АСУ

Описана структура базы данных и дано обоснование ее выбора для некоторого класса задач обработки данных в автоматизированных системах управления /АСУ/ организационного типа, характеризующегося большим потоком первичной информации. Особенности данной схемы являются единый массив первичных документов и возможность доступа к различным подмножествам документов с помощью метода виртуальных файлов. Приведены примеры использования описанной структуры базы данных.

Работа выполнена в Лаборатории вычислительной техники и автоматизации ОИЯИ.

Сообщение Объединенного института ядерных исследований. Дубна 1985

Ershov A.M. et al.

P10-85-688

The Data Base Structure for Some Class MIS Problems

The data base structure is described and its choice substantiation for some class data processing problems in the organizing - type management information systems (MIS) with a large stream of initial information is given. The united array of initial documents with access possibility to different subsets of documents by virtual file method is the special feature of the given scheme. Examples of the described data base structure application are given.

The investigation has been performed at the Laboratory of Computing Techniques and Automation, JINR.

Communication of the Joint Institute for Nuclear Research, Dubna 1985