

MTA Számítástechnikai és Automatizálási Kutató Intézet Budapest



MAGYAR TUDOMÁNYOS AKADÉMIA
SZÁMITÁSTECHNIKAI ÉS AUTOMATIZÁLÁSI KUTATO INTÉZETE

PT-II / КНВВТ
"СИСТЕМЫ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ И ИНФОРМАЦИОННЫЕ
СИСТЕМЫ"

СБОРНИК
НАУЧНО-ИССЛЕДОВАТЕЛЬСКИХ РАБОТ

ТОМ II

A kiadásért felelős:

DR VÁMOS TIBOR

ISBN 963 311 129 3

ISSN 0324-2951

С О Д Е Р Ж А Н И Е
Сборник научно-исследовательских работ
рабочей группы РГ-ИИ, КНВВТ

выпуск I3I - том I

Предисловие	7
Александров, А.П. Проблемы создания вычислительных центров коллективного пользования в Комитете по Единой системе социальной информации	II
Барнев, П., Кр.Марков Проблемы управления в системах информацион- ного обслуживания коллективов	I5
Батурина, Л.Н., Н.А.Лепешинский Имитация работы сетей ЭВМ	25
Бельке, В., Х.-Д. Хартманн, Б.Лойхт Автоматизированная система классификации систем управления базами данных (СУБД)	3I
Боянов, К.Л., В.С.Гетов, Х.А.Турлаков Высокопроизводительные параллельные процес- соры с сетевым программированием	39
Денев, Й.Д., Е.К.Живкова, Р.П.Лесева Средства форматирования	53
Добрев, Д.М., Р.К.Киркова, П.А.Парванов Система планирования и учета вычислительных ресурсов	57
Добрев, Д.М., Й.В.Швертнер Доступ к записям в СУБД БИСЕС	63
Жечковски, В. Язык манипулирования данными системы управ- ления базой данных LINDA	67
Занев, В. Концепции и модель распределенной системы информационного обслуживания коллективов	79

Златарова, Ф.	Эквивалентность моделей данных в распределенных базах данных	89
Киркова, Р.К.	Применение информационных систем при организации научных мероприятий	99
Киркова, Р.К.	СОКРАТ - Система для оперативного контроля над реализацией и отчетом заграничных служебных командировок	I05
Кондратьев, А.И.	Подход к построению математической теории для классов информационных систем	III
	Список мероприятий РГ-II "Системы управления базами данных и информационные системы"	I23
	Имена и адреса участников РГ-II	I25
выпуск I32 - том II		
Кузнецов, Е.П.	Некоторые вопросы обработки информации в условиях ВЦКП СО АН СССР	7
Ласкин, Л.Ф., В.Н.Безряков	Об использовании мини-ЭВМ в системе ВЦКП	2I
Либл, П.	Базы данных и их назначение в процессе сбора и обработки информации	3I
Марчук, Г.И., О.В.Москалев	Проблемы и эволюция вычислительных центров коллективного пользования	39
Метляев, Ю.В.	Технические средства ВЦКП СО АН СССР	49

Радулов, И.Р., И.Л.Владиков, М.Т.Калчева	
Об одном подходе при включении цифрового терминала ИЗОТ 8500 в системы коллективного пользования	59
Савинков, В.М., О.М.Вейнеров, М.С.Казаров, А.А.Александров	
Обобщенные процедуры логического проектирования баз данных и уточнения инфологической модели: формальный подход	67
Стогний, А.А., А.И.Кондратьев	
О построении математического аппарата для описания процессов проектирования и функционирования информационных систем	83
Терзиев, А.И.	
Управление формата отчетов и поиска информации в базах данных вычислительных систем с множественным доступом	97
Швертнер, И.В.	
Защита корректности данных при вводе в СУБД БИСЕС	105
Швертнер, И.В., Л.Манасиев	
Системный журнал в СУБД БИСЕС	III
Эскенази, А.М., Н.М.Манева, В.Т.Петрова	
Поиск при помощи инвертированных файлов в системе БИСЕС	II7

выпуск I33 - том III

Barnev, P., At.Radensky, P.Azalov, Kr.Markov, Z.Vassilev	
A local information station - Version one	7
Benczur, A.	
Problems in modelling of data base performance	27
Bittner, J.	
DBS/R - A system of practice	43

Demetrovics, J., Gy. Gyepesi	
Logical dependencies in Relational Data Base	59
Kerékfy, P.	
Some remarks on statistical data processing	79
Havel, I., P. Liebl	
A relational DBMS in Concurrent PASCAL	99
Riha, A.	
Modifiable query system for casual Data base user	III
Werner, W., D. Koch	
Natural language interfaces to Data bases: a day-dream or a realistic goal?	I23

НЕКОТОРЫЕ ВОПРОСЫ ОБРАБОТКИ ИНФОРМАЦИИ В УСЛОВИЯХ ВЦКП СО АН СССР

Е.П.Кузнецов

ВЦ СО АН СССР, Новосибирск, СССР

АННОТАЦИЯ. В работе предпринята попытка путем анализа операционной обстановки [8], порождаемой комплексом технических средств и архитектурных решений, сформулировать некоторые требования и вытекающие из них условия к обработке информации в условиях ВЦКП СО АН СССР.

Наряду с созданием программных методов и технических средств для ассоциативного поиска информации, методов распараллеливания операций, специализированных процессоров для манипулирования с базами данных, в оптимизации процессов обработки информации право на жизнь приобретают и усовершенствования в архитектуре больших банков данных, к чему, в первую очередь, относится стратегия распределенной обработки информации [9, I3-I7].

Переход к распределенным БД существенно изменяет концепции построения программного обеспечения баз данных. Это обуславливается в первую очередь тем, что такой БД состоит из ряда практически независимых баз, управляемых собственными системами управления.

Считается, что сложность распределенного БД приблизительно на порядок больше сложности существующих централизованных банков.

Наиболее существенный фактор сложности системы в целом зависит от того, является ли управление распределенным или централизованным, т.к. в первом случае имеет место динамическое соподчинение процессоров, так что связи между ними организуются весьма гибко без наличия центрального коммутирующего блока.

Во втором случае применяется иерархическая структура со статическим подчинением всех процессоров одному центральному процессору.

Рассмотрим операционную обстановку, порождаемую комплексом архитектурных решений, средств вычислительной техники, способами организации функциональных компонент и общесистемного программного обеспечения ВЦКП СО АН СССР^{*}) и сформулируем требования, предъявляемые к методам и способам обработки информации в этой операционной обстановке.

Вычислительный центр коллективного пользования, создаваемый в Сибирском отделении АН СССР, является многомашинным территориально-распределенным вычислительным комплексом общего назначения типа сети ЭВМ, базирующемся на разнородных средствах отечественной вычислительной техники и представляющем для пользователей единую вычислительную систему, обеспечивающую возможность использования комплекса технических средств и информационно-вычислительных ресурсов в режимах разделения времени при контакте пользователя с системой и пакетной обработки при решении основных задач.

Архитектура ВЦКП СО АН СССР (рис. I) отражает многомашинную конфигурацию, территориальное распределение вычислительных средств, разнотипность входящего в состав комплекса оборудования и развитую сеть терминального оборудования и телекоммуникаций.

Базовые вычислительные комплексы (БВК) включают высокопроизводительные модели ЭВМ и сосредотачивают основные вычислительные и информационные мощности ВЦКП.

Функционально БВК обеспечивают вычисления и обработку информационных данных в режиме дистанционной пакетной обработки. В составе ВЦКП БВК достаточно автономен, т.к. обладает собственной развитой сетью терминалов, обеспечивающих контакт абонентов с ЭВМ БВК не только через ЦЦО, но и через собственный ЦЦД, что представляет возможность организации коллективного доступа к ЭВМ БВК, предусмотренного и обеспечиваемого операционными системами ЭВМ БВК.

^{*}) Для целей настоящей работы желательно предварительное ознакомление с концепцией построения, архитектурой и организацией функциональных компонент ВЦКП СО АН СССР. Исчерпывающие описания и обоснования принятых решений даны в работах [1-4]. Здесь же повторены некоторые положения, необходимые для дальнейших рассуждений.

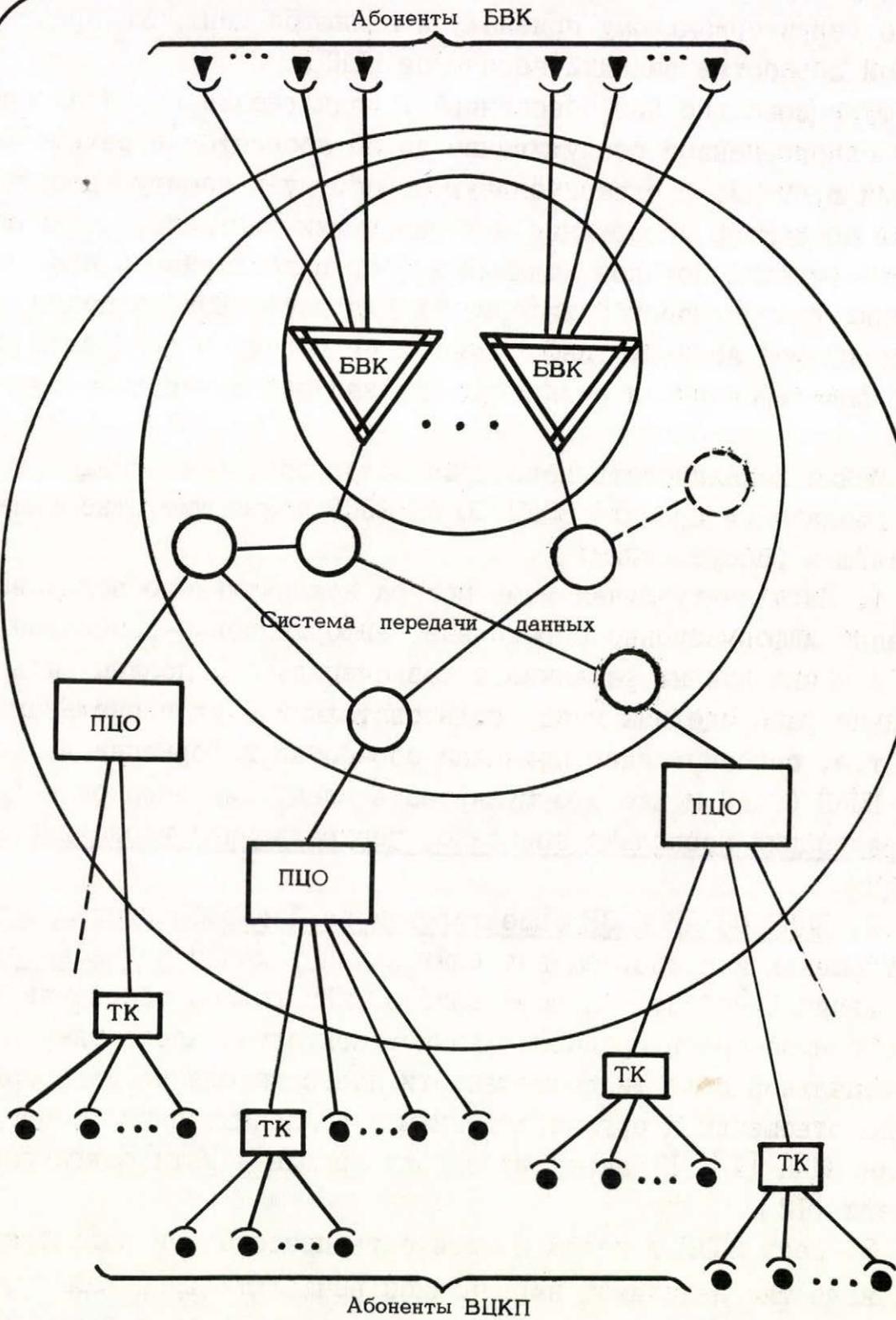


Рис. 1. Функциональная схема ВЦКП СОАН СССР

Периферийные центры обработки (ПЦО) концентрируют абонентов по территориальному признаку и являются центрами предварительной обработки заданий абонентов ВЦКП.

Функционально ПЦО обеспечивает подключение до 64 абонентов и одновременное обслуживание до 50 абонентов в режиме разделения времени; предварительную обработку и защиту поступающей от абонентов информации и управляющих директив; первичную диспетчеризацию потоков заданий и сбор находящейся в его компетенции статистической информации для всего ВЦКП в целом; прием адресуемой абоненту информации и ее выдачу в требуемом формате; пакетирование и распределение заданий для обработки на БВК.

Можно локализовать некоторые положения, положенные в основу реализации проекта ВЦКП СО АН СССР и имеющие отношения к дальнейшим рассуждениям:

1. Интеллектуальная мощь центра коллективного пользования, а именно информационно-справочные, информационно-поисковые системы и банки данных различного назначения, АСУ должны быть оптимально распределены между пользователями - их владельцами и БВК, т.е. основную идею принципа обработки информации в условиях ВЦКП СО АН можно сформулировать следующим образом - "децентрализация настолько возможно, централизация настолько необходимо".

2. ВЦКП СО АН СССР ориентирован на "непрофессионального" по отношению к средствам вычислительной техники пользователя, т.е. взаимодействие пользователей с ВЦКП должно обеспечиваться проблемно-ориентированным языком контакта, адекватным профессиональной области деятельности пользователя и инвариантным по отношению к организационным и реализационным особенностям ВЦКП [1]. К такому же выводу пришли и наши болгарские коллеги [12].

3. Весь ВЦКП в целом должен быть представлен пользователю в качестве неделимой виртуальной вычислительной машины, т.е. формулирование вычислительной системы заданий на работу должно осуществляться пользователями преимущественно в режиме диалога при работе терминального абонента ПЦО в режиме разделения времени, а основным режимом исполнения заданий пользова-

телей является режим целостного исполнения заданий на БВК, т.е. режим пакетной обработки[2].

4. ВЦКП СО АН СССР представляет собой комплекс средств вычислительной техники и общесистемного программного обеспечения, являющихся технической и технологической базой для развертывания таких основных прикладных программных систем, как системы обеспечения научных исследований (кроме управления экспериментом в реальном времени), информационно-справочные и информационно-поисковые системы общего и специального назначения, автоматизированные системы управления предприятиями, территориально-промышленными комплексами, регионами и т.п. [1].

5. Конечной целью разработки и внедрения любого проекта является эксплуатация. Являясь последним звеном в реализации проекта, эксплуатация оказывает значительное влияние на конечный результат - эффективность функционирования ВЦКП, т.е. эффективность комплексного взаимодействия технических, программных, информационных средств, обслуживающего персонала и абонентов ВЦКП.

Повышение культуры и уровня обслуживания пользователей в условиях распределенной обработки информации в большей мере зависит от управления процесса выполнения заданий, от технологичности этого процесса.

Для обеспечения коллективного, территориально-распределенного характера использования средств вычислительной техники ВЦКП СО АН СССР выбрана технология распределенного управления.

Этим в общих чертах определяется глобальная операционная обстановка, в которую погружена любая проблемная система ВЦКП. Но каждая проблемная система порождает свою локальную проблемную операционную обстановку.

Какие же требования можно выдвинуть по отношению к проблемной системе для того, чтобы порождаемая ею локальная операционная обстановка и ее операционное окружение не вступали в противоречие.

Специализация в распределении сфер влияния в науке и управлении оказывает существенное влияние на специализацию интеллектуальной мощи ВЦКП, именно — банков данных и эти банки данных должны создаваться высококвалифицированными специалистами, которые будут поддерживать их на самом современном уровне, проводить их авторское сопровождение, регламентацию доступа и будут заинтересованы в их развитии и использовании.

В условиях Сибирского отделения такие проблемно-ориентированные и специализированные банки данных целесообразно размещать на ведомственных вычислительных комплексах (ВВК) и только при наличии объективных предпосылок — на БВК ВЦКП. Возможность доступа к таким банкам данных обеспечивается средствами ВЦКП.

Другой вид информации — классификаторы, каталоги, справочные системы, автоматизированные системы управления регионом и т.п., т.е. общезначимую информацию, ориентированную на широкий круг пользователей целесообразно размещать на БВК ВЦКП.

Территориальное распределение функциональных компонент комплекса, распределенное управление обработкой информации и принцип представления комплекса в качестве виртуальной вычислительной машины диктуют необходимость распределения компонент проблемных систем по функциональным компонентам комплекса.

Если первичную (мажорную) информацию, представляющую суть проблемной системы, целесообразно размещать на ВВК или БВК, то для обеспечения связи с этой системой необходимую служебную (минорную) информацию необходимо размещать на ЦСО.

Причем эта минорная информация является чисто служебной с точки зрения ВЦКП.

Однако в проблемных системах такого класса, как ИПС, АСУ целесообразно провести дальнейшую классификацию информации, отражающую иерархию управления [II].

Дело в том, что уже начиная со второго уровня управления довольно часто требуется не детальная информация о каждом объекте проблемной системы, а некоторая вторичная (агрегированная) о совокупности объектов, причем такая информация порождаемая нижележащим уровнем более необходима вышележащему. Создание и

накопление такой агрегированной информации может быть и не предусмотрено функциями проблемной системы или создаваться на основе нескольких проблемных систем. В этом случае возникает некий тандем владельцев такой подсистемы, каждый из которых со своего уровня видит ее по-своему. Необходимым условием функционирования такой подсистемы является согласованность и двусторонняя заинтересованность владельцев. Рычаги, влияющие на заинтересованность, чаще всего находятся в области правовых отношений.

Агрегированную информацию такого рода можно разделить еще на два класса:

- * статические мажоранты - представляющие собой объекты верхнего уровня подсистемы, имеющие относительно статическую или регламентируемую по сроку достоверность (например, фонд заработной платы института, директивные плановые показатели и т.п.). Такие объекты создаются или перемещаются с нижнего уровня подсистемы на верхний и отслеживаются в соответствии с регламентом на достоверность.
- * динамические мажоранты - представляющие собой информацию нижнего уровня, возникающую динамически и, которую необходимо и агрегировать динамически (например, реальное выполнение плановых показателей, поставка материалов, численность работающих и т.п.). Стратегия размещения и создания такой информации выбирается в каждом случае отдельно.

В условиях коллективного доступа большую значимость приобретают вопросы обеспечения защиты и регламентации использования информации. Обеспечение желаемой глубины защиты прямо пропорционально системным затратам.

В условиях ВЦКП, когда проблемные системы в прямом и переносном смысле распределены по функциональным компонентам, эта проблема решается также путем распределения функций защиты по компонентам комплекса, а именно: на уровне пользователь - ИЦО защищаются проблемные программы от пользователей, на уровне ИЦО - проблемные системы защищаются информация от проблемных программ [7].

Таким образом, информационный срез ВЦКП по одной проблемной системе можно представить следующим образом (рис. 2).

Технология прохождения заданий пользователей ВЦКП СО АН СССР представлена на рис. 3.

На двух произвольно взятых ЦО одновременно работают абоненты А, В, ..., I, которые формулируют задания к проблемным системам Π^1 и Π^2 расположенным на одном БВК, Π^3 и справочной системе C^1 расположенным на другом БВК и проблемной системе Π^4 расположенной на БВК, причем к системе Π^1 обращаются абоненты А и F; к системе Π^2 - абоненты В и I, к системе Π^3 - абоненты Е и D, к системе Π^4 - абоненты С и H, к системе C^1 - абонент G.

Формулирование задания в режиме диалога с ЦО не предполагает знание пользователем местоположения необходимой ему проблемной системы. На этом этапе задание подвергается синтаксическому контролю и предварительной семантической защите [10].

Поступившие на ЦО задания пакетируются и распределяются для передачи на соответствующие БВК или ЦО. Серией модельных экспериментов [5,6] обоснованы оптимальные параметры длины сообщений, коэффициентов пакетирования, пропускной способности каналов передачи данных.

По системе передачи данных пакеты поступают в соответствующие связные процессоры (СВП) БВК или ЦО, где они перепакетируются по соответствующим проблемным системам и поступают на выполнение в ЭВМ. На ЭВМ БВК при просмотре внешней памяти выполняются все задания пакета.

Резюмируя сказанное, можно сформулировать следующие требования к проблемным системам, функционирующим в условиях ВЦКП СО АН СССР:

1. Функциональные компоненты проблемных систем должны быть распределены по функциональным компонентам ВЦКП;
2. Архитектура проблемных систем должна стремиться отражать иерархию той проблемной области, которую она отражает;
3. При создании проблемных систем в близких проблемных областях целесообразно стремиться к концентрации пересекающейся информации в рамках одной системы, а не к ее дублированию.

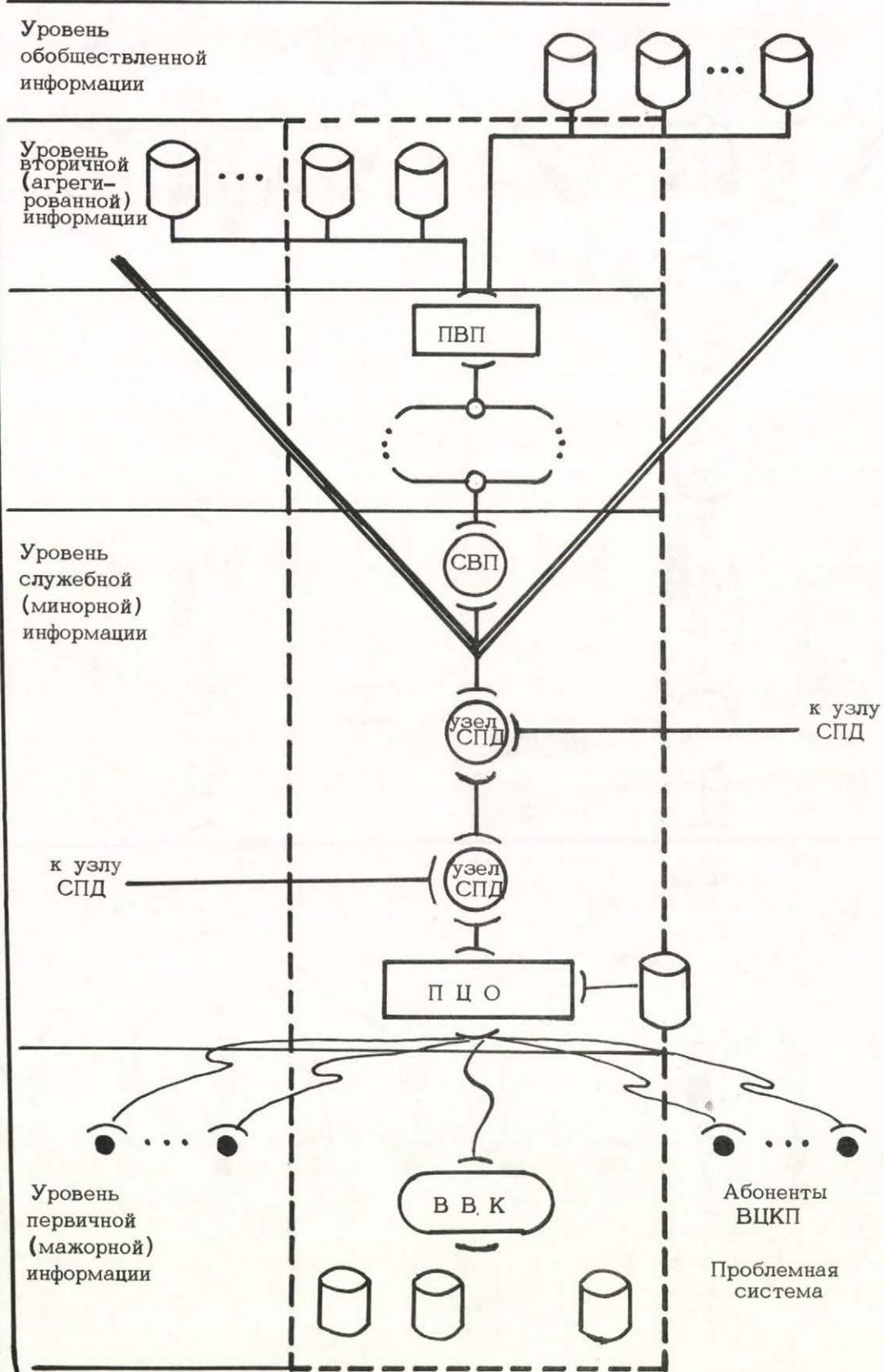


Рис.2. Информационный срез по проблемной системе

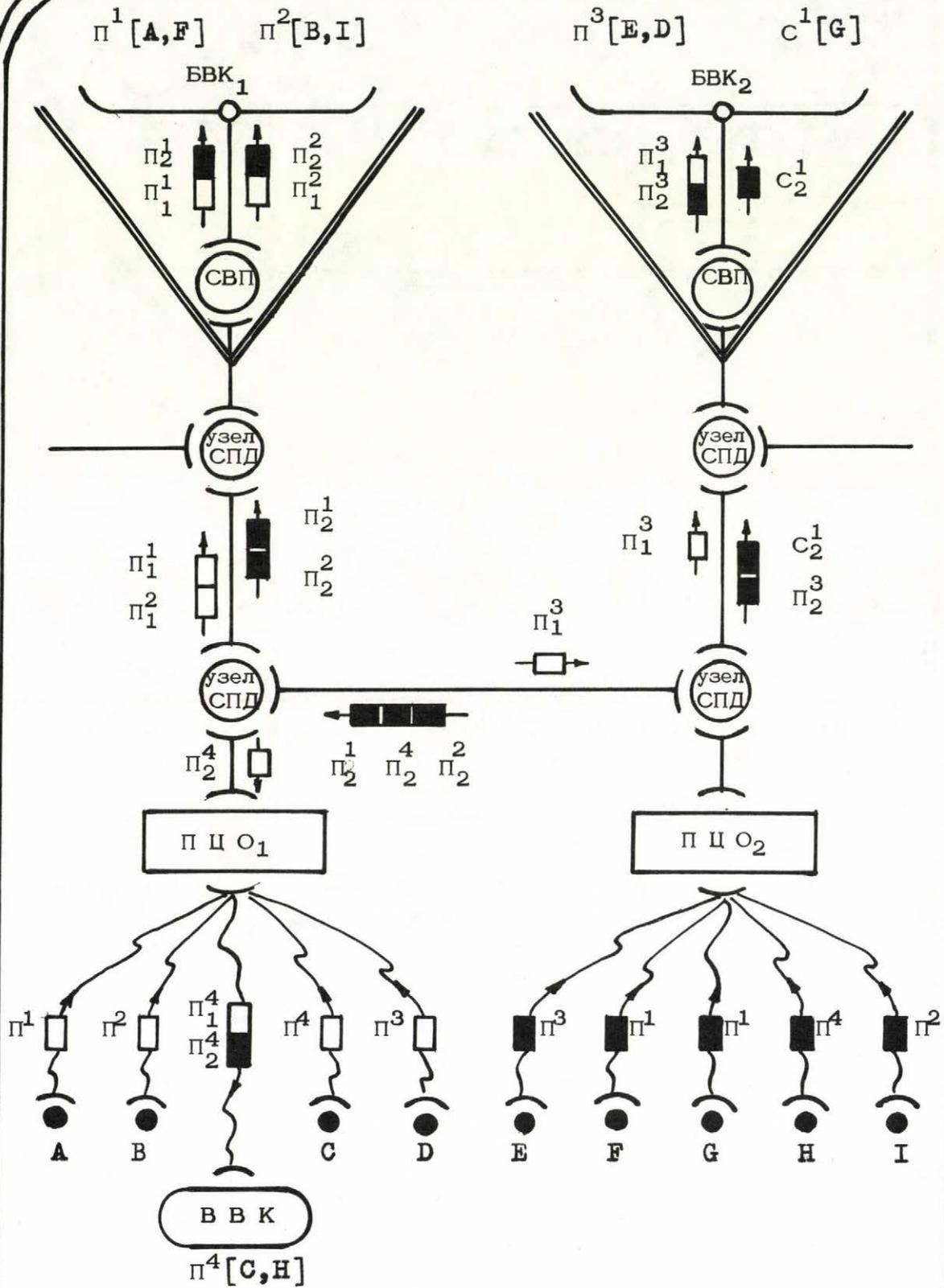


Рис.3. Схема прохождения заданий на ВЦКП СОАН СССР

4. Для обеспечения требуемой сложности проблемной системы и образования необходимой минорной информации, создание проблемных систем должно проводиться в тесном сотрудничестве с системными программистами общесистемного программного обеспечения (ОСПО) ВЦКП;

5. Вопросы образования и размещения мажорант, связанные со статическим или динамическим агрегированием информации об объектах проблемной системы или систем, должны находиться в компетенции наивысшего в иерархии пользователей уровня, заинтересованного в такой информации.

В работе сознательно не затронуты вопросы, касающиеся языков общения пользователей с ВЦКП, языков создания проблемных систем, стратегии их согласованности и некоторые другие. В той или иной мере эти проблемы рассмотрены в публикациях по ВЦКП СО АН СССР.

В заключении автор считает своим долгом выразить искреннюю признательность коллегам по работе и, в первую очередь, О.В.Москалеву, Л.Б.Эфросу, Ю.В.Метляеву за доброжелательные, плодотворные контакты, в процессе которых сложились основные положения данной работы.

Л и т е р а т у р а

1. МАРЧУК Г.И., КУЗНЕЦОВ Е.П., МОСКАЛЕВ О.В., МЕТЛЯЕВ Ю.В., ЭФРОС Л.Б. О программе работ по созданию вычислительного комплекса (центра) коллективного пользования в Новосибирском научном центре СО АН СССР (проект "ВЦКП") - В сб.: "Программное и техническое обеспечение ВЦКП", Новосибирск, ВЦ СО АН СССР, 1978, стр. 7-39.
2. МЕТЛЯЕВ Ю.В., МОСКАЛЕВ О.В., ЭФРОС Л.Б. Архитектура вычислительного комплекса (центра) коллективного пользования СО АН СССР. - В сб.: "Вычислительная техника", Новосибирск, ВЦ СО АН СССР, 1976, стр. 6-14.

3. МЕТЛЯЕВ Ю.В. Базовый вычислительный комплекс на основе ЭВМ ЕС. - В сб.: "Обеспечение ВЦКП", Новосибирск, ВЦ СО АН СССР, 1977, стр. 88-125.
4. МЕТЛЯЕВ Ю.В. Периферийный центр обработки на базе мини-ЭВМ (М-7000, СМ-2) и системы КАМАК. - В сб.: "Программное и техническое обеспечение ВЦКП", Новосибирск, ВЦ СО АН СССР, 1978, стр. 169-185.
5. КУРБАНГУЛОВ В.Х., МИТРОФАНОВ Ю.И., МОСКАЛЕВ О.В. Результаты аналитического моделирования ВЦКП СО АН СССР первой очереди. - В сб.: "Обеспечение ВЦКП", Новосибирск, ВЦ СО АН СССР, 1977, стр. 8-21.
6. МИТРОФАНОВ Ю.И., КУРБАНГУЛОВ В.Х., ШУЛЬГА Н.П. Результаты моделирования ВЦКП СО АН СССР показательными сетями обслуживания. - В сб.: "Программное и техническое обеспечение ВЦКП", Новосибирск, ВЦ СО АН СССР, 1978, стр. 40-109.
7. ПЕТУХОВ А.Д., ПЕТУХОВ В.Д. Один из подходов организации работы с программами и данными в ВЦКП. - В сб.: "Обеспечение ВЦКП", Новосибирск, ВЦ СО АН СССР, 1977, стр. 41-48.
8. ЭФРОС Л.Б. Концептуальный анализ программных систем. - "Управляющие системы и машины", 1979, № 2, стр. 25-32.
9. КУЗНЕЦОВ Е.П. Некоторые вопросы сравнительного анализа и создания баз данных. - В сб.: "Обеспечение ВЦКП", Новосибирск, ВЦ СО АН СССР, 1977, стр. 71-80.
10. КУЗНЕЦОВ Е.П., ПАВЛОВСКАЯ И.Ю. Система управления данными в ЭТА-системе программирования. - В сб.: "Обеспечение ВЦКП", Новосибирск, ВЦ СО АН СССР, 1977, стр. 33-40.
11. КАРПАЧЕВ Г.И., КУЗНЕЦОВ Е.П. О создании автоматизированной системы обработки данных для целей управления районом. - В сб.: "Вычислительная техника", Новосибирск, ВЦ СО АН СССР, 1976, стр. 86-91.
12. BARNEV P. Systems for information servicing of collectivities. *Serdica*, 4, 1978, 164-179.
13. DAVENPORT R.A. Distributed or centralised data base. "Comput. J.", 1978, N1, 7-14

14. FOSTER JOHN D. The development of a concept for distributive processing. "IEEE 1976 COMPCOM, Spring", 1976, 28-30
15. KOLLECK BERND, SKAISKI DETLEF. "3 Internationaler Kongres fur Datenverarbeitung IKD, Berlin, 1978(12-15.09)" und "4th International Conference on Very Large Databases VLDB., Berlin, 1978(13-15.09)" "ABT-Inform.", 1978, N27, 23-41.
16. SHERWOOD H.F. IBM'S strategy in terminals and distributed processing. "Datamation", 1978, 24, N3, 92-98.
17. SCHERR A.L. Distributed data processing. "IBM Syst. J." 1978, 17, N4, 324-344.

ОБ ИСПОЛЬЗОВАНИИ МИНИ-ЭВМ В СИСТЕМЕ ВЦКП

Л.Ф.Ласкин, В.Н.Безряков
ВЦ СО АН СССР, Новосибирск, СССР

АННОТАЦИЯ. В докладе описывается характер работ по созданию математического обеспечения мини-ЭВМ в рамках реализации проекта ВЦКП СО АН СССР. Обсуждаются основные принципы построения и архитектура двух компонент матобеспечения – ОС ДИРАК и САП МАСМ – определяющих системную операционную обстановку вычислительной системы на базе СМ ЭВМ.

В соответствии с концепцией создания и принятой архитектурой ВЦКП [1] мини-ЭВМ играет существенную роль в обеспечении функционирования всех вычислительных ресурсов, входящих в состав ВЦКП. А именно, предполагается, что мини-ЭВМ будет использоваться в рамках ВЦКП в качестве периферийного центра обработки информации (ПЦО), связанного процессора базового вычислительного комплекса (СВП) коммуникационного процессора (КП СПД) системы передачи данных, а также в качестве автономного вычислительного комплекса (АВВ).

Кроме того, важным является тот факт, что в качестве мини-ЭВМ в рамках центра коллективного пользования могут использоваться различные по типу, но приблизительно равноценные по своим вычислительным возможностям, вычислительные машины (считается, что такими мини-ЭВМ являются машины серии СМ). Очевидно, что в данных условиях возникает необходимость разработки такого матобеспечения, которое бы обеспечило переносимость программного продукта в рамках используемого в ВЦКП класса мини-машин.

Цель данного доклада – дать обзор основных работ, посвященных созданию математического обеспечения мини-ЭВМ в рамках проекта ВЦКП СО АН СССР [2,5]. В связи с этим авторов доклада следует рассматривать в качестве представителей соответствующего коллектива разработчиков.

Характер использования мини-ЭВМ в рамках ВЦКП таков, что необходима настраиваемость математического обеспечения на конкретный вариант функционирования вычислительной установки (когда мини-ЭВМ используется, например, либо как автономный ведомственный вычислительный комплекс, либо как коммуникационный процессор системы передачи данных и т.д.). Очевидно данный подход требует адаптируемости операционной системы к конкретному варианту использования мини-ЭВМ. Такого рода операционной системой является ОС ДИРАК.

ДИРАК обеспечивает работу мини-ЭВМ в режимах разделения времени, пакетной и дистанционной пакетной обработки.

За счет соответствующей проблемной надстройки (что определяется параметрами генерации системы), ДИРАК осуществляет управление мини-ЭВМ как ПЦО, СВП, КП СЦД или АВВ.

ОС ДИРАК оперирует с такими понятиями и объектами - задача, процесс, куст, устройство, рабочая область, событие, semaфор, информационный пакет.

Структура задачи в системе ДИРАК

ДИРАК различает задачи двух следующих типов: задачи универсального типа и задачи кассетного типа.

Задачи универсального типа представляют собой единый объектный массив, в котором ОС не различает никаких структурных единиц. В объектном массиве таких задач команды, константы и рабочие ячейки перемешаны.

В объектных массивах задач кассетного типа ОС выделяет две зоны: статическую (не меняющуюся в процессе выполнения) и динамическую (изменяемую при выполнении). Динамическая часть программы (кассета) является связной областью объектного массива и следует непосредственно за статической частью.

Задачи кассетного типа являются параллельно используемыми (реентерабельными). Изложенные принципы построения задач по сути дела не налагают существенных ограничений на ее разработку и, кроме того, эти принципы могут быть реализованы с помощью соответствующей системы автоматизации программирования.

Процессы. При инициировании задачи ОС ДИРАК порождает процесс. Процесс не есть сама задача, это есть единичное вы-

полнение задачи для определенного пользователя в определенное время. В каждый конкретный момент времени в системе может существовать сразу несколько процессов порожденных на базе одной и той же задачи. При запуске задачи универсального типа для образования процесса операционная система осуществляет дублирование объектного массива задачи. Несмотря на то, что в оперативной памяти уже может существовать процесс порожденный из той же задачи.

При инициировании задачи кассетного типа учитывается, что процесс для задачи такого типа есть выполнение статической части объектного модуля с использованием определенной кассеты. При этом статическая часть объектного модуля может использоваться и другими процессами, в то время как кассета всегда индивидуальна для каждого процесса.

Образовать процесс, т.е. запустить задачу в решение можно либо непосредственным образованием процесса по директиве другого процесса, либо помещением ее в пакет, подлежащий выполнению. Эти действия, а также и инициирование пакета, могут быть реализованы оператором системы, пользователем за терминалом или по директиве активно исполняемого процесса.

Наличие нескольких способов инициирования задачи позволяет программисту строить вычислительную работу таким образом, чтобы ее выполнение складывалось из выполнения набора взаимодействующих процессов.

Понятие куста. Один или несколько взаимодействующих процессов, выполняющихся на одной машине и обеспечивающих выполнение одной вычислительной работы, называется кустом.

Процессы в рамках куста не имеют отношения подчиненности. Между любыми процессами в рамках куста может быть установлена информационная (передача и получение сообщений) и управленческая (инициирование, приостановка процесса, синхронизация работы процессов и т.д.) связь.

При выполнении процессов куста операционная система выделяет по запросам процессов ресурсы, которые принадлежат кусту, а не отдельному процессу.

К ресурсам относятся - внешние устройства ввода, вывода и отображения информации; рабочие области (области прямоадресуемой памяти, которые операционная система выделяет кусту толь-

ко на время его существования); семафоры (двоичные семафоры Дейкстры); события; процессы; информационные пакеты (области памяти, выделяемые кусту на время его выполнения и предназначенные для накопления и последующей передачи информации в другие ЭВМ).

Перечисленные ресурсы называются объектами системы.

Как уже отмечалось, объекты определяются на уровне куста, а не на уровне процесса. В рамках куста объект появляется после того, как один (любой) из процессов выдал директиву "образовать объект", указав при этом свойства образуемого объекта и возможно математическое имя (последнее интерпретируется как одно из средств обобществления объекта). В результате обработки директивы "образовать объект" операционная система вырабатывает системное имя образованного объекта и возвращает его процессу. Образовать дважды один и тот же объект нельзя.

Для того чтобы сделать объект известным и доступным процессу, последний должен выдать директиву "открыть объект", указав при этом либо математическое, либо системное имя открываемого объекта. Во всех последующих директивах работы с объектами процессы должны использовать системное имя объекта.

Математическое имя объекта удобно использовать тогда, когда в процессе создания вычислительной работы (при выполнении представляющей собой набор взаимодействующих процессов - куст) статически (априорно) известны объекты, используемые кустом, т.е. еще на этапе проектирования они могут получить математические имена, под которыми они могут быть открыты различными процессами куста. Если же объекты по своей природе динамические, то эти объекты могут быть открыты, а следовательно, доступны только тому процессу, который их образовал, т.к. только он имеет их системные имена. Для того, чтобы эти объекты разделялись другими процессами куста, необходимо посредством механизма передачи сообщений передать процессам системные имена объектов.

Процесс теряет возможность работы с объектом после того, как он выдал директиву "закрыть объект". Объект исчезает из куста, после того как один (любой) из процессов куста выдал директиву "уничтожить объект".

Процесс может использовать любой из объектов монопольно, для чего ему необходимо выдать директиву "монополизировать объект". Данная директива приостанавливает все другие процессы, пытающиеся использовать монополизированный объект до поступления директивы "обобществить объект".

Итак над всеми объектами системы определены следующие директивы: образовать объект; открыть объект по математическому имени; открыть объект по системному имени; закрыть объект; уничтожить объект; монополизировать объект; обобществить объект.

События. Событием называется математический объект, с которым отождествляется некоторый физический факт, под воздействием которого меняется состояние системы в целом. Например, завершение чтения с магнитной ленты, завершение некоторого процесса и т.д. — все это можно считать событиями.

При помощи событий программист может управлять ходом выполнения процессов куста и координировать взаимодействие процессов.

Событие для процесса определяется значениями параметров "признак свершения события (П)" и "сопроводительная информация, характеризующая произошедшее событие (СОП)". Данная информация поставляется процессом, объявляющим событие произошедшим (в том числе и внутрисистемным процессом ОС).

Над объектом вида "событие" определены следующие директивы — ждать событие; объявить событие свершенным; опросить состояние события; объявить событие несвершенным.

Тот факт, что произошло некоторое событие, устанавливается либо по директиве "объявить событие", либо фиксируется операционной системой. Директива опроса состояния события позволяет процессу получить значения параметров П и СОП, характеризующих событие в данный момент. Процессу предоставляется возможность ожидать свершения некоторого события. Это осуществляется посредством директивы "ждать событие", которая приостанавливает процесс до того момента времени, когда указанное в директиве событие будет объявлено свершенным или будет уничтожено.

Семафоры. В системе определены двоичные семафоры Дейкстры и каноническая $V(S)$ -операция.

При вхождении в критический интервал, контролируемый семафором S , процесс выполняет $P(S)$ -операцию Дейкстры. При этом у семафора может накопиться очередь процессов, ожидающих его открытия. В канонической трактовке $P(S)$ -операции неопределено, какому из ожидающих процессов будет предоставлена возможность войти в свой критический интервал. В работе [6] была показана возможность и целесообразность учета приоритета процесса вошедшего в свой критический интервал. В связи с этим в ДИРАКе $P(S)$ -операция заменена операцией $P(S, \Pi)$, где Π - приоритет процесса на критическом интервале. В рамках ДИРАКа первым в решение будет инициирован процесс задавший большее значение Π в операции $P(S, \Pi)$. При равенстве значений Π порядок инициирования процессов не определен.

Устройства. Все устройства ввода, вывода и отображения информации, подключенные к системе, являются объектами системы.

Директивами обмена с такими устройствами должны предшествовать директивы образования и открытия объекта, соответствующего данному устройству. Такой подход позволяет использовать всем процессам куста одни и те же съемные носители, т.е. по сути дела разделять их. Для АЦПУ такой подход означает, что пользователю предоставляется возможным образом группировать выводимую информацию.

Рабочая область. Это объект операционной системы, который по требованию процесса выделяется кусту. на время существования последнего и представляет собой прямоадресуемую область памяти. Данный объект может быть использован только в директивах обмена с рабочей областью. Адресация внутри рабочей области начинается с нуля и осуществляется с точностью до слова.

Информационный пакет. В рамках центра коллективного пользования между ЭВМ циркулируют потоки информации. Источниками и приемниками этой информации являются процессы исполняемые на разных ЭВМ комплекса. Информационные пакеты являются объектами операционной системы, которые используются для накопления информации, предназначенной для передачи в другие ЭВМ комплекса. Пакет состоит из двух частей; структура одной из них -

- заголовок - определяется операционной системой, а на структуру информационной части пакета операционная система не налагает никаких ограничений.

Заголовок пакета определяется набором параметров, куда входят:

- имя приемника информационного пакета. Приемником может быть как задача, так и процесс на другой ЭВМ-приемнике;
- административное имя ЭВМ-приемника пакета;
- имя алфавита, используемого для представления информации в информационной части пакета;
- максимально возможная длина информационной части пакета.

Над объектом вида "информационный пакет" для процессов, формирующих и передающих пакет, определены следующие директивы - записать в пакет, передать пакет.

Для процессов, принимающих пакет, определены директивы - опрос наличия пакета, ждать информационный пакет, выдать параметры заголовка пакета, выдать очередную порцию информации из информационной части пакета.

Процессы как объекты куста. В рамках куста между процессами может быть установлена информационная и управленческая связь.

Информационная связь устанавливается посредством механизма сообщений. Сообщения имеют стандартный формат и состоят (как и пакеты) из заголовка и информационной части.

Заголовок сообщения определяется совокупностью трех параметров, куда входят: системное имя процесса-приемника сообщения; имя процесса-источника сообщения; длина информационной части сообщения в словах.

В ДИРАКе предусмотрен следующий набор директив для работы с сообщениями - передать сообщение, опросить очередь сообщений процессу, выдать сообщение.

Используя директиву "передать сообщение", один процесс может послать другому процессу куста сообщение. Директива "опросить очередь сообщений процессу" позволяет процессу узнать есть ли сообщения предназначенные ему. По директиве "выдать сообщение" операционная система переписывает сообщение предназначенное процессу, в его память. Сообщения предназна-

ченные процессу буферизуются в порядке поступления. Действия же, предусмотренные директивой "выдать сообщение" могут быть определены как над первым, так и над последним сообщением в очереди сообщений процессу.

Управленческая связь между процессами куста поддерживается следующим набором директив - инициировать процесс, передать процессу параметры инициирования, приостановить процесс, продолжить процесс.

При инициировании процесса (постановке его в очередь на обслуживание процессором) процесс-инициатор может передать иницилируемому процессу параметры инициирования. Параметры инициирования оформляются в виде сообщения процессу и выдаются инициированному процессу по директиве "дай параметры инициирования".

Приведенное здесь описание основных возможностей операционной системы слишком общее, чтобы в полной мере отразить ту системную операционную обстановку вычислительной системы, в которую погружены программы пользователей. Однако, мы надеемся, что оно (описание) дает достаточно полное представление о нашем подходе к проектированию ОС, об архитектуре последней и, наконец, раскрывает достаточно большой комплекс вопросов и проблем, которые пришлось решать группе разработчиков ДИРАК на этапах проектирования операционной системы.

Разработка операционной системы ведется в предположении, что задачи, которым предстоит выполняться под управлением ДИРАКа, либо пишутся на языке МАСМ, либо на структуру транслятора другого языка налагаются некоторые ограничения, так как в смысле реализации директив транслятора с языка МАСМ и операционная система ДИРАК представляет собой единую систему.

Конечно это не означает, что в системе "МАСМ-ДИРАК" недопустима реализация трансляторов других языков программирования, осуществляющих перевод программ сразу на уровень команд объектной ЭВМ без фиксации уровня языка МАСМ (хотя именно этот последний путь авторы считают наиболее целесообразным). Если же идти минуя уровень МАСМ, все равно, построенный таким образом транслятор должен иметь возможность выходить на промежуточный уровень, при котором можно было бы использовать механизм макрогенерации в том виде, в каком этот механизм

определен в системе "МАСМ-ДИРАК".

В данном докладе авторы не нашли возможности так же подробно, как это сделано при описании средств ОС, раскрыть языковые средства и конструкции МАСМ. Здесь приводятся лишь некоторые принципы, положенные в основу разработки языка.

Первым, наиболее важным принципом является принцип обеспечения переносимости программного продукта в рамках класса мини-ЭВМ, которые будут работать под управлением ОС ДИРАК. Этот принцип влечет за собой наличие в языке достаточно общих операторных конструкций, близких по способу изображения к конструкциям языка высокого уровня.

С другой стороны (и это второй принцип) языковые средства МАСМ должны позволить в полной мере использовать особенности технических средств.

Главным образом, реализация второго принципа в сочетании с первым может быть доступна за счет мощных средств макрогенерации, в частности, за счет развитой библиотеки системных, по возможности, машинно-независимых макросов. Средства макрогенерации должны включать возможность управления генерацией текста контекстом макрорасширения, в частности, значениями атрибутов объектов базового языка. Разумеется, пользователю должна быть предоставлена возможность работать в терминах команд машины, создавать свои личные макросы и хранить их в архиве системы.

Последний принцип, который имеет смысл упомянуть, связан с возможностью использования языка МАСМ как промежуточного при трансляции с языков более высокого уровня. Между прочим, реализация именно этого принципа обусловила выбор для представления основных вычислительных конструкций - выражений - в польской записи.

В заключение следует отметить, что технологический комплекс САП МАСМ используется в настоящее время в основном разработчиками ОС ДИРАК для программирования и отладки почти всех модулей операционной системы. Отдельные модули в большей своей части машинно-независимы и программируются на языке МАСМ. Это конечно же облегчит перенос ОС с мини-машин типа СМ-2 (М-7000) на другие, для которых будет реализован (и уже реализуется) соответствующий генератор образов (машин-

но-зависимая часть транслятора) операторных конструкций базового языка МАСМ.

Л и т е р а т у р а

1. МЕТЛЯЕВ Ю.В., МОСКАЛЕВ О.В., ЭФРОС Л.Б. Архитектура вычислительного комплекса (центра коллективного пользования). - В сб: "Вычислительная техника", Новосибирск, ВЦ СО АН СССР, 1976.
2. БЕЗРЯКОВ В.Н., ЭФРОС Л.Б. МАСМ - язык системного программирования для ЭВМ АСВТ М-7000, снабженной ОС ДИРАК. - В сб: "Обеспечение ВЦКП", Новосибирск, ВЦ СО АН СССР, 1977.
3. ЛАСКИН Л.Ф., ТРЕСКОВА С.П. Функциональные возможности ОС ДИРАК. - В сб: "Обеспечение ВЦКП", Новосибирск, ВЦ СО АН СССР, 1977.
4. ЖУРАВЛЕВА Г.Г., ЛАСКИН Л.Ф., ПОГРЕБНЯК В.Ф., СКОК Ю.С., ТРЕСКОВА С.П., ШКОЛЬНИК К.М. Операционная система ДИРАК: аспекты реализации. - В сб: "Программное и техническое обеспечение ВЦКП", Новосибирск, ВЦ СО АН СССР, 1978.
5. БЕЗРЯКОВ В.Н., ЛЕЛЬЧУК Т.И. Особенности реализации контекстно управляемой макрогенерации. - В сб: "Программное и техническое обеспечение ВЦКП", Новосибирск, ВЦ СО АН СССР, 1978.
6. ЭФРОС Л.Б. О приоритетном обслуживании параллельных процессов на критических интервалах. "Программирование", № 5, 1976.

БАЗЫ ДАННЫХ И ИХ НАЗНАЧЕНИЕ В ПРОЦЕССЕ СБОРА И ОБРАБОТКИ ИНФОРМАЦИИ

П. Либл

Математический институт ЧСАН, Прага, ЧССР

Введение

Базы данных, их теория и практика развивается в настоящее время в основном в связи с большими информационными системами. Свойства, которые в этом смысле являются определяющими для баз данных, следующие:

- большой объем данных в базе
- полнота базы в том смысле, что она отражает все важные черты описываемого отрезка мира
- доступ к данным разнообразных пользователей с разными интересами и разными правами доступа
- большое значение независимости данных

Группа сотрудников Математического института ЧСАН в течение последних лет столкнулась с задачей создать систему управления базой данных в немножко других обстоятельствах. Основной задачей является создать программное обеспечение для ЭВМ ЕС 8540 (заводское обозначение КА-10; раньше обозначалась ЕС 8507) чехословацкого производства, которая предназначена для сбора и предварительной обработки первичной информации. Машина сама малая, память 128 к со скоростью выше 10^6 сек⁻¹, для временного накопления данных имеются дисковые модули по 5 М и для сбора данных и обратной связи назначен богатый набор терминалов. Предполагается связь с машиной типа ЕС 10xx офф-лине посредством магнитной ленты и тоже он-лине специальным каналом.

База данных в сборе информации

Анализ конкретных практических приложений, где будет ЕС 8540 работать, показал следующую черту, которую мы считаем важной и интересной: Хотя первоначально поставленная задача говорит о пассивном сборе данных с терминалов и их

передаче, после какой-то обработки, на высшую машину, более подробный анализ задачи ведет к режиму, где данные на диске скорее служат для управления реальным процессом. Переход от сбора к управлению ведет через развитую обратную связь. Оказывается, что первичную информацию редко следует просто собрать и записать на диск. Скорее ее надо непосредственно сравнить с информацией уже имеющейся на диске и выдать о результате сравнения справку на терминал. Накопленные данные, вместе с информацией поступившей из высшей машины (напр. о плане продукции), служат источником ответов на вопросы; в дальнейшем шаге анализа они уже служат основанием для рекомендаций или даже приказов.

Картина сбора данных с терминалов через ЕС 8540 на машину ЕС10xx меняется, короче говоря, в такую, где на дисках ЕС 8540 накоплена структура данных, относительно постоянная, подвергающаяся непрерывному обмену информацией с терминалами с одной стороны и с машиной ЕС 10xx с другой стороны.

В так поставленном положении было принято решение смотреть на данные, накопленные на дисках, по идее как на базу данных.

Рассмотрим особенности, которые показывает база данных в таком положении, и обратим внимание на черты, которые общие с другими типами баз данных. Емкость дисковых модулей машины ЕС 8540 позволяет хранить всего несколько М информации. Это меньше чем считается привычным для настоящей базы данных. Первые проекты однако показали, что емкость дисков хватает для набора данных описывающих напр. месячный план производства цеха или малого завода или дневный оборот вагонов большой железнодорожной станции. Во всяком случае хорошо определенный отрезок реального мира, все существенные свойства которого описаны данными на дисках - основательное свойство базы данных.

Способ подхода к БД

Широко принята схема, по которой имеются 4 способа подхода: системный подход, подход программиста, параметрический подход и подход на основе коммуникационного языка; в качестве

пятого способа добавляются подход случайного пользователя. Приведем короткие характеристики только что указанных способов.

Системный подход самый простой: используя системные средства, напр. с пульта управления, сотрудник прочитает все содержимое дисков, или вручную изменит некоторое значение.

Подход программиста характерен тем, что сам процесс подхода не совпадает во времени с наличием значений данных: Программист пишет программу, которая работает так, что читается или меняется содержимое базы.

Параметрический подход, это действие у терминала, пример которого мы все знаем из бюро авиолиний: сотрудник, обычно сотрудница невысокой квалификации но приятного вида, имеет в своем распоряжении малый конечный набор типов транзакций; она выбирает тип, например бронирование места, задает значения по заданной схеме, и тем подход кончается, хотя часто еще следует выдача определенных значений - в нашем случае номер бронированного места.

Четвертый способ подхода самый популярный и теоретически хорошо разработан. Потребитель задает с терминала выражение на особом, именно коммуникационном языке, вследствие чего произойдет действие с базой данных, чтение или изменение. Существенная разница против параметрического подхода такая же, как разница между написанием заявления, приказа или письма и заполнением анкеты, бланкета справки или пропуска: против простого "языка" параметрического подхода стоит богатый, потенциально бесконечный коммуникационный язык.

Подход случайного пользователя отличается от предыдущих тем, что не требуется, чтобы потребитель знал структуру накопленных данных, подвергался определенной схеме или даже ясно знал, что ему надо. Стандартный пример тут домработница, которая звонит в справочное бюро, чтобы узнать, где сегодня продаются дешевые апельсины.

Оказалось, что свойства создаваемой системы управления, связанной со сбором первичной информации, накладывают определенные требования на подход к базе данных. Так, системный

подход оказывается опасным и не целесообразным, особенно с точки зрения защиты данных. Характер хранимой информации не допускает, чтобы системный программист имел возможность менять значения данных, а иногда даже не годится чтобы имел возможность прочитать значения, хотя он ответствен за структуру данных.

Подход программиста, вместе с параметрическим подходом, оказался основоположным в наших обстоятельствах. На самом деле, естественно, чтобы форма транзакции, формат вступающей или выступающей информации, определялся давно перед моментом фактической передачи данных. Программист, перед которым стоит задача запрограммировать скажем действия связанные с наличием определенного грузового вагона в поезде только что прибывшего в станцию, принимает решения о типе, формате и смысле данных читаемых с терминала, вместе с решением о контролях и действиях над этими данными в процессе их накопления в базу данных. Программа потом работает так, что она требует от сотрудника у терминала параметрический подход вполне определенной формы.

Коммуникационный язык несомненно прогрессивное средство подхода к базе данных. В наших обстоятельствах его применению соответствует положение, когда сотрудник, скорее всего заведующий, ответственностью владеющий работник, обращается к базе данных с только что придуманным вопросом, системой не предусмотренным. Коммуникационные языки назначенные для задавания вопросов, иногда тоже используются для изменения или дополнения содержимого базы данных. В нашем примере это соответствует случаю, когда ответственный работник решает вычеркнуть определенную строку из плана или не сделаную работу обозначить как сделаную. Уже примеры, а самое главное анализ задач показывает, что в сборе первичной информации и в управлении реальным процессом коммуникационный язык не владеет таким центральным положением, как в информационной системе, где база данных служит в основном источником ответов на вопросы. Скорее, коммуникационный язык используется к решению одиночных проблем, включая нестандартные вопросы на счет содержимого базы.

Действия над базой данных

Всякая база данных отражает существенные свойства определенного отрезка реального мира. Мир непрерывно меняется, все течет, итак содержимое базы данных, чтобы не отставать, должно также меняться все время. У некоторых баз данных эта черта меньше выразительна, когда отражаемый отрезок мира на первый взгляд не изменен. Приведем в качестве примера базы данных, описывающую определенный отрезок истории, скажем японскую политику двадцатых годов. Все действующие лица уже умерли, итак содержимое базы кажется постоянным. Но даже этот пример показывает относительность такой неизменности: японская история не меняется, а меняются наши знания о ней. Следовательно, база данных и в этом случае подвергается изменениям, каждое из которых отвечает уточнению или пополнению знаний об описываемой отрезке мира.

Обратим внимание на базу данных связанную со сбором первичной информации и управлением реальным процессом. Каждое появление новой информации должно быть связано с обращением к базе данных с целью внести туда новую информацию. Подходом к базе данных тоже проявится уточнение, когда окажется, что переданная информация была ошибочной или неполной - и такое положение надо считать вполне нормальным всюду, где первичную информацию передает человек, заботы которого в конечном счете касаются другого чем максимальной точности при нажатии кнопок терминала.

Важный вопрос - так называемый первоначальный ввод данных в базу. На первый взгляд кажется, что это действие, например перенос месячного плана с запоминающих средств высшей машины на ЕС 8540, является действием отличающимся от выше упомянутых. По смыслу оно действительно отличается: здесь речь идет не об изменении окружающего мира, которому отвечало бы изменение содержимого базы. Имеются однако соображения, по которым оказывается целесообразным использовать для первоначального ввода данных в принципе такие же программные средства, как и для всех других случаев ввода информации. Возможно применить здесь идею о пополнении знаний об описываемом отрезке мира: передавая первоначальную информацию по

строкам, давайте смотреть на каждую такую передачу как на новую - с точки зрения нашей базы данных - информацию. Те же соображения приходится применить к якобы блоковой передаче собранных данных на высшую машину.

Языковые средства

Есть важные и интересные вопросы, возникающие около баз данных, а именно в связи с защитой данных, моделью базы данных, и отношением между структурой базы данных и ее накоплением в запоминающем устройстве. Из-за нехватки места однако ограничимся языками около баз данных.

Язык является основополагающим понятием современного матобеспечения. Здесь мы не будем излагать общепринятые идеи о языках описания данных и языках манипуляции данными, о процедуральных языках, о самостоятельных языках и языках - гостях. Вместо того посмотрим, как на вопросы на счет языков отвечать в случае базы данных участвующей в сборе первичной информации и управлении реальным процессом.

Задача, которую здесь программирует программист, типично имеет следующую структуру: Чтение информации с терминала - контроль формальной исправности - контроль логической исправности - обращение к базе данных - контроль непротиворечивости с содержанием базы - формулировка и выдача сообщения о результате.

Здесь много действий, для Запрограммирования которых хороший процедуральный язык почти необходим - арифметические и логические действия с данными в процессе контроля, логическая схема управления в самых разнообразных случаях совпадения результатов контроля. Из этого вытекает естественность стремления выразить средствами богатого и современного языка программирования не только упомянутые действия, а в е, включая подход к базе данных. Где это невозможно, надо стремиться включить средства обращения к базе данных в виде языковых конструкций - гостя.

Когда принята идея, что основным, если не даже единственным, языковым средством должен быть современный процедуральный язык, сразу возникает вопрос, что использовать в качестве языка описания данных. Это сложный вопрос, так как средства

описания, имеющиеся в каждом высшем языке программирования, а именно средства для декларации типов дат и переменных, не совсем удобны для описания такой структуры, которой является база данных. Также как у языка манипуляции, и здесь наверно можно пополнить основной язык конструкциями со свойствами гостя.

Это некоторые соображения, которые возникают из основной идеи смотреть на структуру временно накопленных данных в задачах сбора информации и управления реальным процессом как на базу данных. Они, может быть, окажутся интересными и для всех, которые занимаются базами данных с других точек зрения.

ПРОБЛЕМЫ И ЭВОЛЮЦИЯ ВЫЧИСЛИТЕЛЬНЫХ ЦЕНТРОВ КОЛЛЕКТИВНОГО ПОЛЬЗОВАНИЯ

Г.И.Марчук, О.В.Москалев
ВЦ СО АН СССР, Новосибирск, СССР

АННОТАЦИЯ. В статье Г.И.Марчука и О.В.Москалева "Проблемы и эволюция вычислительных центров коллективного пользования" рассматриваются проблемы, связанные с созданием вычислительных центров коллективного пользования и принципы эволюции вычислительных комплексов коллективного пользования к локальным сетям ЭВМ на примере проекта "ВЦКП СО АН СССР".

В настоящее время основной тенденцией научно-технического прогресса является концентрация производства. В концентрации заложены большие возможности для более глубокой специализации, более активного внедрения научно-технических достижений. В процессе общей концентрации производства происходит и концентрация средств вычислительной техники. Электронная вычислительная техника проникает во все сферы производства на разных уровнях и берет на себя решение все более сложных задач, в основном оптимизационного характера, которые могут быть решены только с помощью наиболее мощных вычислительных средств, на машинах, комплексах и системах высокой производительности с большим числом емких носителей информации, с развитой коммуникационной сетью, связывающей мощные центральные процессоры с периферийным оборудованием, в том числе с ЭВМ меньшей производительности и большей специализации.

Таким образом становится типичной иерархическая структура вычислительных средств, когда в центре информационно-вычислительной системы находятся мощные ЭВМ или процессоры, далее, ближе к периферии малые машины и микропроцессоры, обеспечивающие интерфейс с различными средствами периферийного оборудования. Такая иерархическая структура информационно-вычислительных систем является естественным отображением современной тенденции коллективного использования вычислительной техники.

Кроме того, такая структура соответствует нынешним представлениям о том, какова должна быть эффективная архитектура мощных перспективных вычислительных машин и систем, так как между развитием современных вычислительных центров коллективного пользования (ВЦКП) и тенденциями, прослеживаемыми в архитектуре мощных ЭВМ будущего существует тесная связь. К этим тенденциям можно отнести: внедрение параллелизма на всех уровнях обработки информации, децентрализацию обработки данных и равномерное ее распределение по всем уровням и устройствам системы, асинхронное взаимодействие всех компонент и устройств вычислительной системы или машины, виртуализацию отдельных устройств и всей системы в целом, развитие специализации компонент системы и в частности ее процессоров, тегирование данных или их самоидентификацию, модульность и перестраиваемость конфигураций вычислительных систем.

Перечисленные тенденции все более полно реализуются в современных ЭВМ и проектах перспективных машин. Почти все эти тенденции присущи и вычислительному центру коллективного пользования, который можно представить, как сильно распределенную ЭВМ нового поколения. Проекты ЭВМ вберут в себя те архитектурные принципы, которые уже сейчас апробируются на структурах центров коллективного пользования. В связи с этим, задача состоит не только в том, чтобы найти правильное решение современных ВЦКП, но и провести при их создании тщательный и кропотливый анализ новых принципов распределенной обработки информации.

С одной стороны методология создания ВЦКП смыкается с методологией создания мощных вычислительных машин, с другой стороны ВЦКП уже небольшая сеть ЭВМ. С этой точки зрения ВЦКП присущи многие черты сетей с большими информационными банками данных, мощными потоками информации, передаваемой на значительные расстояния. Следовательно, разрабатывая центры коллективного пользования мы должны учитывать проблемы построения сетей ЭВМ.

Третья важная проблема, которая связана с развитием методологии вычислительных центров коллективного пользования — это проблема банков данных. В связи с тем, что в настоящее время невозможно (да и неразумно) заложить даже в самые мощные ЭВМ всю необходимую для пользователей информацию, естественным

решением является иерархическая организация банков данных, рациональное распределение банков между пользователем и самой машиной и их специализация.

Таким образом, проектирование и создание ВЦКП - серьезная научная и практическая задача, решение которой требует организации и проведения исследований в области математического и имитационного моделирования вычислительных систем и сетей и их компонентов, в области общесистемного, проблемного программного обеспечения и комплексирования технических средств, в области натуральных исследований на вычислительных комплексах для выяснения законов их функционирования и определения наборов фактических параметров для моделей, а также опыта проектирования систем и организации коллективного обслуживания пользователей в различных режимах вычислительного процесса.

В Сибирском отделении АН СССР эта задача решается в соответствии с проектом "ВЦКП СО АН СССР", который осуществляется объединенными усилиями Сибирского отделения АН СССР и заинтересованных ведомств с участием Единого центра математики и механики АН НРБ. Главными целями реализации проекта являются: обеспечение пользователей вычислительными и информационными мощностями на коммерческой основе, создание технической и технологической базы для развертывания прикладных программных систем, ориентированных на обеспечение научных исследований и развитие автоматизированных систем управления производством и территорией, исследование различных аспектов построения, реализации и эксплуатации подобных комплексов автоматизированной распределенной обработки информации.

Архитектура вычислительного комплекса коллективного пользования (ВККП) СО АН СССР отражает многомашинную конфигурацию, территориальное распределение вычислительных средств, разнотипность входящего в состав комплекса оборудования и обеспечивает возможности для реализации различных методов доступа пользователей к ресурсам комплекса через развитую сеть терминального оборудования и телекоммуникаций. При разработке архитектуры ВЦКП проведена серия модельных экспериментов и исследованы зависимости основных параметров функционирования от характеристик технических средств, программного обеспечения и

внешнего окружения. Такой комплекс представляет собой локальную сеть ЭВМ, хотя и имеет ряд отличительных черт по сравнению с развитыми сетями (рис. 1).

Абонентом комплекса коллективного пользования (АККП) может быть абонентский пункт, терминальное устройство, отдельная ЭВМ или развитый автономный вычислительный комплекс, удовлетворяющий условиям технического и программного интерфейса с ВККП. Абонент комплекса является для ВККП источником и приемником информации и управляющих воздействий (директив). Связь абонента с ВККП обеспечивается терминальной сетью периферийного центра обработки.

Периферийный центр обработки (ПЦО), разворачиваемый на базе мини-ЭВМ, концентрирует абонентов в основном по территориальному признаку. Функционально ПЦО обеспечивает прием, предварительную обработку и приведение к каноническому для системы виду поступающей от абонента информации и управляющих директив, первичную диспетчеризацию потоков заданий и сбор находящейся в его компетенции статистической информации для ВККП в целом, прием адресуемой абоненту информации и ее выдачу в требуемом формате. Таким образом, периферийные центры обработки обеспечивают как доступ абонентов к основным вычислительным и информационным мощностям, так и связь абонентов между собой. В последнем случае осуществляется лишь транзит сообщений абонентов друг другу без анализа их синтаксиса и семантики.

Основные же вычислительные и информационные мощности сосредотачиваются в базовых вычислительных комплексах (БВК) ВККП. Проектирование ВККП ведется с ориентацией на создание базовых вычислительных комплексов на основе ЭВМ БЭСМ-6, ЕС и МВК "Эльбрус", причем принципиальных ограничений для увеличения числа БВК и ПЦО в составе ВККП не имеется. Связь базового вычислительного комплекса с остальными компонентами ВККП реализуется через связной процессор (СВП), разворачиваемый на базе отдельной мини-ЭВМ. В рамках БВК связной процессор обеспечивает прием адресуемой БВК информации и ее приведение к каноническому для БВК формату, формирование заданий и диспетчеризацию работы БВК, сбор статистической информации о работе

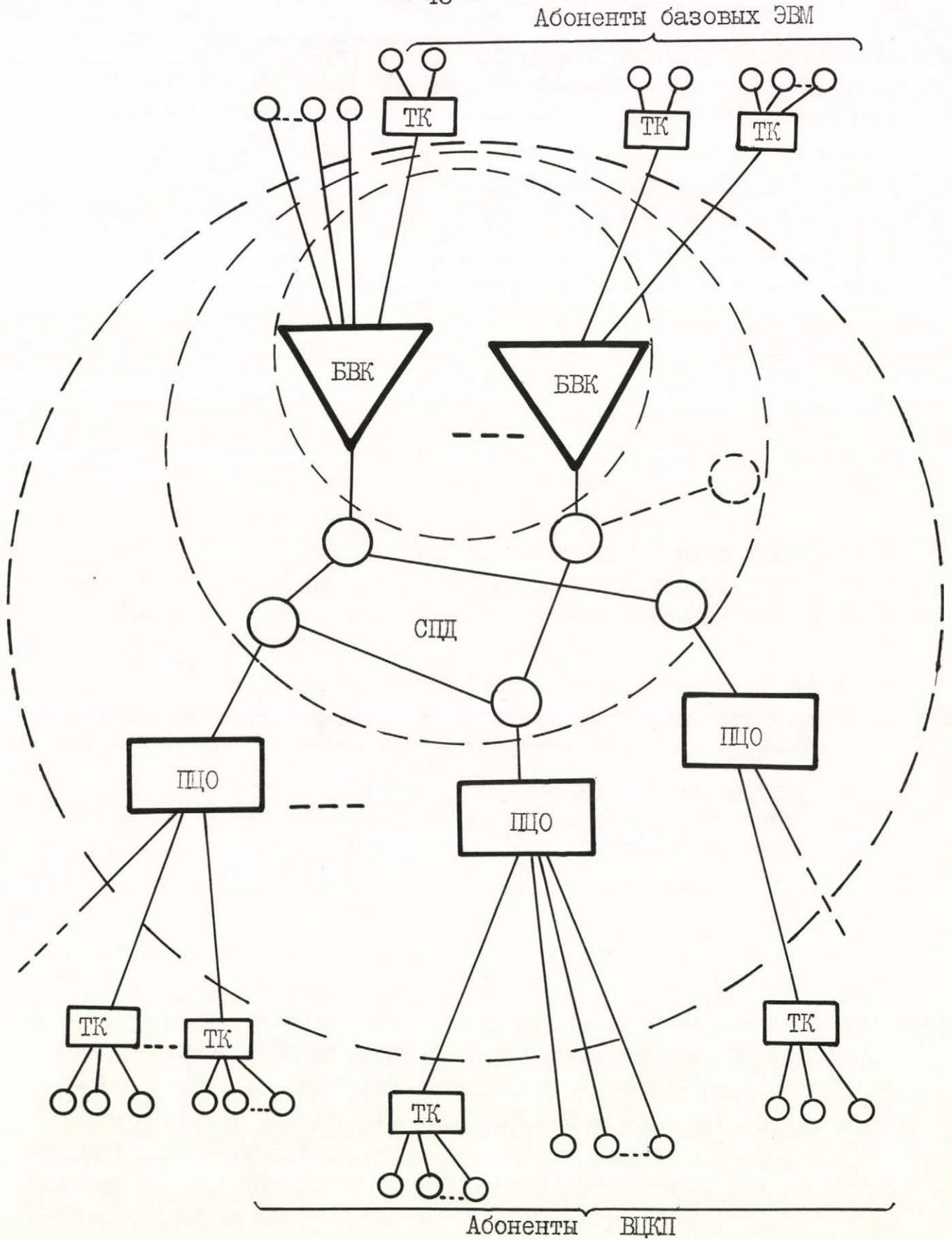


Рис. 1. Проект "ВЦКП СО АН СССР"
Функциональная схема

БВК, прием и передачу адресуемой другим компонентам ВККП информации. Функционально базовый вычислительный комплекс обеспечивает эффективные вычисления и обработку информационных данных в режиме дистанционной пакетной обработки по заданиям, формируемым связным процессором.

Связующим звеном комплекса коллективного пользования является система передачи данных (СПД), обеспечивающая передачу информации и управляющих воздействий между мини-ЭВМ, реализующими функции периферийных центров обработки информации и связных процессоров базовых вычислительных комплексов. СПД состоит из нескольких узлов соединенных линиями связи и обеспечивает установление соединений и передачу информации по методу коммутации пакетов. Узел СПД реализуется на основе мини-ЭВМ и специального канального оборудования.

Структура ВККП предусматривает доступ к отдельным ЭВМ БВК и их стандартным программным средствам как через ЦО, так и через собственные терминальные сети, организованные на магистральных линиях связи, но в этом случае не обеспечивается доступ абонентов к центральной информационной базе ВККП. В этом смысле обеспечивается только транзит сообщений АБВК в соответствующий ЦО и транзит сообщений АККП отдельной ЭВМ БВК, т.е. находясь за одним терминалом, можно быть либо абонентом ВККП, либо абонентом отдельной ЭВМ БВК. Кроме того, предоставляется возможность связи абонентов между собой для передачи текстовых сообщений.

Каждый из БВК (БЭСМ-6, ЕС, "Эльбрус") и ЦО, а также их возможные комбинации с использованием СПД могут быть автономной технико-технологической системой и на их основе могут быть созданы самостоятельные ВККП различного назначения и производительности.

В настоящее время завершается реализация первой очереди вычислительного комплекса коллективного пользования, в состав которой входят БВК БЭСМ-6, БВК-ЕС, два ЦО и СПД (рис.2).

Общесистемное программное обеспечение первой очереди представлено операционной системой ДИРАК, инструментальной системой автоматизации программирования МАСМ и системой автоматизации программирования ЭТА. Операционная система ДИРАК реализу-

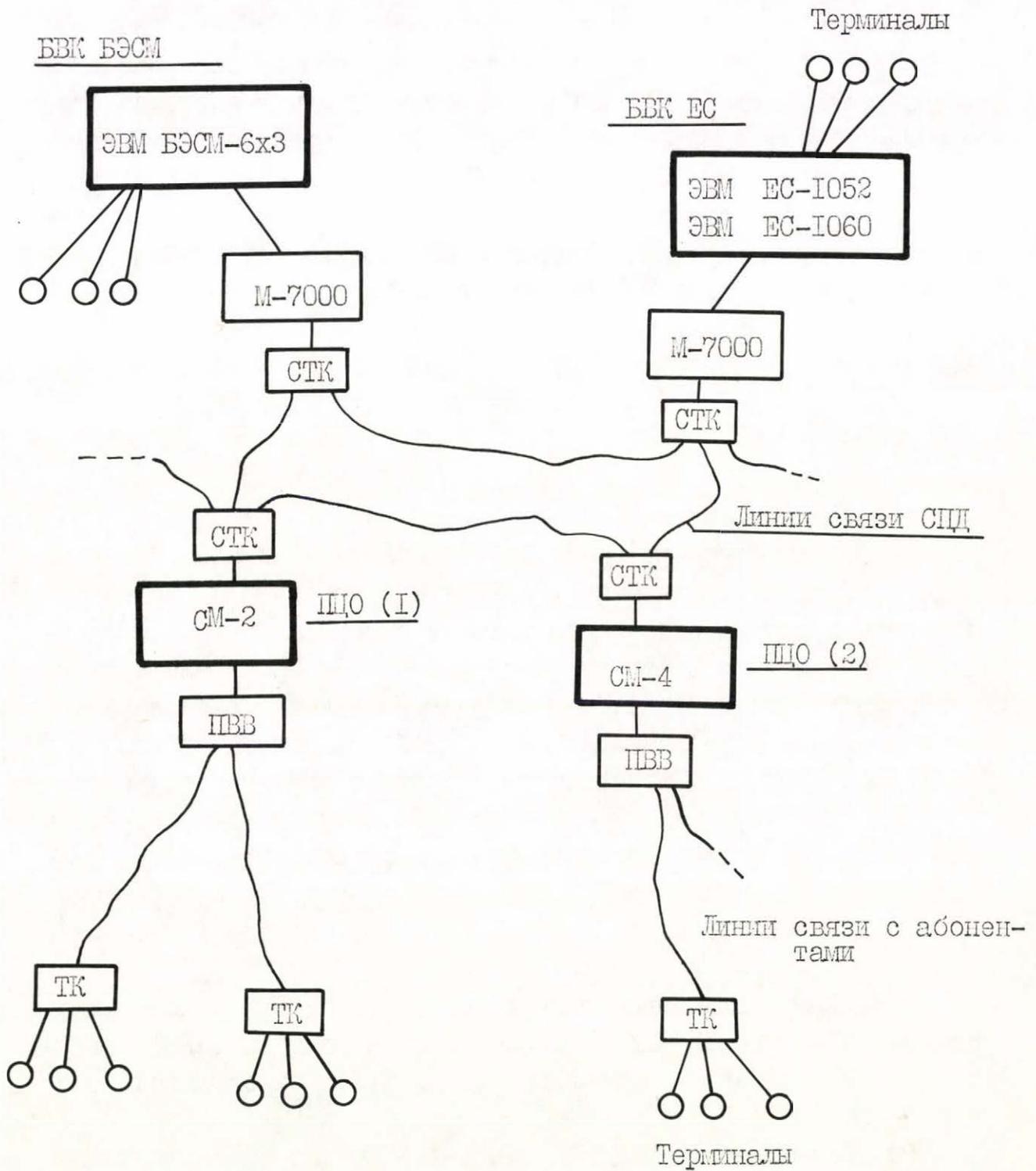


Рис. 2. Проект "ВЦКП СО АН СССР"
Первая очередь

ется на мини-ЭВМ М-7000 (СМ-2) и СМ-4 и может быть прогенерирована на работы в режимах: периферийного центра обработки с разделением времени, связанного процессора БВК БЭСМ-6, связанного процессора БВК ЕС, коммуникационного процессора СПД в совмещенной реализации с ПЦО и СВП. Инструментальная система автоматизации программирования с языка МАСМ обеспечивает подготовку программ для ЭВМ М-7000 и СМ-2, СМ-3 и СМ-4. Опытный вариант системы автоматизации программирования ЭТА обеспечивает разработку прикладных программных систем (пакетов прикладных программ) для ЭВМ базовых вычислительных комплексов.

В первой очереди ВЦКП возможно выполнение следующих режимов работы пользователей.

На БВК БЭСМ-6 - штатного режима работы подключенных терминалов по линии ПЦО-СВП-БЭСМ-6, с использованием коммутатора внешней памяти (КВП) и за счет модификации операционной системы ЭВМ БЭСМ-6.

На БВК ЕС осуществляется дистанционная пакетная обработка с формированием заданий на уровне языка управления заданиями с вводом в режиме разделения времени с терминалов ПЦО. Обеспечивается доступ к штатному банку данных ЭВМ ЕС в режиме дистанционной пакетной обработки с формированием в режиме разделения времени с терминалов ПЦО на проблемно-ориентированном языке контакта. В дальнейшем будет осуществлено расширение номенклатуры терминальных устройств, а также количественное и качественное расширение программных процессоров, обеспечивающих доступ с терминалов ПЦО к штатным прикладным программным системам с формированием заданий на работу на проблемно-ориентированных языках контакта.

Мини-ЭВМ используются в качестве ПЦО, СВП БВК и коммуникационных процессоров СПД. Процессоры ввода-вывода (ПВВ), терминальные и сетевые контроллеры (ТК и СТК) выполняются в стандарте "КАМАК".

Развитие проекта "ВЦКП" в 1981-1985 годах ориентировано на создание академической региональной информационно-вычислительной сети (АРС) СО АН СССР и предполагает развитие универсальной коммуникационной сети ЭВМ СО АН, создание специализированных коммуникационных узлов, обеспечивающих взаимодейст-

вие сети СО АН с Академсетью, а также развитие и подключение к коммуникационным узлам сети СО АН автономных вычислительных комплексов различной проблемной ориентации, построенных в основном на базе мини-ЭВМ. В области программного обеспечения будет осуществлено развитие базовых систем для мини-ЭВМ, обеспечивающих эффективное функционирование технических средств сети, а также банков данных и диалоговых систем типа БЭИСИК, программных процессоров реализации протоколов связи и системы сбора и обработки статистической информации. Будут созданы общесистемные средства технологического обеспечения разработки прикладных программных систем (пакетов прикладных программ), ориентированных на работу в условиях сети ЭВМ и осуществлена разработка прикладных программных систем, ориентированных на использование возможностей архитектуры сети СО АН и специфику вычислительного процесса. В части организационно-технологического обеспечения будут развиты принципы организации вычислительного процесса в условиях вычислительного комплекса с сетевой архитектурой и работы с распределенными банками данных.

Таковы, на наш взгляд, проблемы, связанные с созданием вычислительных центров коллективного пользования и принципы эволюции вычислительного комплекса коллективного пользования к локальной сети ЭВМ, рассмотренные на примере проекта "ВЦКП СО АН СССР", который реализуется в основном на серийных средствах вычислительной техники и обеспечивает значительное повышение эффективности их использования.

Л и т е р а т у р а

1. МАРЧУК Г.И., МОСКАЛЕВ О.В. О проекте создания территориального ВЦКП в Новосибирском Академгородке. (Концепция). Препринт ВЦ СО АН, Новосибирск, 1976.
2. МЕГЛЯЕВ Ю.В., МОСКАЛЕВ О.В., ЭФРОС Л.Б. Архитектура вычислительного комплекса (центра) коллективного пользования СО АН СССР. - "Обеспечение ВЦКП", сб., Новосибирск, ВЦ СО АН СССР, 1976.

3. МАРЧУК Г.И., МОСКАЛЕВ О.В. Методология создания территориального вычислительного центра коллективного пользования Сибирского отделения АН СССР. "Кибернетика", № 6, К., ИК АН УССР, 1977.
4. МАРЧУК Г.И., КУЗНЕЦОВ Е.П., МОСКАЛЕВ О.В., МЕТЛЯЕВ Ю.В., ЭФРОС Л.Б. О программе работ по созданию вычислительного комплекса (центра) коллективного пользования в Новосибирском научном центре СО АН СССР (проект "ВЦКП"). - В сб.: "Программное и техническое обеспечение ВЦКП", Новосибирск, ВЦ СО АН СССР, 1978.
5. ЭФРОС Л.Б. Концептуальный анализ программных систем. "Управляющие системы и машины", № 2, К., "Наукова думка", 1979.
6. МИТРОФАНОВ Ю.И., КУРБАНГУЛОВ В.К., ШУЛЬГА Н.П. Результаты моделирования ВЦКП СО АН СССР показательными сетями обслуживания. - В сб.: "Программное и техническое обеспечение ВЦКП", Новосибирск, ВЦ СО АН СССР, 1978.
7. BARNEV P. Systems for information servicing of collectivities, Serdica, 4, 1978 .

ТЕХНИЧЕСКИЕ СРЕДСТВА ВЦКП СО АН СССР

Ю.В.Метляев

ВЦ СО АН СССР, Новосибирск, СССР

Основные предпосылки и принципы создания ВЦКП в Сибирском отделении АН СССР изложены в работах [1,2,3]. Характерные особенности комплекса технического оборудования создаваемого ВЦКП заключаются в следующем:

многомашинная конфигурация, где в едином комплексе используются ЭВМ различного типа и класса;

территориальная распределенность технического оборудования, объединенного в единую систему линиями связи;

функциональная специализация оборудования, что является одним из основных факторов обеспечения высокой эффективности использования конкретных технических средств;

двухуровневая система дистанционных связей, где локальные терминальные сети используются в сочетании с системой передачи данных между ЭВМ.

Вопросы технической реализации отдельных подсистем ВЦКП ранее рассматривались в нескольких статьях [4,5,6]. Цель данного сообщения – осветить комплексно ряд вопросов, связанных с конкретной технической реализацией первой очереди создаваемого ВЦКП, где в минимальном объеме будут представлены все основные его компоненты: два базовых вычислительных комплекса (БВК), два периферийных центра обработки (ПЦО), четыре узла системы передачи данных (СПД).

Конфигурация и состав технического оборудования первой очереди реализуемого комплекса приведены на рис.1. В его состав входят: БВК БЭСМ-6; БВК ЕС, ПЦО на базе мини-ЭВМ СМ-2; ПЦО на базе мини-ЭВМ СМ-4; терминальные комплексы ПЦО, реализуемые на базе системы КАМАК; СПД, объединяющая БВК и ПЦО в единый комплекс.

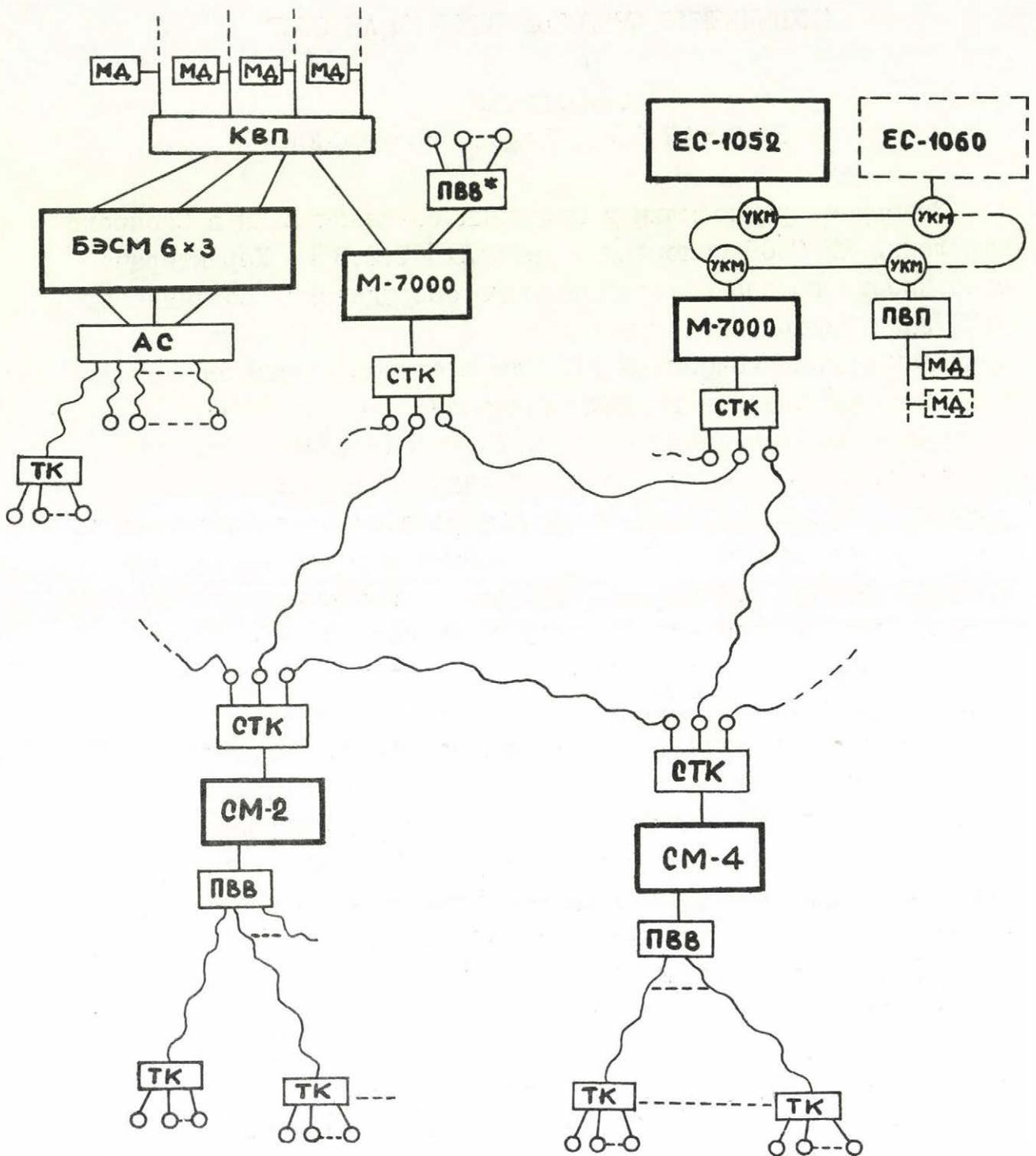


Рис. 1. Технические средства ВЦКП СО АН (I очередь)

БВК на основе ЭВМ БЭСМ-6

Автономный вычислительный комплекс состоит из трех ЭВМ БЭСМ-6, связанных общим полем дисковой памяти [7]. С помощью коммутатора внешней памяти (КВП) все ЭВМ БЭСМ-6 имеют доступ к четырем коммутаторам магнитных дисков. В составе комплекса используются также планшетные графопостроители, электронный фотопостроитель, устройства вывода информации на микрофиши и другие специализированные устройства. В комплекс на уровне КВП включены две мини-ЭВМ (М-6000 и НР-2000). ЭВМ М-6000 осуществляет вывод информации на микрофиши, а также вывод изображений. ЭВМ НР-2000 обеспечивает работу 32-х терминалов в режиме разделения времени на собственных ресурсах, а также предоставляет пользователям возможность с этих же терминалов формировать и редактировать программы для ЭВМ БЭСМ-6. ЭВМ БЭСМ-6 имеют собственную сеть терминальных устройств (в основном - дисплеи "Видеотон-340"). Разрабатываемая аппаратура сопряжения (АС) позволит расширить терминальную сеть комплекса, и обеспечит возможности для подключения любого терминала к любой из трех ЭВМ БЭСМ-6, а также подключить к БЭСМ-6 терминальные комплексы (ТК), которые позволяют осуществлять дистанционный ввод заданий и получение результатов.

С целью обеспечения функционирования этого действующего комплекса в качестве БВК ВЦКП в его состав включается связанной процессор, реализуемый на базе мини-ЭВМ М-7000. Связь М-7000 с любой из трех ЭВМ БЭСМ-6 осуществляется через КВП и общую дисковую память. При этом предполагается, что абоненты ВЦКП могут пользоваться всеми штатными возможностями ЭВМ БЭСМ-6. Для отладки программного обеспечения и имитации различных режимов работы пользователей в системе ВЦКП к ЭВМ М-7000 подключается процессор ввода-вывода (ПВВ), с помощью которого обеспечивается связь с восьмью терминалами и возможности работы связанного процессора БВК БЭСМ в режиме ПЦО.

БВК на основе ЭВМ ЕС

Основная цель создания многомашинного комплекса для ЭВМ ЕС состоит в том, чтобы реализовать общий по отношению к нескольким ЭВМ ЕС канал дистанционной связи с периферийным оборудованием (абонентами ВЦКП), обеспечить более эффективный ре-

жим загрузки ЭВМ ЕС на основе создания системы автоматизированного управления работой многомашинного комплекса, а также обеспечить возможности наращивания (изменения) общей производительности используемого парка ЭВМ ЕС. Кроме этого, наличие в комплексе нескольких однотипных машин позволяет реализовать режим проблемно-ориентированного использования отдельных машин в случае необходимости обеспечения многофункциональности комплекса или режима "горячего резерва" в случае необходимости обеспечения повышенной надежности его работы.

В БВК ЕС объединяются несколько (до четырех) ЭВМ ЕС, мини-ЭВМ (М-7000, СМ-2), специализированный процессор внешней памяти (ПВП). При этом используется следующая функциональная специализация ЭВМ в комплексе:

- мини-ЭВМ является связным процессором (СВП) комплекса, с помощью которого выполняются функции системного диспетчера (управление работами на уровне ввода/вывода заданий и распределения заданий между ЭВМ ЕС), а также функции связи комплекса с СПД;

- ЭВМ ЕС используются как рабочие ЭВМ комплекса (выполнение заданий в пакетном режиме), отдельная ЭВМ ЕС работает автономно по отношению к другим ЭВМ ЕС и имеет связь только с диспетчерской ЭВМ (прием заданий, выдача результатов, обмен служебной информацией);

- ПВП рассматривается как специализированная ЭВМ комплекса, на основе которой реализуются системные средства хранения информации (оперативный буфер ввода-вывода, системный архив, базы данных различного назначения).

Объединение ЭВМ в комплекс осуществляется с помощью мультиплексной магистрали, которая состоит из кабельных шин связи замкнутых в кольцо. В местах подключения ЭВМ к магистрали создаются узлы кольцевой магистрали (УКМ), где обеспечивается сопряжение с конкретной ЭВМ, коммутация, контроль и регенерация передаваемых сообщений. Количество параллельных линий связи в магистрали - 16, каждая линия физически выполнена в виде пары свитых проводов. Максимальная скорость передачи информации по магистрали $16 \cdot 10^6$ бит в секунду, при тактовой частоте передачи сигналов в линии 1 МГц. Максимальная длина соедини-

тельных линий между смежными узлами магистрали 150 метров. Общая длина магистрали определяется конфигурацией комплекса (количеством ЭВМ и их размещением), в общем случае длина магистрали не ограничена, т.к. допускается использование УКМ, выполняющих только функции регенерации сигналов в магистрали.

Подключение ЭВМ ЕС к магистрали осуществляется через мультиплексный канал с использованием стандартного интерфейса ввода-вывода ЭВМ ЕС. Подключение диспетчерской мини-ЭВМ к магистрали осуществляется с помощью программно-управляемого канала связи, т.е. с использованием рабочего процессора мини-ЭВМ для приема-передачи данных и управляющих команд. Подключение ПВП к магистрали осуществляется в соответствии с конкретной схемой реализации ПВП и в соответствии с общим алгоритмом функционирования магистрали. В простом варианте ПВП реализуется в виде концентратора нескольких внешних накопителей с использованием соответствующих устройств управления накопителями и простого процессора обеспечивающего выполнение команд (директив) обмена информацией с внешними накопителями.

При любом варианте состава комплекса приоритетное положение в отношении реализации средств управления и организации обмена информацией между ЭВМ занимает мини-ЭВМ, в которой реализуется диспетчер комплекса. Другие ЭВМ (ЭВМ ЕС, ПВП или другие специализированные ЭВМ) рассматриваются как "подчиненные", наиболее типичная в данном случае связь в комплексе - связь между диспетчерской ЭВМ и "подчиненной" ЭВМ. Таким образом обеспечивается централизованное управление работой комплекса со стороны диспетчерской ЭВМ и автономность функционирования прочих ЭВМ по отношению друг к другу.

Периферийные центры обработки

Основное функциональное назначение ПЦО заключается в том, чтобы обеспечить связь с абонентами ВЦКП и выполнение функций предварительной обработки информации, в частности, функций связи абонентов с БВК. Основным режим работы ПЦО - оперативное обслуживание множества абонентов в режиме разделения времени.

Техническая реализация ПЦО осуществляется с помощью мини-ЭВМ, в качестве которых могут быть использованы ЭВМ серии М-7000, СМ-2; или ЭВМ серии СМ-3, СМ-4. На основе мини-ЭВМ

создается локальная абонентская сеть, обеспечивающая концентрацию и дистанционное подключение до 64 абонентов. В качестве абонентов могут быть использованы: отдельный терминал или простой абонентский пункт, отдельная ЭВМ, программно-управляемый абонентский пункт, специализированный информационно-вычислительный комплекс. С целью реализации эффективной многоканальной связи ПЦО с абонентами используется терминальный комплекс, состоящий из специализированного процессора ввода-вывода (ПВВ) и нескольких (до восьми) дистанционно подключаемых к ПВВ терминальных контроллеров (ТК). ПВВ подключается к мини-ЭВМ на уровне основного процессора и выполняет функции мультиплексного канала, а также функции предварительной обработки, анализа и контроля символьной информации. Отдельный ТК рассчитан на подключение до восьми устройств (абонентов) и выполняет функции управления работой конкретных устройств. ТК подключается к ПВВ с помощью соединительных линий и соответствующих линейных адаптеров. Длина отдельной линии связи составляет порядка 10 км, скорость передачи в линии - порядка $20 \cdot 10^3$ бит в секунду. Пропускная способность ПВВ составляет порядка 10-15 Кбайт в секунду, что обеспечивает параллельную работу до 64 абонентов со средней скоростью порядка 150 байт в секунду.

Базовая комплектация мини-ЭВМ ПЦО: оперативная память - порядка 96К байт, быстрая внешняя память (диски с фиксированными головками) - 1-2М байт, внешняя память (диски с подвижными головками) - порядка 10М байт.

Система передачи данных

СПД - система межмашинных каналов, объединяющая БВК и ПЦО в единый комплекс. Технические средства СПД ориентированы на реализацию каналов связи протяженностью до 20 км и на передачу по ним информации по принципу коммутации пакетов. В состав СПД входят несколько идентичных узлов, соединенных линиями связи в соответствии с некоторой принятой конфигурацией. Отдельный узел состоит из коммуникационного (сетевое) процессора, где реализуются программные протоколы СПД и ее буферная память, и сетевого контроллера, где сосредоточено линейное

оборудование и осуществляется предварительная обработка и коммутация сетевых пакетов. Отдельная линия связи между узлами выполняется в виде четырех соединительных проводов, обеспечивающих в общем случае дуплексную передачу данных.

В состав СТК входят три линейных адаптера, буферная память для пакетов, модуль связи с мини-ЭВМ (коммуникационным процессором), а также управляющий процессор, с помощью которого реализуется линейный протокол обмена данными между смежными узлами СПД. Наличие в составе СТК трех линейных адаптеров, каждый из которых рассчитан на дуплексный режим приема/передачи данных, обеспечивает возможность создания независимых или дублирующих путей передачи сообщений между любой парой узлов СПД. Линейный адаптер состоит из двух модулей - приемника и передатчика, каждый из которых имеет буферную память для хранения одного пакета, а также соответствующие средства синхронизации, контроля, формирования и декодирования управляющих сообщений. Это позволяет в рамках двух связанных линейных адаптеров аппаратно реализовать выполнение основной части линейного протокола, в качестве которого используется протокол HDLC.

С помощью СТК, имеющем в своем составе управляющий процессор, выполняется простой алгоритм маршрутизации пакетов - в зависимости от адреса назначения пакета и в соответствии с заложенной в каждом узле таблицей маршрутов выбирается основное или резервное направление передачи конкретного пакета.

Другая часть функций узла СПД - протокол обмена сообщениями между узлами, протокол обмена с источниками или потребителями сообщений в рамках БВК или ИЦО, а также ряд служебных функций СПД, выполняются с помощью программного коммуникационного процессора, который реализуется с помощью мини-ЭВМ в данном случае совмещенного с ИЦО или БВК.

Заключение

Основной принцип технической реализации ВЦКП в Сибирском отделении АН СССР - использование стандартного серийного вычислительного оборудования, а также магистрально-модульной системы КАМАК, на базе которой реализуются специализированные системные средства создаваемого комплекса. Предусматривается

использование нескольких (15+20) базовых модулей КАМАК, с помощью которых будут компоноваться различные по своему назначению устройства (ПВВ, ТК, СТК, УКМ). Большая часть модулей предназначена для сопряжения разнотипного оборудования с магистралью КАМАК и тем самым будет обеспечена его компоновка в едином комплексе. Несколько "системных" модулей предназначены для реализации достаточно сложных в функциональном отношении устройств на основе принципов микропрограммирования. Используемый при этом модуль — управляющий процессор крейта КАМАК (УПК), обеспечивает выполнение микропрограммы, "защитой" в постоянном запоминающем устройстве (модуль ПЗУ) объемом до 1К команд со скоростью порядка $2 \cdot 10^6$ операций в секунду.

Для реализации линий связи в СПД, а также абонентских линий связи в ПЦО предусматривается прокладка на территории Новосибирского научного центра специальной кабельной сети. При этом в отдельных научных институтах и организациях будут устанавливаться кроссирующие устройства, с помощью которых будет задаваться необходимая коммутация соединительных линий в СПД и в локальных абонентских сетях. Связь с абонентами, расположенными в других районах г. Новосибирска или в других городах, будет осуществляться через районную АТС с использованием стандартных телеграфных или телефонных каналов связи и соответствующей аппаратуры передачи данных.

Л и т е р а т у р а

1. МАРЧУК Г.И., МОСКАЛЕВ О.В. О проекте создания территориального ВЦКП в Новосибирском Академгородке. (Концепция). Препринт ВЦ СО АН СССР. Новосибирск, 1976.
2. МАРЧУК Г.И., МОСКАЛЕВ О.В. Методология создания территориального вычислительного центра коллективного пользования Сибирского отделения АН СССР. "Кибернетика", № 6, К., ИК АН УССР, 1977.

3. МЕТЛЯЕВ Ю.В., МОСКАЛЕВ О.В., ЭФРОС Л.Б. Архитектура вычислительного комплекса (центра) коллективного пользования СО АН СССР. В сб.: "Вычислительная техника", ВЦ СО АН СССР, Новосибирск, 1976.
4. МАРЧУК Г.И., КУЗНЕЦОВ Е.П., МОСКАЛЕВ О.В., МЕТЛЯЕВ Ю.В., ЭФРОС Л.Б. О программе работ по созданию вычислительного комплекса (центра) коллективного пользования в Новосибирском научном центре СО АН СССР /проект "ВЦКП"/. В сб.: "Программное и техническое обеспечение ВЦКП", Новосибирск, ВЦ СО АН СССР, 1978.
5. МЕТЛЯЕВ Ю.В. Базовый вычислительный комплекс на основе ЕС ЭВМ. В сб.: "Обеспечение ВЦКП", ВЦ СО АН СССР, Новосибирск, 1977.
6. МЕТЛЯЕВ Ю.В. Периферийный центр обработки на базе мини-ЭВМ (М-7000, СМ-2) и системы КАМАК. В сб.: "Программное и техническое обеспечение ВЦКП", ВЦ СО АН СССР, Новосибирск, 1978.
7. КУЛЬКОВ Н.В. Многомашинный комплекс универсального применения. Тезисы докладов Всесоюзной конференции "Многомашинные системы автоматизации научных исследований", ИЭ и ВТ Латв.ССР, Рига, 1978.

ОБ ОДНОМ ПОДХОДЕ ПРИ ВКЛЮЧЕНИИ
ЦИФРОВОГО ТЕРМИНАЛА ИЗОТ 8500
В СИСТЕМЫ КОЛЛЕКТИВНОГО ПОЛЬЗОВАНИЯ

Радулов Н.Р.

Комитет по единой системе социальной информации, София

Владиков И.Л., Калчева М.Т.

Институт вычислительной техники, София, Болгария

На современном этапе развития вычислительной техники для проектирования систем, работающих в реальном масштабе времени с коллективным использованием базой данных, непрерывно растут требования к гибкости создания терминальных сетей, зависящих от потребностей конкретных потребителей. В соответствии с этими требованиями в НРБ разработана система ЕСТЕЛ, которая ориентирована на коммуникации между ЭВМ и терминалами (или терминальными станциями). Система управления информацией потребителя (СУИП), является программной компонентой системы ЕСТЕЛ, с помощью которой создаются системы для телеобработки данных с распределённым использованием ресурсов терминальной сети.

Непрерывное расширение сферы применения системы ЕСТЕЛ в народном хозяйстве делает необходимым использование терминалов с новыми функциональными возможностями. Одним из них является терминал ИЗОТ 8500.

Терминал ИЗОТ 8500 имеет три компоненты ввода:

1. Клавиатура – для ввода цифровой информации и специальных символов типа точка, минус, пробел.
2. Устройство чтения перфожемонов (содержащих постоянную информацию) – для ввода десяти разрядного числа.
3. Устройство чтения перфокарт – для ввода буквенно-цифровой информации со стандартных перфокарт (до 80 байтов).

Компоненты вывода устройства ИЗОТ 8500 следующие:

1. Цифровой дисплей с 13 позициями, для цифр $0 \div 9$, десятичной

точки, минуса и пробела.

2. Печатающее устройство с 16 позициями (13 цифровых, пробел и две служебные позиции для специальных знаков и символов).

Все компоненты ввода/вывода, за исключением устройства чтения перфокарт, встроены в корпус терминала. Его минимальная конфигурация включает одну компоненту ввода (клавиатура) и две компоненты вывода (печатающее устройство и цифровой дисплей). Остальные компоненты являются необязательными.

Терминал ИЗОТ 8500 разработан на базе микропроцессора с встроенными модемами. Он небольшой по объёму (356x345x142 мм) и легко обслуживается. У него есть буфер ввода/вывода размером 128 байтов. Терминал позволяет многоточечное подключение к линии связи (рис.1).

Функции, связанные с управлением терминалами различных видов, включённых в терминальную сеть системы СУИП, реализуются управляющей программой терминалов (УПТ) 1,2. Программа УПТ имеет модульную структуру, дающую возможность расширять терминальную среду системы СУИП новыми типами терминалов. Она состоит из общей управляющей логики и из модулей, зависящих от типа терминала. Общая логика сканирует линии, проверяя их активность. Если обнаружена активность на некоторой линии, управление передаётся модулю, зависящему от типа терминала для исполнения требуемых действий. Программа УПТ использует информацию о состоянии терминальной сети, которая сохраняется и динамически изменяется в Таблице управления терминалами (ТСТ). Эта программа работает в системе СУИП как самостоятельная задача с самым высшим приоритетом.

Проблемы программного обеспечения терминала ИЗОТ 8500 в системе СУИП обуславливаются в основном конструктивными ограничениями для вывода сообщения.

Сообщения системы СУИП, относящиеся к терминалу ИЗОТ 8500, получаются, управляются, диспетчеризируются системными программами: Управление ненормальными состояниями программ и Управ-

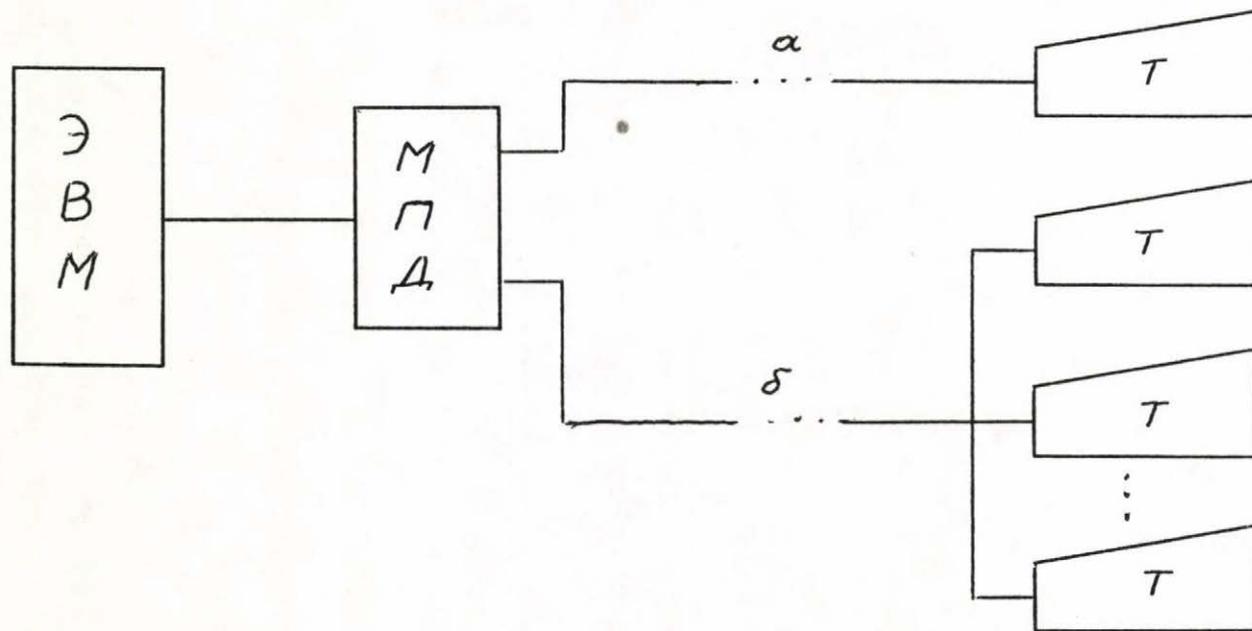


Рис.1 Способы подключения терминалов
ИЗОТ 8500

а. точка-точка

б. многоточечная связь

ление ненормальными состояниями терминалов, а также обслуживающими программами: Главный терминал, Подключение к системе и Отключение от системы.

Для решений этих проблем пришлось описать новый класс терминалов (цифровых) в Таблице управления терминалами (ТСТ). Созданы макроинструкции

```
FQWTCT TYPE=ERRMSG и
FQWTCT TYPE=RDMSG
```

для генерирования неповторяющихся цифровых последовательностей, которые посылаются потребителям каждой линии, с которой связаны терминалы цифрового класса. В каждой из перечисленных выше программ введена логика для выделения цифрового класса терминала, передающая управление подпрограмме, которая посылает соответствующее сообщение терминала (рис.2).

Сообщение о готовности к работе (*RDMSG*) используется системой СУИП:

- после успешного подключения к системе;
- в ответ на заявку для чтения в случае, если она не находится в последовательности писать-читать.

Сообщение об ошибке (*ERRMSG*) используется системой СУИП, когда:

- введён неправильный код транзакции;
- допущена ошибка при подключении к системе или отключении от системы;
- транзакция, связанная с цифровым терминалом, закончилась ненормально.

Эти цифровые сообщения используются прикладными программами с помощью соответствующем адресированием к ним.

Терминал ИЗОТ 8500 имеет алгоритм работы, аналогичный терминальной станции ЕС 8501, что позволило использовать один и тот же модуль в УПТ, зависящий от типа терминала. В общей части программы УПТ добавлена логика, которая передаёт управление модулю для терминалов ИЗОТ 8500 и ЕС 8501.

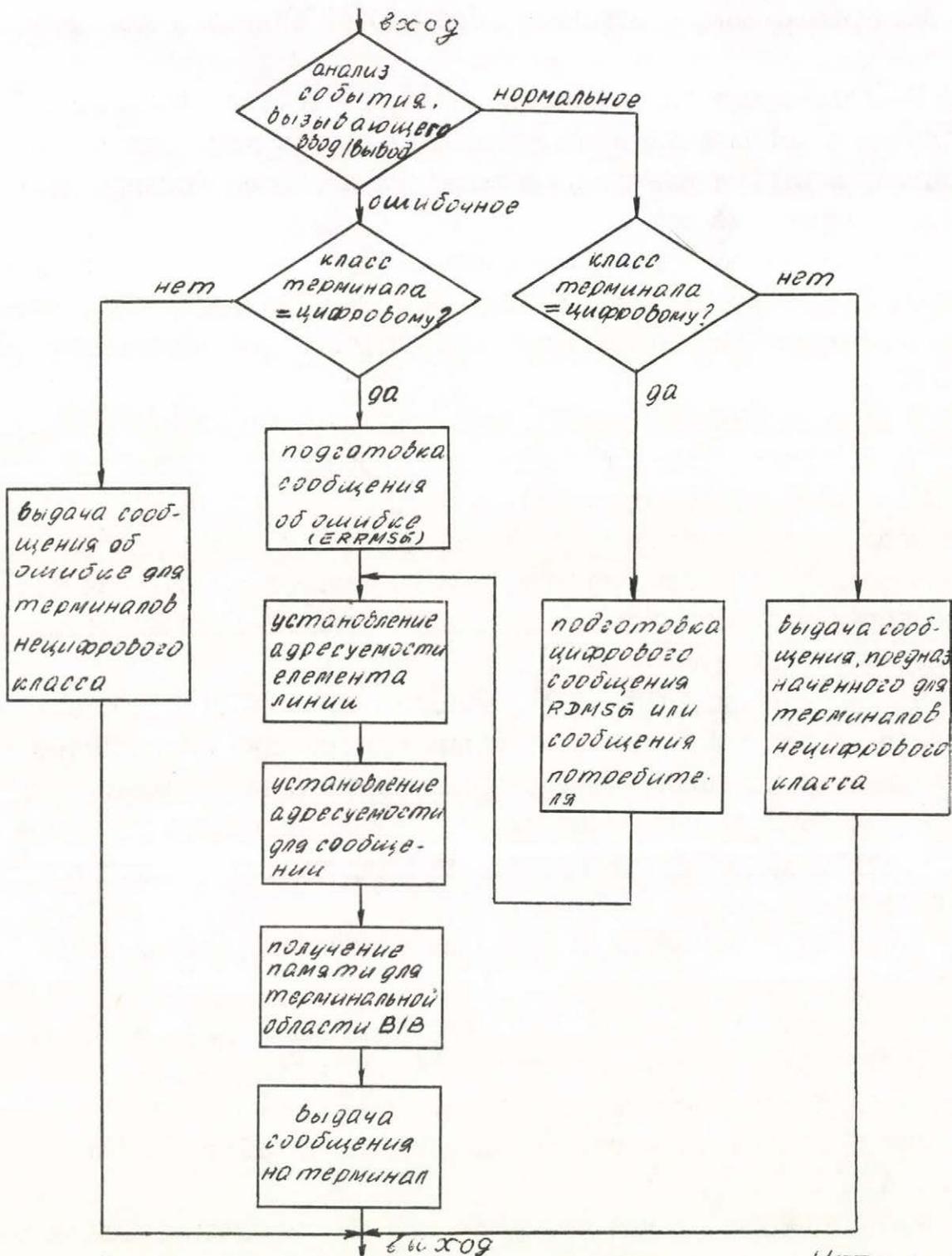


Рис 2 Дополнение логики программы УПТ

В этом модуле сделаны изменения, которые учитывают наличие входного буфера и ограниченного набора символов для ИЗОТ 8500.

С целью повышения контроля данных в системе произведены изменения и дополнения в обслуживающих программах: Подключение к системе и Отключение от системы. Обслуживающая Таблица подключения в системе сохраняет данные о секретности и приоритете оператора терминала. Доступ к системе осуществляется после успешного исполнения транзакции для цифрового подключения в системе. Предусмотрена возможность для цифрового отключения от системы.

В системе СУИП поддерживается функция Главный терминал, которая позволяет изменять параметры системы, состояние работы одного или группы терминалов и т.д.

Ограниченные возможности ведения диалога с терминалом ИЗОТ 8500 делают бессмысленным использование этой функции с этого терминала. Поэтому использование функции Главный терминал не обеспечивается.

Чтобы проверить работу терминала в системе СУИП и чтобы продемонстрировать некоторые конкретные применения, разработаны прикладные транзакции. Эти транзакции предназначены для:

- передачи данных между цифровыми терминалами;
- передачи данных между цифровым терминалом и другими терминалами;
- записи, обновления и добавления данных в файлах с цифрового терминала.

Чтобы продемонстрировать перечисленные применения, в библиотеки СУИП каталогированы следующие таблицы:

- Таблица управления терминалами (ТСТ) - содержащая информацию о терминальной сети (включительно и информацию о терминале ИЗОТ 8500);
- Таблица управления программами (РСТ) - описывающая всех используемых программ;

- Таблица обрабатываемых программ (РРТ) - содержащая информацию о каждой транзакции в системе;

- Таблица управления файлами (FCT) - содержащая описание файлов системы;

- Таблица управления назначениями (DCT) - описывающая внутренние и внешние назначения в системе СУИП.

Включением терминала ИЗОТ 8500 в системы коллективного пользования, разработанные на базе системы СУИП, расширяется сфера применения системы ЕСТЕЛ. Его маленький объём, настольная конструкция, простота обслуживания, возможность многоточечной связи делают его очень удобным для применения в материальном производстве, сельском хозяйстве и т.д., где нужно создавать крупные наборы данных.

Литература:

1. Владиков И., Радулов Н. "Программное обеспечение телеграфных аппаратов в системах телеобработки. II. СУИП", IV-ая международная конференция "Передача данных 1979г.", книга I, г.Карлови вари, ЧССР.

2. Железов Ж., Новачков А., Баев Б., Владиков И. "Об одном подходе при проектировании системы в реальном времени, независимой от вида терминалов", III симпозиум по телеобработке данных, 1976, г.Варна.

ОБОБЩЕННЫЕ ПРОЦЕДУРЫ ЛОГИЧЕСКОГО ПРОЕКТИРОВАНИЯ
БАЗ ДАННЫХ И УТОЧНЕНИЯ ИНФОЛОГИЧЕСКОЙ
МОДЕЛИ: ФОРМАЛЬНЫЙ ПОДХОД

В.М.Савинков, О.М.Вейнеров, М.С.Казаров, А.А.Александров

В I приведена общая концепция проектирования баз данных (БД) и методы формализации процесса автоматизированного проектирования. В данной статье конкретизируется формальный аппарат, который может быть положен в основу обобщенных (не зависящих от целевой модели данных) процедур одного из наиболее важных этапов этого процесса – этапа логического проектирования структуры БД. Кроме этого рассмотрен синтез инфологической модели (ИЛМ) в контексте его "обратной связи" с синтезом логических структур. Такая связь выделяется по следующим причинам:

– процедуры логического проектирования, обрабатывая данные, специфицированные в ИЛМ, могут обнаружить неполноту и (или) противоречивость этих данных;

– обработка процедурами логического проектирования специфицированной неполно ИЛМ позволяет существенно ограничить множество возможных вариантов ее полной спецификации. Привлечение полученных ограничений упрощает процесс синтеза ИЛМ.

I. Критерии качества логического проекта.

Для формализации процесса проектирования должны быть сформулированы и систематизированы критерии качества проекта, в соответствии с которыми будет оцениваться корректность и эффективность процедур проектирования.

а) Критерии первой группы определяют способность логической структуры автоматически контролировать и поддерживать соответ-

вие данных общим свойствам, специфицированным в ИЛМ. Применение этого критерия для существующих моделей данных обеспечивает подавление избыточности (контролируемую избыточность) данных, а также аномалий актуализации.

Поясним понятие избыточности, рассматривая два вида - избыточность на уровне отношений и атрибутов. Избыточность на уровне отношения (избыточность первого рода) имеет место, когда сочетание значений некоторых из атрибутов отношения не зависит от контекста. Поэтому модификация этого сочетания в одном из кортежей отношения приводит к необходимости модификации всех других включающих это сочетание кортежей.

Напротив, значение одиночного атрибута, как правило, жестко определяется контекстом. Поэтому модификация значения атрибута в одном из кортежей не влечет необходимости модификации значения атрибута в других кортежах (если, конечно, подавлена избыточность первого рода или это значение не представляет собой кортеж унарного отношения). В этом случае избыточность кодирования значений атрибута можно подавлять с помощью установления адресных указателей на единственный экземпляр.

б) Критерии второй группы отражают влияние логической организации на временные характеристики доступа. Важную роль здесь играют структурные единицы, являющиеся единицей обмена между СУБД и рабочей областью прикладной программы.

Кроме того, схема БД должна быть ориентирована на наиболее часто использующие ее приложения. В этом случае в схеме БД допускается дублирование отдельных компонентов или установление дополнительных путей доступа. Такая избыточность явно выражена в схеме, в отличие от "скрытой" избыточности первого рода.

в). Критерии третьей группы поддерживают требование интерпретировать запросы, реализуемые на БД, минимальным числом операторов манипулирования данными.

2. Синтез отношений нормальной формы как обобщенная процедура проектирования логической структуры.

Основными этапами логического проектирования являются:

- этап синтеза системы отношений нормальной формы (НФ-отношения);
- этап отображения отношений на фрагменты структуры БД применительно к СУБД, для которой ведется проектирование;
- этап объединения фрагментов в структуру БД и выбор эффективного варианта структуры.

Синтез НФ-отношений как универсальная процедура логического проектирования реляционных БД [2] применима для СУБД, поддерживающих и другие модели. Обоснуем это утверждение.

Как известно, система НФ-отношений определяет такую структуру реляционной БД, которая при определенных условиях обеспечивает избыточное представление данных и автоматическое поддержание их достоверности. Например, система ЗНФ-отношений может автоматически поддерживать свойства данных, выражаемые структурой функциональных зависимостей (ФЗ). Приведение к ЗНФ выполняется в общих чертах следующим образом. Реляционная БД представляется совокупностью функциональных отношений, каждое из которых соответствует одной из ФЗ, принадлежащих минимальному покрытию структуры ФЗ или совокупности ФЗ с одинаковой левой частью. При выполнении любой операции актуализации СУБД контролирует "функциональность" каждого отношения, что соответствует автома-

тическому контролю и поддержанию свойств, формально выражаемых структурой ФЗ в целом. Функциональное отношение, соответствующее ФЗ, во многих случаях есть та наименьшая единица синтаксического разложения БД, при котором восстановление отношения, соответствующего БД в целом, можно реализовать путем формального выполнения операций естественного соединения.

Представление функциональных отношений в реляционной СУБД очевидно. Так, реляционная модель данных задает и поддерживает отношение между ключем и неключевыми атрибутами. Однако такие роли, как ключ и неключевой атрибут, свойственны и другим моделям данных. Структуры типа набора в сетевой модели (КОДАСИЛ) также могут являться представлением функциональных отношений.

Следовательно, все достоинства представления данных, которые обеспечиваются при приведении системы отношений реляционной БД к ЗНФ, могут обеспечиваться и для БД, построенных на основе других моделей данных.

Аналогичные утверждения могут быть сформулированы и доказаны применительно к другим формам нормального представления, в том числе 4НФ, которая базируется на понятии многозначной зависимости (МЗ) и которая обеспечивает дальнейшее (по сравнению с ЗНФ) подавление избыточности.

Рассмотрим схему получения системы 4НФ-отношений.

К особенностям декомпозиционного алгоритма приведения отношений к 4НФ [3] можно отнести то, что он не предполагает никаких различий между обработкой ФЗ и МЗ, так как ФЗ является частным случаем МЗ. Такой алгоритм, во-первых, не учитывает специфических свойств ФЗ, которые не присущи в общем случае МЗ. Следовательно,

эта специфика не будет учтена в схеме БД и не будет поддерживаться целевой СУБД. Кроме того, алгоритм не учитывает тот факт, что существующие модели данных имеют достаточно развитые, хотя и ограниченные возможности описания структуры ФЗ в схеме БД. В то же время для выражения структуры МЗ эти возможности недостаточны.

Предлагаемая в работе схема получения 4НФ-отношений предусматривает выполнение нормализации в два этапа. На первом этапе на основе структуры ФЗ синтезируются отношения 3НФ. Схема синтеза, в основном, соответствует [4] и использует явное представление структуры ФЗ схемами системы синтезируемых отношений. На втором этапе на основе МЗ производится приведение 3НФ-отношений к 4НФ.

Приведение отношений к 4НФ в два этапа не только свободно от указанных выше недостатков одноэтапного алгоритма, но и эффективно в реализации. Это связано с тем, что число типов отношений в 3НФ, которые необходимо рассматривать как объекты дальнейшей нормализации, а также множество вариантов задания МЗ на атрибутах схемы таких отношений весьма ограничено. Докажем это, предварительно приведя систему аксиом МЗ и смешанную аксиоматику ФЗ-МЗ [5].

Система аксиом МЗ. Пусть X, Y, Z - подмножества множества атрибутов A отношения $R(A)$ и в общем случае эти подмножества могут быть пересекающимися:

МЗ.0 (дополнение): Пусть $XUYUZ = A$ и $Y \cap Z \subseteq X$
тогда $X \rightarrow \rightarrow Y \Leftrightarrow X \rightarrow \rightarrow Z$

МЗ.1 (рефлексивность): Если $Y \subseteq X$, тогда $X \rightarrow \rightarrow Y$.

МЗ.2 (присоединение): Если $Z \subseteq W$ и $X \rightarrow \rightarrow Y$, то $XW \rightarrow \rightarrow ZY$ (XW следует трактовать всюду как упрощенную

запись объединения XUW).

МЗ.3 (транзитивность): Если: $X \rightarrow \rightarrow Z$, $Y \rightarrow \rightarrow Z$, то $X \rightarrow \rightarrow Z \setminus Y$.

МЗ.4 (псевдотранзитивность): Если $X \rightarrow \rightarrow Y$, $YW \rightarrow \rightarrow Z$, то $XW \rightarrow \rightarrow Z \setminus YW$.

МЗ.5 (объединение): Если $X \rightarrow \rightarrow Y$ и $X \rightarrow \rightarrow Y_2$, то $X \rightarrow \rightarrow Y_1 Y_2$.

МЗ.6 (декомпозиция) Если $X \rightarrow \rightarrow Y_1$ и $X \rightarrow \rightarrow Y_2$, то $X \rightarrow \rightarrow Y_1 \cap Y_2$, $X \rightarrow \rightarrow Y_1 \setminus Y_2$ и $X \rightarrow \rightarrow Y_2 \setminus Y_1$.

Система аксиом ФЗ-МЗ:

ФЗ-МЗ.1. Если $X \rightarrow Y$, то $X \rightarrow \rightarrow Y$

ФЗ-МЗ.2. Если $X \rightarrow \rightarrow Z$ и $Y \rightarrow Z' \setminus (Z' \subseteq Z)$, где Y и Z разбьединены, то $X \rightarrow Z'$.

ФЗ-МЗ.3. Если $X \rightarrow \rightarrow Y$ и $XY \rightarrow Z$, то $X \rightarrow Z \setminus Y$.

Теорема I. Пусть $S = \{R_i\}$ - множество отношений ЗНФ, полученное в результате реализации первого этапа процедуры синтеза. Выделим на множестве S три непересекающихся подмножества S_1, S_2 и S_3 , обладающие следующими свойствами:

S_1 содержит такие R_i , на атрибутах которых определены нетривиальные ФЗ, а ключ находится в нетривиальной ФЗ от ключа системы отношений.

S_2 аналогично S_1 , однако ключи R_i находятся в тривиальной ФЗ от ключа системы отношений. Напомним, что ФЗ вида $X \rightarrow Y$ называется тривиальной, если $Y \subseteq X$;

S_3 содержит такие R_i , на атрибутах которых определены только тривиальные ФЗ.

Отношения, входящие в S_1, S_2 и S_3 будем называть соответственно отношениями S_1, S_2 и S_3 .

Покажем, что следующие типы отношений находятся в 4НФ:

- отношения типа S_1 ;
- отношения типа S_2' , если на атрибутах отношения не задана такая структура МЗ, что все первичные атрибуты входят в левую часть МЗ;
- отношения типа S_2'' такие, что все атрибуты входят в тот или иной возможный ключ.

Введем следующие обозначения: $A_i = \{A_k^*\}$ - множество атрибутов i -го отношения; P - множество первичных атрибутов; Q - множество непервичных атрибутов; L - левая часть МЗ, заданной на A_i ; M - правая часть МЗ, заданной на A_i ; K - ключ системы отношений, входящих в S .

Для S_1 разобьем множество вариантов задания МЗ на два подмножества, характеризующиеся следующими свойствами:

$$\begin{aligned} P \subseteq L \\ P \not\subseteq L \end{aligned}$$

Для первого подмножества отношение по определению находится в 4НФ: так как левая часть МЗ включает в себя ключ отношения, то

$$\forall A_k^* \in A_i (L \rightarrow A_k^*).$$

Переходя к рассмотрению второго подмножества, укажем, что наряду с заданной МЗ вида $L \twoheadrightarrow M$, используя аксиому МЗ.0, можно вывести МЗ_I вида $L \twoheadrightarrow M_1$, где $M_1 = A_k \setminus (L \cup M)$.

Очевидно, что

$$\forall A_k^* \in A_i (A_k^* \in M \vee A_k^* \in M_1 \Rightarrow K \rightarrow A_k^* \vee P \rightarrow A_k^*).$$

Тогда, используя ФЗ-МЗ.2, получим $L \rightarrow M, L \rightarrow M_1$, т.е.

МЗ фактически является ФЗ. Анализировать корректность выявленной таким путем структуры ФЗ не имеет смысла: рассматриваемое

отношение находится в ЗНФ по определению. Если выявленная структура не соответствует этому определению, это означает только то, что введенная нами структура МЗ при заданных исходных условиях не может существовать.

Для S'_2 разобьем множество вариантов задания МЗ на три подмножества, характеризующихся следующими свойствами:

$$\begin{aligned} P &\subseteq L; \\ Q &\subseteq L, P \not\subseteq L; \\ Q &\subseteq L; \end{aligned}$$

Ход рассуждений и выводы для первого подмножества соответствуют ходу рассуждений и выводам первого подмножества для S_1 .

Для второго подмножества

$$\exists Q_i \subset Q (Q \subset M \vee Q_i \subset M_1).$$

Здесь через M_1 , как и ранее, обозначено множество атрибутов правой части МЗ, выводимой с помощью правила МЗ.0. Так как $P \rightarrow Q_i$, то в соответствии с ФЗ - МЗ.2 имеет место $L \rightarrow Q_i$ и в отношении существует транзитивная ФЗ вида $P \rightarrow L \rightarrow Q_i$, что противоречит определению ЗНФ.

Для третьего подмножества все $A_k^* \in (M \cup M_1)$ находятся в тривиальной функциональной зависимости от ключа R .

Следовательно, правило ФЗ-МЗ.2 в данном случае неприменимо и на атрибутах отношения может быть определена такая структура МЗ, что отношение не находится в 4НФ.

Для S''_2 , как бы не была определена МЗ, атрибуты, входящие в ее правую часть, входят в правую часть ФЗ, левой частью которой является тот или иной возможный ключ. Таким образом, в соответствии с ФЗ-МЗ.3, МЗ в этом случае фактически является ФЗ.

В отношениях типа S_3 выделим два подтипа отношений:

- отношения, подмножества атрибутов которых функционально определяются ключом другого отношения;

- отношения, в которых такие подмножества отсутствуют.

Следующее утверждение определяет ограничения, которым удовлетворяет подструктура, заданная на атрибутах отношения первого подтипа.

Утверждение I. Пусть $R(A)$ - отношение первого подтипа; $A' \subset A$ и $D \rightarrow A'$, где D - ключ некоторого другого отношения.

Тогда на A может быть задана только такая подструктура МЗ, что A' находится в левой части МЗ.

Если это не так, и имеет место МЗ вида $A'' \rightarrow A'$, где $A'' \subset A$, то отсюда и из $D \rightarrow A'$ в соответствии с ФЗ-МЗ.2 следует $A'' \rightarrow A'$, и в отношении существует МЗ, что противоречит исходной посылке.

Это утверждение показывает, что сложные подструктуры МЗ, обуславливающие многовариантность декомпозиции, по-видимому, возможны только для отношений второго подтипа. Но эти отношения, очевидно, представляют собой уникальный ключ совокупности всех отношений, образующих ВД.

3. Взаимосвязь синтеза отношений 4НФ и синтеза инфологической структуры

Рассмотрим определения сложной структуры ФЗ и МЗ. Ее задание требует просмотра большого числа сочетаний различных атрибутов. Особые трудности сопряжены с заданием структуры МЗ, что связано со свойством МЗ, которое принято называть зависимостью МЗ от контекста. Заметим, что свойство контекстной

зависимости МЗ не означает, что ФЗ не зависит от общего контекста, то есть от смысла, выражаемого отношением. В самом деле, наличие или отсутствие ФЗ между атрибутами с одними и теми же именами, а также семантика этой ФЗ непосредственно определяются отношением, в схему которого входят эти атрибуты. Однако, как бы ни проецировалось конкретное отношение, подструктура ФЗ, определенная на атрибутах проекции, полностью совпадает с подструктурой, определенной на соответствующих атрибутах проецируемого отношения.

Иное дело со структурой МЗ. В общем случае множество атрибутов правой части МЗ, определенной на атрибутах проекции, является лишь подмножеством атрибутов правой части МЗ, определенной на атрибутах проецируемого отношения (ситуация так называемых "встроенных" МЗ). Поэтому, в общем случае, при определении МЗ следует в отличие от определения ФЗ рассматривать все отношение, а не его проекции. Следует подчеркнуть также, что в отличие от случая, когда определяется ФЗ, при определении МЗ приходится рассматривать не только атрибуты, на которых собственно задается предполагаемая МЗ, но и прочие атрибуты. Критерием правильности задания МЗ, как и принципом ее выделения, является наличие полной комбинаторики конкатенаций значений правой части МЗ с конкатенацией значений других атрибутов при фиксированном значении левой части.

Сложности, отмеченные выше, затрудняют определения структуры МЗ. Эта проблема будет решена, если ограничить множество атрибутов, которое необходимо рассматривать при определении структуры МЗ. Покажем, что разбиение множества атрибутов на подмножества, соответствующие ЗНФ отношениям, позволяют естественным образом получить такие ограничения.

Как и при доказательстве теоремы I, мы будем основывать наши рассуждения на сопоставлении структур Φ_3 и M_3 . При этом будут рассмотрены два случая, соответствующие различным гипотезам о полноте структуры Φ_3 . Первая, наиболее распространенная гипотеза, постулирует полноту Φ_3 . Это означает, что любая Φ_3 , выводимая с помощью смешанной аксиоматики Φ_3 - M_3 , и которая не может быть выведена из структуры Φ_3 , рассматривается как недостоверная. В свою очередь это влечет за собой определение M_3 , приведшей к выводу такой Φ_3 , как недостоверной M_3 [6]. Вторая гипотеза не предполагает априорной полноты структуры Φ_3 .

Утверждение 2. Если первая гипотеза справедлива, то атрибуты схем отношений $3НФ$, принадлежащих к типам, для которых на основе теоремы I устанавливается их принадлежность к $4НФ$, составляют такое подмножество, что на нем не может быть задана достоверная структура M_3 .

Доказательство этого утверждения непосредственно вытекает из доказательства теоремы I, когда было показано, что задание M_3 на атрибутах отношений $3НФ$, принадлежащих к указанному типу, приводит к выводу таких Φ_3 , что отношение утрачивает свою принадлежность к $3НФ$. Такие Φ_3 очевидно не могут быть выведены из первоначально заданной структуры Φ_3 , так как в противном случае их наличие было бы учтено при синтезе $3НФ$ -отношений. Отсюда непосредственно следует вывод о недостоверности M_3 .

Утверждение 3. Если справедлива вторая гипотеза, то атрибуты схем отношений $3НФ$, принадлежащие к типу S_1 и подтипу S_2 (который характеризуется вхождением всех атрибутов в тот или иной возмож-

ный ключ), составляют такое множество, что на нем не может быть задана достоверная структура МЗ, не являющихся в то же время ФЗ.

Так же как в предыдущем случае, доказательство утверждения вытекает из доказательства теоремы I, где показано, что МЗ, задаваемая на атрибутах указанных типов отношений фактически оказываются ФЗ. Так как было положено, что вторая гипотеза справедлива, то эти ФЗ уже не отвергаются как недостоверные.

Утверждения 2 и 3 позволяют при спецификации структуры МЗ в ИЛМ исключить ряд подмножеств из рассмотрения. Это должно быть сделано на этапе, когда для подмножеств определяются "внутренние" подструктуры МЗ.

Нам остается рассмотреть случай, когда МЗ задается на атрибутах схем различных отношений ЗНФ, и показать, что и в этом случае обеспечивается ограниченность вариантов задания структуры МЗ. Заметим, что при доказательстве теоремы I мы использовали только такие структуры МЗ, наличие которых могло привести к необходимости дальнейшего разложения ЗНФ-отношений. Поэтому случай МЗ, связывающий атрибуты различных отношений, не подвергался анализу. Теперь мы рассмотрим этот случай в связи с проблемой выявления полной структуры МЗ. Соответствие данных этой структуре так или иначе должно контролироваться при ведении БД.

Теорема 2. На атрибутах, принадлежащих схемам отношений ЗНФ, можно задать только такие МЗ, правые части которых состоят только из таких атрибутов, которые входят в ключ отношения типа S_2' или любых атрибутов отношения типа S_3' , которые не находятся в функциональной зависимости от ключа какого-либо другого отношения.

Доказательство для гипотезы I. Пусть на множестве атрибутов, принадлежащих схемам различных отношений, задана МЗ: $L \rightarrow M_1 M_2$,

где M_2 не удовлетворяет утверждению теоремы. Тогда очевидно, что существует некоторая ФЗ: $B \rightarrow M_2$, такая, что $B \cap M_2 = \emptyset$.

Тогда в соответствии с ФЗ-МЗ.2, $L \rightarrow M_2$. В силу предположения о полноте структуры ФЗ, если ФЗ существует в заданной структуре ФЗ (выводится из нее), то в соответствии с ФЗ-МЗ.1 и МЗ.6 имеем $L \rightarrow M_1$, где M_1 удовлетворяет утверждению. Если же ФЗ: $L \rightarrow M_2$ не выводится из заданной структуры, то МЗ: $L \rightarrow M_1 M_2$ не достоверна.

Доказательство для гипотезы 2 совпадает с только что приведенным, за исключением того, что в соответствии с гипотезой 2, выведенная ФЗ: $L \rightarrow M_2$ считается достоверной.

Положения, сформулированные и доказанные в теоремах 1 и 2, а также в утверждениях 1 - 3 позволяют определить рациональную схему реализации интегрированного процесса выполнения обобщенных процедур логического проектирования и уточнения инфологической модели (рис.1).

Приведенные результаты позволяют применить идеи нормализации при проектировании БД, отличных от реляционных, доказывают эффективность последовательного (от ЗНФ к 4НФ) преобразования нормального представления и определяют принципы "обратной" связи обобщенных процедур логического проектирования и уточнения информационно-логической модели. На этой основе определяется схема реализации взаимосвязанных автоматизированных процессов спецификации подмодели ИЛМ и выполнения обобщенных процедур логического проектирования БД.

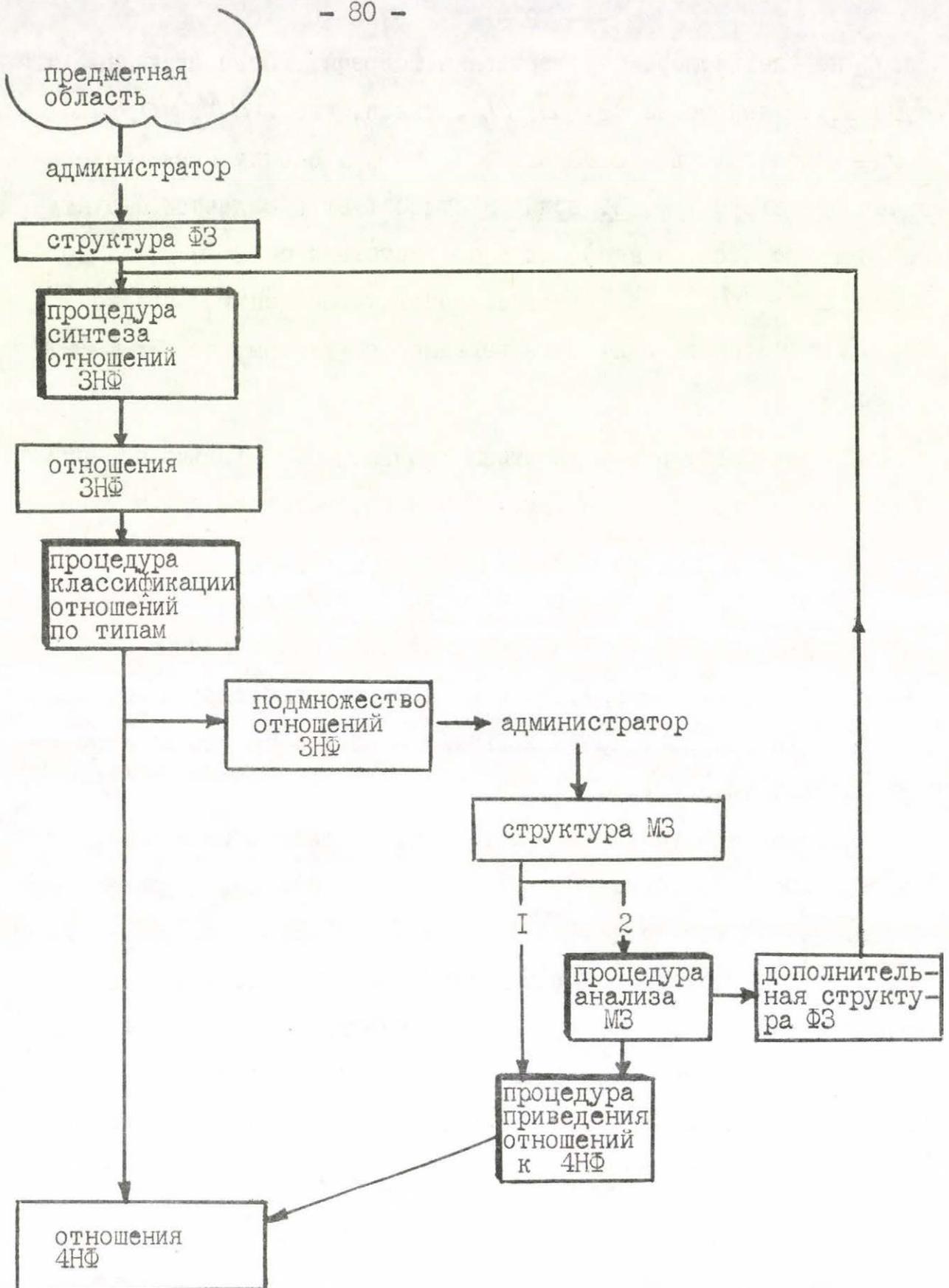


Рис. 1. Процедуры логического проектирования БД.

1 - гипотеза 1: задана полная структура ФЗ.

2 - гипотеза 2: задана неполная структура ФЗ.

ЛИТЕРАТУРА:

- 1 Савинков В.М., Вейнеров О.М., Казаров М.С. Основные концепции автоматизации проектирования баз данных - Прикладная информатика, М., Статистика, 1982, вып. I, с. 30-41.
- 2 Atre s. Data base: structured techniques for design, performance, and management.- Jhn Wiley Sons, Inc., 1980.
- 3 Beeri C., Bernstein P.A., Goodman N.A. Sophisticated introduction to data base normalization theory.- In: Proc. 4th Intern. Conf. Very Large Data Bases, West Berlin, 1978, p. 113-124
- 4 Bernstein P.A. Synthesizing third normal form relations from functional dependencies.- In: ACM Tran. on Database Systems, V. I, No 4, 1976, p. 277-298
- 5 Beeri C., Fagin R.A. A complete axiomatization for functional and multivalued dependencies in data relations.- In: ACM-SIGMOD Conf. Proc., Toronto, 1977, p. 47-61.
- 6 Luk W.S. Possible membership of multivalued dependency in relation database.- Information Processing Letter, 17 August 1979, V. 2, No 2.

О ПОСТРОЕНИИ МАТЕМАТИЧЕСКОГО АППАРАТА ДЛЯ ОПИСАНИЯ ПРОЦЕССОВ ПРОЕКТИРОВАНИЯ И ФУНК- ЦИОНИРОВАНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ

Стогний А.А., Кондратьев А.И.

институт кибернетики АН УССР

В настоящее время в теории АИС банков данных пока не создан математический аппарат, который отвечал бы следующим требованиям: - в нем описывались бы процессы проектирования и функционирования;

- была бы возможна классификация, включающая существующее разделение АИС на классы: информационно-поисковые, информационно-справочные, информационно-логические, информационно-распознающие;

- вводился бы единый критерий оптимальности для оценки качества проекта; качества функционирования, характеристики предметной области, поведение пользователей и др. аспекты.

Необходимость и актуальность создания такого аппарата вытекает из следующих соображений:

1) процесс проектирования АИС в настоящее время является скорее искусством, чем наукой;

2) необходимо выделить научный, прикладной и инженерный аспекты и теории АИС и определить перспективы развития каждого из этих аспектов;

3) необходимо разработать эффективный инструментарий для автоматизации процесса проектирования АИС.

Все объекты и процессы, рассматриваемые в АИС, являются либо конфликтными (между проектировщиками АИС и будущими пользователями, между проектировщиками при ограниченных выделенных ресурсах и между пользователями при ограниченных вычислительных ресурсах), либо содержат наличие неопределенности в процессе принятия решений (на этапе информационного обследования /1/ при проектировании, при определении информационных потребностей при функционировании АИС). Все это дает возможность использовать теорию игр при создании единого математического аппарата

АИС. Этот математический аппарат предложен авторами в /1,2/ и назван нормативной теорией игр. В изучении конфликта выделено три последовательных этапа: дескриптивный, на котором конфликт описывается; конструктивный, где задаются связи между отдельными компонентами конфликта и нормативный, на котором решаются три основные задачи теории игр и находится численное значение оптимальной реализации той или иной игры. В статье последовательно рассмотрены все три этапа изучения конфликта, описывающего объекты и процессы АИС, при этом используются результаты /1-4/. Необходимость одновременного рассмотрения процессов проектирования и функционирования связана с тем, что АИС и банк данных являются системными объектами и оценить качества проекта можно только при функционировании банка данных. Итак, процесс функционирования банка данных дает необходимые сведения как для оценки собственно функционирования, так и для оценки качества проекта.

1. Описание процессов проектирования и функционирования банков данных

Прежде чем перейти к дальнейшему изложению, приведем целый ряд известных сведений. Во-первых - это технологический цикл обработки информации на ЭВМ, содержащий информативный и директивный контура /3/ и состоящий из этапов:

- подготовка данных;
- ввод;
- предварительная обработка;
- хранение;
- ведение;
- содержательная обработка;
- постобработка и редактирование;
- вывод;
- использование программ и данных.

Функционирование АИС на самом общем уровне описывается так /10/:

- 1) выражение пользователем своей информационной потребности;
- 2) формулировка запроса;
- 3) собственно функционирование АИС, состоящее из этапов технологического цикла;

- 4) оценка качества выдаваемой информации, уточнение информационной потребности и переход к пункту 2.

Процесс проектирования банка данных – это последовательность выполнения этапов определенных работ по созданию АИС. Эти этапы и их подробное содержание изложено в /10/. В /10/ показано, что процесс проектирования АИС сводится к проектированию информационного и директивного контуров. В /10/ рассмотрено подробное содержание этапов информативного и директивного контуров, основные задачи, возникающие на этапах этих контуров и их решение.

Процесс функционирования АИС, как указано в /10/ также тесно связан с информативным и директивным контуром.

Взаимосвязь банка данных с пользователем заключается в решении задач содержательной обработки, в соответствии с которыми АИС разделяются на следующие классы /10/:

- информационно-поисковые системы (в них содержательная обработка отсутствует);
- информационно-справочные системы (в них содержательная обработка сводится к вычислению значений арифметических функций);
- информационно-логические системы (в них содержательная обработка сводится к вычислению значений логических функций);
- информационно-распознающие системы (в них рассматривается решение распознавания или классификации, обработка размытой неточно и противоречиво заданной информации; дискретное и процессионное прогнозирование; дискретная оптимизация);
- информационные системы с последующим принятием решений (формирование понятий и образов; обработка информации эвристическими алгоритмами);

- банки знаний с произвольной содержательной обработкой.

Итак, разрабатываемый математический аппарат должен содержать и описывать на формальном уровне все перечисленные выше задачи, процессы, этапы и компоненты.

+

2. Основные этапы формирования математического аппарата

Анализ существующего состояния разработок в области банков данных будет проводиться таким образом:

- 1) системное обследование сфер применения. Анализ и взаимосвязь характеристик СУБД и ИИС со сферами применения;
- 2) детализация и подробное описание этапов, компонент, средств и технологии проектирования АИС для накопления и анализа опыта. Анализ проектирования базы данных и программного обеспечения. Анализ опыта разработки;
- 3) структуризация и анализ функционирования действующих АИС. Анализ функционирования программного обеспечения;
- 4) переход к строгому описанию и формализации по мере накопления данных обследования и построение на этой основе математических моделей конкретных АИС, а затем и классов АИС;
- 5) разработка средств автоматизации проектирования, оптимизации функционирования и других и создание на этой основе математической теории АИС и БИД.

Вкратце опишем, на каком уровне находится математический аппарат для описания процессов проектирования и функционирования банков данных. Бурный прогресс в этой области привел к созданию большого числа как практических систем, так и целого ряда теоретических результатов. Говорить же о едином математическом аппарате в АИС пока еще рано. Выделенные в ней проблемы, которые мы изложим ниже, являются результатом некоторой классификации, а не результатом более общей теории, содержащей основные

выделенные проблемы в качестве внутренних. Насущные требования возникновения такого аппарата, содержащего понятия, методы и средства вытекают из того, что он позволит, во-первых, четко выделить собственно внутреннюю проблематику; во-вторых, определить взаимосвязи с другими областями знания, в частности, с аппаратом математики, определив тем самым внешние проблемы этого аппарата; наконец, в-третьих, выяснить обратную взаимосвязь, т.е. какие задачи в этом аппарате индуцируют те или другие проблемы современной математики. Естественно выделить следующие три этапа формирования этого аппарата:

1°. Предложено большое количество разрозненных методов и реальных систем для решения актуальных практических задач. На этом этапе удалось определить основное содержание и проблемы АИС. В целом вся проблематика АИС, за исключением аппарата ИРС находится на этом этапе. Следующие два этапа характеризуют только математическую теорию ИРС.

2°. Оказалось возможным единые математические формализмы для средств описания и средств решения в основном только для проблем математической теории ИРС. При этом вместо отдельных методов изучаются модели таких методов. Универсальная модель на этом этапе не была создана.

3°. На этом этапе удалось выяснить строение всей совокупности методов решения для задач математической теории ИРС.

Дадим в заключение этого раздела перечень основных проблем рассматриваемого математического аппарата с указанием соответствующих этапов технологического цикла обработки информации.

- 1) Формулировка и решение сложных задач.
- 2) Выделение объекта или процесса.
- 3) Наблюдение за объектом. Планирование эксперимента.

Домашинная
подготовка
информации.

- | | |
|--|--|
| 4) Создание БНД наблюдений на ЭВМ. | Ввод, предвари-
тельная обработ-
ка, хранение,
ведение. |
| 5) Формирование понятий, образов. | |
| 6) Создание информационной модели
объекта или процесса | Содержательная |
| 7) Создание динамической информа-
ционной модели объекта или
процесса. | обработка |
| 8) Распознавание, классификация
образов. | |
| 9) Выделение характеристик объекта. | |
| 10) Создание математической модели
объекта. | |
| 11) Введение принципа оптимальности.
Задача анализа. | Содержательная |
| 12) Реализуемость принципа оптималь-
ности. Задача синтеза. | обработка
информации |
| 13) Поиск оптимального решения.
Задача управления. | |
| 14) Указание оптимального решения
в терминах объекта или процесса. | |
| 15) Анализ решений сложной задачи. | Постобработка,
редактирование,
вывод, исполь-
зование получен-
ной информации. |

3. Основные результаты

процессы проектирования и функционирования банков данных сведены к взаимодействию множества описаний, т.е. множества всевозможных способов представления обрабатываемой информации и множества алгоритмов, преобразующих информацию заданным образом.

Два этих множества являются формализацией и обобщением информативного и директивного контуров, введенных в АИС, и они погружены в общую схему, которая состоит из этапов, связанных между собой последовательно. Каждый из этапов определен множеством и структурой описаний на входе и на выходе, а также множеством алгоритмов, осуществляющих переход от входного описания к выходному на основе накопленной информации. Для представления обрабатываемой информации введен математический объект, названный допустимым понятием, связываемый с описанием в виде вектора, компоненты которого суть элементы или подмножества метрических или полуметрических пространств. Вводятся различия между понятиями по их описаниям: во множество понятий одного уровня отнесены понятия, описания которых совпадают, как по числу компонент, так и по метрическим или полуметрическим пространствам, которым принадлежат их компоненты. Множество понятий представимо в виде дерева или сети, не обязательно конечного. Если это, например, дерево, то к его произвольной вершине приписано, не обязательно конечное, множество понятий одного уровня, объединенных по исследованным принципам и являющихся новым понятием. Стрелки, ребра дерева понятий означают алгоритмы формирования понятий, направленного перехода от одного понятия к другому. Вводятся задачи и алгоритмы связи между понятиями. Директивный контур формально записывается в виде композиции допустимой совокупности алгоритмов распознавания, а информативный контур задается в виде дерева понятий.

4. Теоретико-игровая модель процессов проектирования и функционирования банков данных

4.1. Дескриптивный этап теоретико-игрового моделирования

Разделим все объекты и процессы, относящиеся к АИС на два условных класса: внутренний и внешний. К внешнему классу отнесем

предметную область, будущих пользователей АИС, которые создают эту область, формализацию и представление этой области. К внутреннему классу отнесем разработчиков АИС и процесс проектирования. Обозначим участников внешнего класса через J_1 , а внутреннего - J_2 и будем называть их командами. Команды выполняют действия, множество возможных действий для каждой из них обозначим через S_{J_1} и S_{J_2} . В результате того, что J_1 и J_2 выполняют свои действия - стратегии f_{J_1} и f_{J_2} из S_{J_1} и S_{J_2} соответственно, складывается некоторая ситуация $f_J \in S_J$, $f_J = (f_{J_1}, f_{J_2})$ где $J = J_1 \cup J_2$, в которой выигрыш команд J_1 и J_2 определяется как $a_{J_1}(f_{J_1}, f_{J_2})$ и $b_{J_2}(f_{J_1}, f_{J_2})$. Положим

$$a_{J_1}(f_{J_1}, f_{J_2}) + b_{J_2}(f_{J_1}, f_{J_2}) = c(f_J) \quad (1)$$

В том случае, если $c(f_J)$ - это константа, то конфликт описывается бескоалиционной игрой с постоянной суммой, которая стратегически эквивалентна игре с нулевой суммой, а в данном случае - математической игре. Обозначим число стратегий во множестве

S_{J_1} через M . Конфликт между командами J_1 и J_2 в случае (1) описывается следующей матричной игрой:

$$\Gamma_1 = \langle S_{J_1}, S_{J_2}, \{a_{J_1}(f_{J_1}^m, f_{J_2}^n)\}_{m,n=1}^{M,N} \rangle \quad (2)$$

4.2. Конструктивный этап теоретико-игрового моделирования

Команды J_1 и J_2 имеют в своем составе игроков, из которых могут быть составлены коалиции. Обозначим через \mathcal{K}_1 - множество коалиций, составленных из игроков команды J_1 , а через \mathcal{K}_2 - множество коалиций из игроков команды J_2 . Коалиции и отдельные игроки из \mathcal{K}_1 и \mathcal{K}_2 выполняют действия; обозначим множество допустимых действий для коалиции \tilde{K} через $S_{\tilde{K}}$ (множество коалиционных стратегий коалиции \tilde{K}). В силу того, что коалиции из каждого множества \mathcal{K}_s , $s=1,2$ между собой пересекаются, то их коалици-

онные стратегии должны быть согласованными /5/. Пусть $f_{\alpha_s} = \{f_{\tilde{\kappa}}\}_{\tilde{\kappa} \in \alpha_s}, s=1,2$ - согласованное семейство коалиционных стратегий. Согласованное продолжение этого семейства обозначим через $f_{\mathcal{J}_s}, s=1,2$. В реальном конфликте среди возможных действий $S_{\tilde{\kappa}}$ коалиции $\tilde{\kappa}$ из определенной команды есть действия, которые могут соответствовать целям противоположной команды. Этот факт можно учесть формально следующим образом. Пусть для определенности $\tilde{\kappa}_1 \in \alpha_1$, а $\tilde{\kappa}_2 \in \alpha_2$. Если $\tilde{\kappa}_1$ выполняет действия $f_{\tilde{\kappa}_1}$, которые рассматриваются как действия, соответствующие целям коалиции $\tilde{\kappa}_2$, т.е. и целям команды, то целесообразно рассматривать коалицию $\tilde{\kappa} = \tilde{\kappa}_1 \cup \tilde{\kappa}_2$, выполняющую единое действие. Введем в этой связи множество \mathcal{K} - коалиций из \mathcal{J} , образованных из всевозможных объединений коалиций из α_1 и α_2 . В результате того, что коалиции $\tilde{\kappa}$ из $\alpha_s, s=1,2$ и коалиции κ из \mathcal{K} выбирают свои стратегии $f_{\tilde{\kappa}}$ и α_s образуется ситуация κ , которая является согласованным продолжением согласованного семейства коалиционных стратегий f_{α_s} . Пусть $H_{\tilde{\kappa}}(f_{\mathcal{J}})$ - выигрыш коалиции $\tilde{\kappa}$ в условиях ситуации $f_{\mathcal{J}}$.

Выполнено:

$$\sum_{\tilde{\kappa} \in \alpha_s} H_{\tilde{\kappa}}(f_{\mathcal{J}}^{mn}) = (-1)^{s+1} a_{y_s}(f_{\mathcal{J}}^{mn}), s=1,2 \quad (3)$$

Пусть i - игрок из \mathcal{J} и S_i - множество индивидуальных стратегий игрока $i \in \mathcal{J}$. Определим коалиционную игру Γ_2 :

$$\Gamma_2 = \langle \mathcal{J}, \mathcal{K}, \alpha_1, \alpha_2, \{S_i\}_{i \in \mathcal{J}}, \{H_{\tilde{\kappa}}\}_{\tilde{\kappa} \in \alpha_s, s=1,2} \rangle \quad (4)$$

Далее введем понятия угрозы и контругрозы игроков и коалиций и реализацию игры, используя результаты /4/.

Определение I. Коалиционной игрой с реализациями назовем совокупность /4/:

$$\Gamma\{Q\} = \langle \mathcal{J}, \{Q\}, \mathcal{K}\{Q\}, \alpha_1\{Q\}, \alpha_2\{Q\}, \{S_i\}_{i \in \mathcal{J}}, \{H_{\tilde{\kappa}}\}_{\tilde{\kappa} \in \alpha_s\{Q\}, s=1,2} \rangle,$$

где J - множество игроков; $\{Q\}$ - множество реализаций, а $Q = \langle R_1, P_1, \dots, R_2, P_2 \rangle$, а R_i, P_i - угроза и контругроза коалиций k и \tilde{k} из $\mathcal{A}_i \{Q\}$, $i=1,2$.

Теорема 1. Математической моделью описывающей процессы проектирования и функционирования АИС, является $\Gamma\{Q\}$.

4.3. Нормативный этап теоретико-игрового моделирования

Игра Γ_1 матричная и в ней принцип оптимальности записывается в виде:

$$\max_{\{f_1^m\}} \min_{\{f_2^n\}} \{a_{j_1}(f_1^m, f_2^n)\}_{m,n=1}^{M,N} =$$

$$= \min_{\{f_2^n\}} \max_{\{f_1^m\}} \{a_{j_1}(f_1^m, f_2^n)\}_{m,n=1}^{M,N} =$$

Игра Γ_2 является коалиционной игрой и в ней естественно /5/ считать принципом оптимальности стремление коалиции к Ψ -устойчивым ситуациям.

Ситуацию f_j назовем приемлемой для угрожающей коалиции \tilde{k} относительно контругрожающей коалиции k , если выполнено

$$\forall f_k^* \in S_k, f_{\tilde{k} \setminus k}^* \in S_{\tilde{k} \setminus k}$$

$$H_{\tilde{k}}(f_j \parallel f_{k \cup \tilde{k}}) \leq H_{\tilde{k}}(f_j \parallel f_k^*)$$

Приемлемость ситуации f_j задана для пары коалиций \tilde{k} и k , причем $\tilde{k} \in \mathcal{A}_s, s=1,2$, а $k \in \mathcal{A}$, $k \cup \tilde{k} \in \mathcal{A}$.

Ситуация f_j называется Ψ -устойчивой, если $\forall \tilde{k} \in \mathcal{A}_s$ она приемлема для \tilde{k} относительно $\Psi_{\tilde{k}}$.

Теорема 2. Для Ψ^* -устойчивых ситуаций в смешанном расширении игры Γ_2^* необходимо и достаточно выполняется следующее соотношение:

$$\max_{\{p_{\tilde{k}}\}_{\tilde{k} \in \mathcal{A}_1}} \min_{\{p_{k \cup \tilde{k}}\}_{k \in \mathcal{A}_2}} \sum_{\tilde{k} \in \mathcal{A}_1} H_{\tilde{k}}^*(p_j) =$$

$$= \min_{\{\mu_{\tilde{k}}\}_{\tilde{k} \in \mathcal{A}_2}} \max_{\{\mu_{\tilde{k}}\}_{\tilde{k} \in \mathcal{A}_1}} \sum_{\tilde{k} \in \mathcal{A}_2} H_{\tilde{k}}^*(\mu_{\tilde{k}})$$

при этом $H_{\tilde{k}}^*(\mu_{\tilde{k}}) = \sum_{f_j \in S_j} H_{\tilde{k}}(f_j) \cdot \mu(f_j)$,
 $\tilde{k} \in \mathcal{A}_1, \tilde{k} \in \mathcal{A}_2, k \in \mathcal{A}_1, \varphi^*: \tilde{\mathcal{A}} \rightarrow \mathcal{A}$

5. Практическое использование аппарата для выбора оптимального проектного решения

Поставим такую задачу: как на основе некоторых сведений, полученных на этапе информационного обследования и нашего опыта проектирования реальных систем, выбрать совокупность проектных решений из состава ранее известных и из состава известных с недостаточной полнотой, которая обеспечила определенное качество проекта и при этом созданная АИС по качеству функционирования находилась бы в определенных допустимых пределах.

Определим здесь ряд понятий.

1. Сведения на этапе информационного обследования [3], [4]

Это например, сведения:

1) источники информации (количество, интенсивность информационных потоков, определяемая как частота поступления данных в систему; режимы функционирования (это касается всего множества источников));

2) количественные характеристики БД (общее количество данных системы, количество типов информационных объектов или документов, количество первоначально вводимых данных);

3) пользователь (общая характеристика задач обработки данных, количество выходных документов, количество пользователей, режимы получения информации);

4) условия разработки системы (сроки, ресурсы, условия участия заказчика, техническая реализация в самом общем виде, особые условия (географические характеристики объекта, трудности

доступа, секретность информации) [3], [10].

Эти сведения после определенного кодирования [II] дают вектор, составленный из элементов метрических или полуметрических пространств. Это могут быть и другие сведения, полученные с помощью определенной анкеты.

II. Опыт проектирования реальных систем

Записывается в виде таблицы, в которой по строкам расположены реальные информационные системы, а по столбцам значения параметров, которые являются численными значениями вопросов соответствующих анкет. Для получения такой таблицы необходимо заполнить анкеты проектировщика (подобно 20 заполненным анкет по СУБД и ИИС в [9]). Далее каждая заполненная анкета будет закодирована по определенным правилам и таблица готова. Заметим, что в настоящее время и правила по ее заполнению и кодированию созданы и в скором времени она будет разослана проектировщикам.

III. Известные проектные решения

Эти решения выделены и для каждого из этапов (ИО, ТЗ, ТП, РП, ЭЗ) и для компонент АИС (БД, ТС, ПО, ОММ) достаточно хорошо известны. Они описаны в анкете проектировщика.

IV. Проектные решения, известные с недостаточной полнотой

Определяются как неполностью известные на данном этапе проектирования и с помощью некоторой функции принадлежности и проектное решение рассматривается как размытая переменная. Эта возможность и вариант ее кодирования также предусмотрены в анкете проектировщика.

V. Качество проекта АИС

Определяется некоторым числом, которое получено на основе обработки накопленного опыта, соответствующее здравому смыслу и соответствует значению функции выигрыша теоретико-игровой модели, описывающей процесс проектирования в [4].

VI. Качество функционирования АИС

Этот параметр является также числом и получается на основе обработки накопленного опыта по анкетам, описывающих опыт функционирования реальных АИС. Этот параметр является значением функции выигрыша некоторой теоретико-игровой модели, описывающей процесс функционирования в [4]. Опыт функционирования реальных АИС также представляет собой таблицу, в которой по строкам описаны реальные АИС, а по столбцам параметры, соответствующие вопросам анкеты для функционирования АИС.

Итак, разработанные авторами аппарат позволяет решать указанную задачу. Пока эта задача решается на модельных анкетах, но результаты, по мнению авторов, хорошие. В настоящее время анкеты для проектировщика и анкеты по описанию функционирования и методики по их заполнению и кодированию уже созданы, есть также алгоритмы по решению этих задач, которые в общем виде являются эвристическими, созданы алгоритмы-корректоры эвристических алгоритмов и накопленного опыта. Для полной практической реализации необходимы совместные усилия по организации системного обследования реальных АИС: как по проектированию, так и по функционированию.

Л И Т Е Р А Т У Р А

1. Кондратьев А.И., Стогний А.А. О понятиях нормативной теории игр. - Кибернетика, 1979, № 4.
2. Стогний А.А., Кондратьев А.И. О нормативной теории игр. - ДАН УССР, 1979, № 9.
3. Стогний А.А. Об основных принципах построения автоматизированных информационных систем. - УСИМ, 1972, № 2.
4. Кондратьев А.И. Коалиционные игры с реализациями. - Кибернетика, 1979, № 6.
5. Воробьев Н.Н. Коалиционные игры. - Теория вероятностей и ее применения, 1967, т. XII, № 2.

6. Журавлев В.И. Алгебраический подход к задачам распознавания. - Проблемы кибернетики, 1973, № 22.
7. Кондратьев А.И., Стогний А.А. О построении математического аппарата для описания процессов проектирования банков данных. - В кн.: I Всесоюзной конференции по банкам данных (г.Тбилиси, ноябрь, 1930 г.): Тез.докл. Секция I., М., Изд-во ВНИИПОУ, 1930.
8. Стогний А.А., Кондратьев А.И. Исследование и разработка эвристических алгоритмов в теории игр. - В кн.: Всесоюзной конференции "Методы математической логики в проблемах искусственного интеллекта и систематическое программирование" (паланга, сентябрь, 1930 г.): Тез.докл., Вильнюс: Изд-во Института математики и кибернетики АН ЛитССР, ч.П, 1930.
9. Основные характеристики отечественных СУБД и ИИС (под редакцией А.А.Стогния). Препринт -30-49. К.: ИК АН УССР, 1930, 71с.
10. Стогний А.А., Кондратьев А.И. Информационные системы в управлении (программное обеспечение). - Киев: Знание, 1930.
11. Стогний А.А., Кондратьев А.И. О стратегическом представлении знаний в ИИС. - ДАН УССР, 1930, сер. А, № 5.
12. Стогний А.А., Кондратьев А.И. Вопросы нормативной теории игр и ее приложений. - К.: ИК АН УССР, 1979, 42 с. (АН УССР Институт кибернетики. Препринт 79-52).

УПРАВЛЕНИЕ ФОРМАТА ОТЧЕТОВ И ПОИСКА ИНФОРМАЦИИ
В БАЗАХ ДАННЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ
С МНОЖЕСТВЕННЫМ ДОСТУПОМ

А.И.ТЕРЗИЕВ

ИММ-БАН, г.СОФИЯ - НРБ

Производство средств обработки информации является одной из самых быстрорастущих и наиболее динамичной научно-технической отраслью современности. По мере роста мощности вычислительных средств, объема банков данных и числа пользователей, добивающихся доступа к этим средствам, должна также расти интенсивность конкуренции пользователями за одновременный доступ к вычислительной системе. Их противоречивые требования очевидно приводят к очередям и задержкам, поэтому при коллективном пользовании ресурсов вычислительных систем, появляется необходимость в существовании средств, позволяющих пользователям уменьшить время размышления /1/. Эти средства должны быть на высоком логическом уровне, должны обеспечивать гибкое манипулирование с данными баз данных и ресурсами вычислительной системы, а также избавлять пользователя от забот, связанных с использованием ресурсов /4/.

Важный вопрос при создании СУБД - это обеспечение средств для управления формата отчетов и поиска информации. Этим занимались многие исследователей, разработывающих СУБД. Еще с появлением первых генераторов отчетов началась разработка основных концепций и требований /5/. Ввод вычислительных систем коллективного пользования, приводило до новые требования и решения. Телекоммуникационный метод доступа обеспечил новые возможности. В связи со всем этим мы разработали некоторые логические средства управления форматирования и поиска информации в базах данных вычислительных систем с множественным доступом, которые будут показаны здесь.

Термины, которые будем использовать, приняты из терминологии рабочей группы баз данных к комитету КОДАСИЛ.

Данные в базах данных можем рассматривать как совокупность необработанных фактов и цифр, накопленных вычислительными машинами, а информация — это те сведения, которые извлечены из этой массы фактов и обработаны для определенного лица, для определенной цели или для удовлетворения заданного запроса. Наименьшая семантически значимая единица поименованных данных является элементом данных (поле), а логическая запись — это поименованная совокупность элементов данных /6/. Агрегат данных и сегмент состоят из одного или нескольких элементов данных. Следовательно, мы должны иметь средства для идентифицирования тех логических записей, из которых выводятся данные, а также возможность указать какую информацию из них ищем, т.е. где в записе она накоплена. Существует ряд хорошо развитых средств для идентифицирования необходимых записей /3,7/, поэтому не будем рассматривать этих вопросов. Мы рассмотрим с одной стороны средства для идентифицирования информации в экземплярах элементов данных, а с другой — для идентифицирования требуемых элементов данных в записях. Мы увидим как объединение операции поиска и форматирования влияют на эксплуатационные качества СУБД.

Знаем что в каждой базе данных и соответствующей СУБД существует средства для логического описания данных конкретного пользователя, т.е. средства для логического описания подсхем, которые для разных пользователей могут быть разными, но они всегда подсовокупность схемы /2/. Поэтому при управлении поиска или форматировании отчетов пользователь знаком с идентификаторами (именами) доступных для него элементов данных.

Во время обработки данных потребитель может сохранять или накапливать междинные результаты, необходимы для дальнейших обработок или вывода на печать. Такие средства предоставляет ему следующий оператор:



Оператор типа М (перенести)

$$M \left\{ \begin{array}{l} \text{имя переменной} [(\text{размер})] \\ \text{имя элемента} \end{array} \right\} = \left\{ \begin{array}{l} \text{имя переменной} \\ \text{имя элемента} \\ \text{константа} \end{array} \right\}$$

Этот оператор пересылает данные из области памяти, указанной вторым операндом, в область памяти, определенную первым операндом. Количество переносимых данных определяется явной или неявной длиной области, адресуемой первым операндом.

Оператор типа С (вычислить)

$$\left\{ \begin{array}{l} \text{имя переменной} \\ \text{имя элемента} \end{array} \right\} = \left\{ \begin{array}{l} \text{имя переменной} \\ \text{имя элемента} \\ \text{константа} \end{array} \right\} \left\{ \begin{array}{l} + \\ * \\ / \end{array} \right\} \left\{ \begin{array}{l} \text{имя переменной} \\ \text{имя элемента} \\ \text{константа} \end{array} \right\}$$

Результат выполнения арифметических операций над данными, указанными в двух последних операндах, запоминается в области памяти указанной первым операндом.

Оператор типа S (выбрать, сравнить)

$$S \left\{ \begin{array}{l} \text{имя переменной} \\ \text{имя элемента} \end{array} \right\} \left\{ \begin{array}{l} \geq \\ < \\ [q] \end{array} \right\} \left\{ \begin{array}{l} \text{имя переменной} \\ \text{имя элемента} \\ \text{константа} \end{array} \right\}$$

Проверяется содержимое первого операнда выполняет ли указанную реляцию с содержимым последнего операнда. Если первый операнд указывает на данные символьного типа, то можно указать и позицию (q) или последовательность символов после которой выполняется сравнение. Можно также проверить данные первого операнда содержат ли последовательность символов, указанных вторым операндом. Если ответ сравнения положителен -

происходит переход к выполнению следующего оператора, в противном случае - через один оператор.

Оператор типа В (безусловный переход)

В «адрес оператора»

При выполнении оператора происходит переход по адресу определенному операндом. Операторы типа В и \mathcal{S} позволяют проверки сложных логических условий, а также возможность производить разветвление в "потребительской программе". Они обеспечивают поиска ищемой информации. Условия можно проверить как между данными в одной записи, так и между данными разных логических записей используя и оператор типа М.

Оператор типа С и Р (пропустить строк)

$\left. \begin{array}{l} \{C\} \\ \{P\} \end{array} \right\}$ «счет пропуска строк»

С помощью этих операторов потребитель может регулировать расстояние между строками или пропуск строк при выдаче информации на печать. При оператор типа Р новая строка начинается с новой страницы со строки, указанной операндом.

Оператор типа L (печать литерал)

L {счет пропуска } [коэффициент] 'литерал'
позиции - 1 } [кратности -p]

Этот оператор записывает р раз последовательно данные "литерал" после пропуска 1 позиции за последнего записанного в строке символа. Он позволяет при выдаче информации на печать всавлять дополнительные тексты.

Оператор типа D (печать текущей даты)

D «**Формат записи даты**»

После последнего выводимого на печать символа записывается текущая дата выполнения отчета в формате, указанном операндом.

Оператор типа I (печать данных)

I {имя переменной} [размер области] $\left\{ \begin{matrix} E \\ F \\ B \\ X \\ P \end{matrix} \right\}$ [[константа]_p],[константа]_q }
I {имя элемента} [вывода - d]

После последнего выводимого на печать символа записывается в d позиции информация из данных, указанных первом операндом. Тип данных определяется автономно СУБД используя логической подсхемы, а расположение и вид записей данных в базе данных - физической схемой (описание физического хранения данных).

Если соответствующая данна - символьного типа, то отпечатываются ее символы заключенные между позициями, указанными последними двумя операндами. Если указанная информация не помещается в области вывода, то печать продолжает в следующей строке, но в той же области вывода.

Если соответствующая данна - числового типа, то при указанном формате записи Fp.q данное представляется в виде десятичного числа, состоящего из целой и дробной частей, разделенных точкой, где p - количество всех символов, а q - количество цифр после десятичной точки.

Если соответствующая данна - числового типа, то при указанном формате записи Ep.q данное представляется в виде числа, состоящего из мантисы и порядка, где q - количество цифр мантисы (например как в ФОРТРАНе).

При остальные форматы отпечатывается битовое содер-

жание q - битового поля, содержащегося в указанной данной после p -того бита. Вывод на печать будет в двоичном, шестнадцатичном или десятичном формате в зависимости от типа формата - В, X или P.

Оператор типа I позволяет с достаточной точностью идентифицировать информацию в указанной данной, которую надо выводить на печать. Он самый существенный оператор и существует в ряде СУБД /3,7,8,9/.

Оператор типа R (печать по каталогу)

$R \left\{ \begin{array}{l} \text{имя переменной} \\ \text{имя элемента} \end{array} \right\} \ll \text{имя каталога - K} \gg \left[\ll \text{размер области} \gg \ll \text{вывода - d} \gg \right]$

После последнего выводимого на печать символа записывается в d позиции информация из каталога K, которая соответствует содержанию данной, указанной первым операндом. Каталог K должен быть заранее определен.

Оператор типа R позволяет для баз данных, в которых информация хранится в сжатом виде, выводить на печать реальные значения объектов. Кроме того, при замене одного значения объекта другим, нет необходимости делать реорганизации всей базы данных, а очень часто достаточно заменить в каталоге соответствующее значение новым.

Оператор типа T (разделить)

$T \left\{ \begin{array}{l} \text{имя переменной} \\ \text{имя элемента} \end{array} \right\}, \dots, \left\{ \begin{array}{l} \text{имя переменной} \\ \text{имя элемента} \end{array} \right\} = \ll \text{набор символов} \gg$

Этот оператор задает набор символов, которые могут быть разделителями для данных из первых операндов, превышающих размер "области вывода" и выводящихся на печать в следующих строках.

Оператор типа T обеспечивает нормальную форму при выводе естественных текстов (статей, докладов и т.д.) на печать - не разывая слов при необходимости переносов.

Рассмотренные операторы управления поиска и форматирования выполняются в порядке, в котором они заданы. Используя описанную систему операторов, пользователь может задавать тоже заголовки и подзаголовки отчетов.

На основе представленной системы операторов был создан язык, который идентичен со структурой операторов. Языковой процессор связан со системой БИСЕС, разработанной ИМ - БАН. Он обеспечивает высокую эффективность. Проверка синтаксиса каждого оператора выполняется во время его ввода в систему. При диалоговом режиме работы пользователь имеет возможность исправлять появляющиеся ошибки. Он может задать только ошибочной операнд, игнорировать все следующие операнды и вводить их заново, отказаться исправлять ошибки или игнорировать весь оператор. После ввода "пользовательской программы" система дает ему возможность корректирования, удаления, вставки, замены и добавления операторов.

Языковой процессор работает на двух уровнях. Для системных потребителей он дает во время работы только необходимую информацию, а для не профессионалов - более подробную. Если потребитель хочет, он может получить дополнительные сведения об операндах данного оператора, о базе данных и логической подсхеме, о всех операторах.

При вычислительных системах с множественным доступом к базам данных языковой процессор обеспечивает возможность одновременного доступа нескольких пользователей к их доступным данным при помощи дистанционного пульта. При надлежащем управлении вычислительная система коллективного пользования может обеспечить хорошие характеристики обслуживания многим пользователям при помощи одновременного применения технических средств и уменьшению времени затрачиваемого на фазе раз-

мышлений конкретного пользователя.

Рассмотренная система операторов экспериментирована в нескольких информационных системах. Результаты практической эксплуатации показывают, что она перспективна и эффективна.

ЛИТЕРАТУРА

1. Клеинрок, Л., "Вычислительные системы с очередями", Москва, 1979
2. Мартин, Дж., "Организация баз данных вычислительных системах", Москва, 1978
3. КОДАСИЛ, "Информационные системы общего назначения", Москва, 1975
4. Pike, H., E., "Software production for computers", proceed. of the IEEE, v. 61, N 11, 1973
5. Mc Gee, W., C., "Generalization: key to successful electronic data processing", J. ACM, v. 6, 1, Jan., 1959, 1-23
6. Taylor, R. W., Frank, R. L., "CODASYL Data Base Management Systems", ACM Computing surveys, v. 8, N 1, 1976
7. Fry, J. P., Sybley, E. H., "Evolution of Data Base Management Systems", ACM Computing surveys, v. 8, N 1, March 1976
8. System IBM/360 "Administrative Terminal System", IBM corp.
9. "PAC II - Project Analises and Control", IBM Corp., 1976

ЗАЩИТА КОРЕКТНОСТИ ДАННЫХ ПРИ ВВОДЕ В СУБД БИСЕС

Й. Швертнер ИИ, БАН, София

Одной из основных проблем, возникающих при проектировании систем управления базами данных (СУБД) является проблема обеспечения требуемой достоверности данных. Без принятия особых мер по повышению достоверности при выполнении обновления, а также технологических процессов в СУБД, вероятность появления ошибок превышает предельно допустимое значение. Поэтому разработка методов обеспечения качества информации должна быть неотъемлемой частью процесса создания СУБД. Однако из всех задач СУБД, в большинстве случаев, наименьшее внимание уделяется вопросу поддержания качества данных. Это обстоятельство указывает на существенные просчеты при проектировании СУБД, тем более, что согласно некоторым авторам (3) 40% процедурных разделов современных промышленных программ состоят из операторов проверки ошибок. Поддержание достоверности представляет собой достаточно большую проблему. Отсутствует какой либо существенный опыт, даже в системах с одним пользователем, т.е. в системе, в которой в любой момент времени с базой данных может работать только один пользователь. Поэтому для обеспечения достоверности данных в СУБД жизненно необходимо разрабатывать дополнительное программное обеспечение, предназначенное для проверки достоверности данных.

В некоторых случаях потребители СУБД (особенно в случае СУБД с базовым языком) разрабатывают и подвключают к системе программы проверки на достоверность. Однако в большинстве случаев этот подход обладает недостатками. К ним относятся:

1. Невозможность систематическим способом отражать структуру, семантику и набор правил, характеризующих предметную область. Под предметной областью понимается часть окружающей

действительности, информация о которой сохраняется в базе данных (БД).

2. Так как контроль совершается специально составленными процедурами, последние не создаются на основе хорошо продуманного подхода или методологии. Очень трудно делать перемены в алгоритмах контроля, а также охватить все возможные ошибки.

3. Вышеупомянутые средства контроля требуют большой затраты труда и средств, а также много времени на программирование.

Нами были разработаны средства проверки достоверности данных при обновлении, которые подвключены к функции обновление СУБД БИСЕС. Эта система разработана в Институте математики Болгарской Академии наук под руководством доцента Добрева. Система является СУБД замкнутого типа, но тем не менее есть возможность использовать в качестве базового языка Асемблер для ЕС ЭВМ. Существуют два уровня средств контроля в СУБД БИСЕС. Первый тип контроля это синтаксический контроль как структуры записей на обновление, так и обновляющих данных. Он не может управляться (включаться или выключаться) потребителями и осуществляется в обязательном порядке. Осуществляются проверки на принадлежность содержания обновляющих данных к некоторому алфавиту, на правильность представления числовых элементов и т.д. Вторым типом контроля является семантический контроль. Он является необязательным средством. Данные для него предоставляются потребителями в соответствии с их представлениями о содержании элементов БД. Следующие принципы являются основными для средств семантического контроля данных при обновлении.

ПРИНЦИП 1. Средства контроля являются декларативными. Это означает, что существуют специально разработанные языковые средства, для описания предметной области и для отражения законов и ограничений, которые характеризуют ее. Эти средства сравнительно просты и удобны для работы. Кроме того они являются непроцедурными. Это означает, что потребитель избавлен от необходимости давать шаг за шагом описание операций,

связанных с процессом проверки. Отметим, что обычно к числу подобных операций относятся сложные операции, как например чтение и запись логических записей БД, управление буферами ввода-вывода и т.д. Этот подход к решению задач контроля имеет следующие преимущества. Во-первых, легко отражается информация, необходимая для нормального функционирования системы, а также программных средств, для обеспечения семантической надежности БД. Во-вторых, изменения и дополнения в описании предметной области могут осуществляться просто и с необходимой степенью точности. В-третьих, все ограничения становятся обозримыми и возможно устранить все неточности, а также устранить все противоречия между ограничениями без особых трудностей и затраты труда и времени.

ПРИНЦИП 2. Средства отражения правил и ограничений, отражающих предметную область, имеют достаточно большую выразительную силу. С их помощью можно представить достаточно сложные правила и ограничения. Обычно правила и ограничения составляются средне квалифицированными потребителями, которые знакомы с логической структурой БД и имеют минимальные знания в области логических операций. Поскольку средства контроля при обновлении полностью автоматические, обновление данных может совершаться неквалифицированным потребителем, а средства контроля сообщают ему об обнаруженных ошибках.

ПРИНЦИП 3. Потребитель имеет возможность глобально (на уровне семантического контроля всех элементов) или локально (на уровне семантического контроля отдельного элемента) изменять тяжесть реакции на обнаруженную ошибку. В частности он может при помощи особого декларативного оператора указать следующие реакции при обнаружении ошибки в обновляемом элементе:

- обновление ошибочного элемента отвергается; обновление остальных элементов в той же записи отвергается;
- ошибочный элемент обновляется; другие элементы в той же записи тоже обновляются;
- ошибочный элемент получает особое значение; остальные элементы той же записи обновляются.

Кроме того потребитель может при обнаружении ошибки в данном

элементе подготовить текст, который будет печататься вместе с текстом системного сообщения или вместо него.

ПРИНЦИП 4. Потребитель может получать информацию о видах ограничений и проверок, касающихся отдельных элементов БД.

ПРИНЦИП 5. Различным потребителям предоставлена возможность включать свои наборы ограничений и правил контроля, т.е. иметь свой взгляд на контроль БД. Разработаны также средства динамического управления параметрами контроля. Потребитель может динамически менять наборы ограничений, вызывая другие наборы из архива системы, а также динамически включать или выключать контроль для данного элемента записей БД.

Для составления наборов ограничений используются декларативные операторы, которые дают возможность потребителю управлять процессом контроля. Параметры операторов позиционно го типа.

1. Для элементов числового типа существуют следующие декларативные операторы:

1.1. Проверка на принадлежность к интервалу на числовой оси:

$(X:Y)$

где X - нижняя, а Y - верхняя граница интервала

1.2. Сравнение с другим элементом числового типа в той же записи

$(r1,eee)$

где $r1$ одно из отношений EQ, NE, GT, GE, LT, LE

а eee идентификатор элемента, с которым сравнивается проверяемый элемент.

2. Для элементов строкового типа существуют следующие декларативные операторы:

2.1. Сравнение с эталоном знаков

где $(E, \begin{cases} NE \\ EQ \end{cases}, p, 'text')$

EQ, NE - указывает тип проверки (совпадение или несовпадение)

p - начало подполя в элементе

$'text'$ - эталон знаков, с которым сравнивается указанное

подполе элемента

2.2. Сравнение с содержанием другого элемента строкового типа

$$(C, \begin{cases} EQ \\ NE \end{cases}, p1, l, eee, p2)$$

где

- EQ, NE - смотри (см.) 2.1.
- p1 - начало подполя в сравниваемом элементе
- l - длина подполя
- eee - элемент, с которым сравнивается сравниваемый элемент (оба элемента принадлежат одной записи)
- p2 - начало подполя в элементе

2.3. Принадлежность к объединению множеств знаков

$$(M, \begin{cases} EQ \\ NE \end{cases}, p1, l, M1+M2+\dots+Mn)$$

где

- EQ, NE - см. 2.1.
- p1 - см. 2.2.
- l - см. 2.2.
- M1+...+Mn - Mi - имена множеств знаков, объединение которых определяет используемые для проверки множества знаков. Эти множества указываются потребителями специальным оператором.

2.4. Проверка подполя элемента строкового типа на принадлежность к числовому интервалу

$$(T, p1, l, w.d, x:y)$$

где

- p1 - см. 2.2.
- l - см. 2.2.
- w.d - формат числа, где w - общее число цифр и знака а d - число цифр после десятичной точки
- x:y - см. 1.1.

2.5. Числовое сравнение частей двух элементов строкового типа

$$(Q, p1, l, w.d, r1, eee, p2, l2, w1.d1)$$

где

- p1 - см. 2.2.
- l - см. 2.2.

w.d	- см. 2.4.
r1	- см. 1.2.
eee	- см. 1.2.
p2	- см. 2.2.
l2	- длина подполя в элементе eee
wl.dl	- см. 2.4.

Различные элементарные операторы могут быть соединены между собой логическими операциями конъюнкция и дизъюнкция. Кроме вышеописанных операторов существуют операторы управления процессом контроля и управления печатью сообщений об обнаруженных ошибках в процессе контроля. Разработанные программные средства применены с успехом в ряде информационных систем, сгенерированных с использованием СУБД БИСЕС.

ЛИТЕРАТУРА:

- (1) Добрев Д.М. Архитектура и основные принципы системы БИСЕС, сборник докладов Конференции по системам информационного обслуживания коллективов профессионально связанных потребителей, Варна, 1977
- (2) Добрев Д.М. Архитектура и базисное математическое обеспечение ВЦКП, сборник докладов Конференции по вычислительным центрам коллективного пользования, Благоевград, 1979
- (3) Fry J. P., *Evolution of Data-Base Managment Systems*, ACM Computing Surveys, 1, 1976
- (4) Дейт, Введение в системы баз данных, Москва, 1980
- (5) Пивоваров А.Н. К вопросу о требуемой достоверности информации при ее автоматизированной обработке, Управляющие системы и машины, 3, 1977

СИСТЕМНЫЙ ЖУРНАЛ В СУБД БИСЕС

Йосиф Швертнер, Институт математики БАН

Людмила Манасиев, Институт вычислительной техники

В настоящей статье под понятием системный журнал будем иметь в виду файл, который создан в результате:

1. или системной регистрации заказов на обновление,
2. или системной регистрации модификаций базы данных,

а также программные средства генерирования, управления и использования этого файла. Использование системного журнала не является обязательным. Он генерируется и заполняется данными только если потребитель системы считает это нужным. В зависимости от данных, которые регистрируются в системный журнал, существуют два типа системных журналов, которые могут быть использованы потребителем СУБД БИСЕС. Мы рассмотрим их основные преимущества и недостатки.

Основным предназначением файла системного журнала является возможность использовать данные в качестве информационной базы в целях восстановления физически негодных баз данных в их последнее активное состояние. Кроме того журнальные файлы могут быть использованы для возвращения базы данных в некоторое из ее прежних состояний, которое по некоторым причинам интересует потребителя системы.

Журнальный файл, полученный в результате регистрации за-

казов на обновление позволяет восстановление базы данных в ее последнее состояние/или в некоторое состояние, являющееся близким к последнему активному если используется так называемый откат/1//. Эти состояния могут быть получены с использованием файла системного журнала и копии базы данных в момент генерации системного журнала. К преимуществам этого типа журнального файла следует причислить прежде всего минимальный объем файла, а также простоту, с которой он создается и поддерживается. Минимальный объем файла достигается благодаря обстоятельству, что в файл записываются только заказы на обновление. Таким способом при групповом обновлении, ведущем к обновлению большей части или всех записей в базе данных, избегается занесения в файл системного журнала большого количества информации, а записывается только текст заказа на обновление. Подобное замечание относится к некоторым операциям, которые требуют изменения всех элементов некоторой записи базы данных, как например операция инициализации записи. Маленький объем этого типа журнального файла позволяет даже его хранение на том же носителе данных прямого доступа, на котором находится база данных, хотя это не рекомендуется в виду возможной порчи носителя, при которой будет уничтожен вместе с базой данных и файл системного журнала, а также большой прогон читающе-записывающих головок. Таким способом существенно упрощаются процедуры активизирования системы. Другое важное преимущество, то что фиксируются все действия потребителей, связанных с обновлением данных. В дальнейшем эти данные могут быть анализированы Администратором базы данных или потребителями.

Указанные удобства компенсируются некоторыми неудобствами. Одно из этих неудобств это то, что при системной регистрации заказов на обновление не возможна регенерация базы данных в обратном направлении, т.е. от последнего ее состояния к прежнему. Необходимо отметить, что существование такой возможности увеличивает свободу потребителей, а в некоторых случаях приводит к более быстрому восстановлению базы данных. Другим неудобством является необходимость регистрировать в журнальный файл не только заказы на обновление, но и данные, связанные с обновлением/например таблицы форматов для поэлементного обновления по формату, комплекты данных для семантического контроля при обновлении/, которые хранятся в архиве системы. В противном случае регенерация может быть совершена неправильно. Журнальный файл, полученный регистрацией модификаций элементов базы данных позволяет восстановление базы данных в ее последнее состояние с применением копии базы данных в момент генерации файла системного журнала. Кроме того, используя последнее состояние базы данных, можно получить некоторое из прежних состояний базы данных/регенерация в обратном направлении/. К преимуществам этого типа системного журнала следует причислить возможность регенерации базы данных от прежнего состояния к последнему, а также и наоборот. При этом отпадает необходимость занесения в журнальный файл динамики таблиц, форматов и комплектов данных для семантического контроля, поскольку они учтены при обновлении элементов. В принципе регенерация базы данных совершается быстрее, так как содержание элементов сохраняется во внутреннем виде, после соответствующей

щих преобразований и контроля. Это позволяет не проводить преобразования во внутренний вид и контроль данных при регенерации. К недостаткам этого типа системного журнала следует причислить прежде всего то, что объем журнального файла сильно увеличивается при групповых обновлениях записей в базе данных. Поэтому этот тип журнального файла следует хранить на магнитной ленте. Это приводит к эксплуатационным трудностям. Другой недостаток - это большая сложность программ. Однако этот недостаток несуществен, поскольку быстродействие программных средств не уменьшается, а тяжесть создания программного обеспечения лежит не на потребителях, а на создателях системы. Отметим, что с эксплуатационной точки зрения все действия, связанные с использованием системных журналов, максимально автоматизированы. Существуют несколько типов заказов, которые позволяют управлять всеми процессами использования системных журналов. Программы обслуживания системного журнала выполняют ряд действий, короткое описание которых будет дано ниже.

Первый модуль является модулем генерации файла системного журнала. Этот модуль выполняется каждый раз, когда необходимо сгенерировать новый файл системного журнала. Для журнальных файлов типа системной регистрации модификаций базы данных на магнитной ленте создается файл с последовательной организацией и в нем записывается головная запись. Головная запись содержит ряд данных, характеризующих базу данных, среди которых отметим:

- имя базы данных;
- дату создания базы данных;

- вид базы данных;
- длину логических записей в базе данных;
- символ пароля защиты базы данных.

После головной записи записывается метка конца файла.

Для журнальных файлов типа системной регистрации заказов на обновление на пакете магнитных дисков/ЗУПД/ создается файл с прямой организацией. Физические записи имеют длину 3604/7204/ байта и содержат 45/90/ логических записей с длиной 80 байтов каждый. В файле организована система указателей, указывающих на заполнение файла физическими записями и физических записей - логическими записями. Основная цель этих указателей - создать возможность как прямого, так и последовательного доступа к логическим записям. В частности, это используется для отыскания первой свободной логической записи в файле в обход последовательного чтения всех записей с начала файла. Одним из важных преимуществ выбранного метода является обстоятельство, что при аварии в вычислительной системе файл остается доступным и возможны все обычные манипуляции с ним. Кроме того очень просто добавлять новые записи.

Второй модуль служит для регистрации данных в системный журнал. Логические записи и информация, содержащиеся в данных имеют длину и структуру, соответствующую типу системного журнала. Некоторые атрибуты записей следующие:

- имя задания;
- дата обновления;
- час обновления;
- метод обновления;

- старое и новое содержание элементов обновляемой записи /при регистрации модификаций базы данных/;

- текст заказа на обновление/при регистрации заказов на обновление/.

Третий модуль служит для регенерации базы данных в прямом направлении. Используется копия базы данных в момент генерации системного журнала и, используя системный журнал, она приводится в актуальное состояние.

Четвертый модуль служит для регенерации в обратном направлении. База данных с помощью системного журнала возвращается в свое желаемое прежнее состояние. Это возможно только в случае журнала типа модификаций базы данных.

Пятый модуль/только для журнала типа заказов на обновление/ выдает на печать справку о состоянии и содержании файла системного журнала. Используя этот модуль можно получить представление о степени заполненности файла системного журнала. Кроме того можно получить информацию о том, в каком порядке и каким способом были сделаны перемены в базе данных.

Наличие системного журнала в списке основных средств СУБД БИСЕС увеличивает надежность и эффективность системы.

ЛИТЕРАТУРА:

/1/ К. Дейт, Введение в системы баз данных, Москва, 1980

/2/ Комитет КОДАСИЛ, Информационные системы общего назначения, Москва, 1976

ПОИСК ПРИ ПОМОЩИ ИНВЕРТИРОВАННЫХ ФАЙЛОВ В СИСТЕМЕ БИСЕС

А. Эскенази, Н. Манева, В. Петрова

Институт математики БАН, София, Болгария

В первых версиях системы БИСЕС была реализована функция "выбор"/1/. Эта функция позволяет по произвольному логическому условию над произвольным подмножеством элементов записей (документов) в базе, выбрать те документы, которые удовлетворяют заданному условию. Функция выбора дает возможность осуществлять поиск во всей базе или в подмножестве базы, отметить требуемые записи в обрабатываемой базе или создать новую базу, содержащую только выбранные документы. В связи с тем, что первые версии системы БИСЕС были разработаны для работы в пакетном режиме, при реализации алгоритмов была сделана оптимизация, имея в виду особенности этого режима, т.е. увеличение компактности программ, без дополнительной информации на внешних носителях. В связи с развитием системы и расширением возможностей работы в режиме реального времени, появилась необходимость создать аналогичные способы поиска и выбора, которые оптимальны по времени ответа на запрос пользователя.

Согласно /2/ существуют два основных способа организации и использования данных в информационных системах. При первом, каждая запись содержит значение атрибутов каждого объекта и это удобно, когда большинство заказов относится к свойствам данного объекта. Второй способ - инверсия первого - получают идентификаторы объектов, связанные с заданными атрибутами и удобен, когда заказ относится к объектам ~~имеющим~~ данное свойство. Поиск в таких инвертированных файлах осуществляется по вторичному дополнительному ключу, т.е. используется ключ, который определяет не только одну запись, а все записи, обладающие данным свойством.

Предмет настоящего сообщения - разработанные в системе

БИСЕС средства быстрого поиска, реализованного при помощи инвертированных файлов. Мы будем называть инвертированные файлы - быстрые каталоги в терминах системы БИСЕС.

1. ПОИСК С ТОЧКИ ЗРЕНИЯ ПОЛЬЗОВАТЕЛЯ

Здесь мы рассмотрим функции по поддержке и использованию каталогов с точки зрения пользователя. Вызов подсистемы обслуживания каталогов осуществляется заказом, который содержит одну начальную карту и карты с описанием других функций по поддержке каталогов. Здесь слово карта употребляется в смысле перфокарты или строки на видеотерминале. Начальная карта выглядит следующим образом:

SKaoi

SK - заказ на вызов подсистемы;

o - защитный код базы;

i - номер результатной базы;

- номер вводной базы (номер базы = 0,1,2,3).

1.1 ОТКРЫТИЕ КАТАЛОГОВ

Пользователь имеет возможность определить не более 8 каталогов. Для каждого каталога в заказе указывается:

- номер каталога, который становится его идентификатором для других обработок;

- длина элемента каталога, т.е. длина вторичного ключа;

- описатели для формирования вторичного ключа.

Вторичные ключи могут содержать:

1) элемент

2) часть элемента (начальная или нена начальная)

3) комбинацию 1) и 2).

При помощи специального языка, подмножество языка функции "переструктурирование" в системе БИСЕС, описатели E выглядят следующим образом:

$$E = \left\{ \begin{array}{l} nnn-n_1 n_1 n_1 \\ nnn [111 [ppp]] \end{array} \right\}$$

$n_{11}n_1n_1n_1$ - содержание вторичного ключа формируется из содержаний элементов $n_{11}, n_{11}+1, \dots, n_1n_1n_1$

во втором случае n_{11} - номер элемента, l_{11} - число позиций, которые надо взять из этого элемента (по умолчанию берется длина элемента с номером n_{11}), r_{11} - номер позиции в элементе, с которой надо начинать (по умолчанию $r_{11} = 1$).

Цифровые элементы могут участвовать во вторичном ключе только полностью. Когда нет совпадения указанной при открытии каталога и сформированной через описатели длины, происходит выравнивание влево. Причем (если необходимо) производится сокращение до требуемой длины или расширение пробелами. Если длина сформированного ключа больше указанной в заказе и последний элемент в ключе цифровой, то он теряется и пользователь получает подходящее сообщение. Пример:

K7040011,002-004,107001027

- открывается новый каталог с порядковым номером 7, с длиной ключа - 40 байтов. Ключ формируется из содержаний элементов 11, 2, 3 и 4 и 27-ого байта элемента с порядковым номером 107.

Если указан номер каталога 0, все остальные позиции заказа не обрабатываются и производится форматирование всех полей и таблиц, связанных с каталогами. Пользователь должен однократно, перед открытием какого-либо каталога, дать заказ:
K0

Очевидно, что такой заказ можно использовать для уничтожения всех дефинированных до этого момента каталогов.

1.2. УНИЧТОЖЕНИЕ КАТАЛОГА

В данный момент пользователь системы может решить, что некоторые из уже созданных каталогов, ему не нужны. Заказ на уничтожение каталога выглядит Dn , где n - номер каталога, подлежащего уничтожению. После уничтожения каталога он становится недоступным с точки зрения пользователя, а с точки зрения системы - информация, которая связана с этим каталогом,

удаляется.

1.3. ПОИСК

Обеспечивается выбор тех документов, которые удовлетворяют заданным условиям. Предоставляется возможность отметить маской требуемые документы в основной базе. Маска, это последовательность из 7 битов, со стандартным расположением для каждого документа. Возможно и создание новой информационной базы, содержащей только выбранные документы, вывод, по желанию пользователя, списка номеров тех документов, которые удовлетворяют заданному условию.

Предлагаемый язык прост и удобен и выглядит следующим образом:

$Snr_b r_c r_m$

xx 'шаблон', 'шаблон'

S - заказ на поиск;

n - номер каталога;

r_b - указывает наличие или отсутствие результирующей базы и ее тип;

r_c - указывает требует пользователь или нет вывода списка с номерами выбранных документов;

r_m - указывает наличие вводной и/или выводной маски;

xx - задает отношение между заданным шаблоном/заданными шаблонами и элементами каталога.

Следующие функции 1.4 и 1.5 поддерживаются автоматически, но все таки пользователю дана возможность указать нужны они ему или нет.

1.4. РЕОРГАНИЗАЦИЯ

Заказ на реорганизацию выглядит следующим образом: Rn
n - номер каталога, который подлежит реорганизации. Так как каталоги физически организованы как индексно-последовательные файлы, при реорганизации, в последовательном файле переписы-

ваются только актуальные элементы, а потом возвращаются в индексно-последовательный файл.

1.5. ПОДДЕРЖКА КАТАЛОГОВ

В поддержку каталогов входит первоначальное создание и обновление. Первоначальное создание - это заполнение соответствующего каталога, используя записанную при открытии каталога информацию.

Некоторые подсистемы (например подсистема ввода) меняют содержание базы, т.е. содержание некоторых элементов, которые может быть участвуют в каталогах. Поэтому нужны обновление и поддержка каталогов в соответствии со состоянием базы. Заказ на поддержку: S_n , где n - номер каталога.

2. РЕАЛИЗАЦИЯ

2.1. ИДЕЯ РЕАЛИЗАЦИИ

Для реализации описанных возможностей используются элементы определенного метода, который мы уже применяли несколько раз /4/. Этот метод разработан на основе индексно-последовательного метода доступа в ДЭС /5/. Индексно-последовательный метод доступа в ДЭС позволяет прямое обращение к записи по ее ключу, а так же - и последовательную обработку по возрастанию ключа всего файла или его части. К сожалению, в этой универсальности все-таки есть некоторые недостатки. Самые главные из них - накопление записей в дорожках переполнения и трудное выполнение произвольных последовательностей разных типов обработок. Причина второго недостатка та, что функции обработки слишком различны и для каждой из них необходима специальная таблица для дефиниции файла (ДФ-таблица). Здесь мы не будем рассматривать подробно метод, который описан в /4/. Мы отметим только, что его сущность - дефинирование 3 или 4 логических файлов (т.е. 3 или 4 ДФ-таблицы) для единственного физически существующего индексно-последовательного файла.

По заказу на определенный тип обработки, если имеется другой открытый логической файл, он закрывается. Открывается, если еще не открыт, необходимый для требуемой обработки логический файл, и производится с ним необходимая обработка. Это из за того, чтобы в каждый момент иметь не более одного открытого логического файла и при этом оптимизировать число открытий файлов. С другой стороны приняты меры и для реорганизации файла (автоматично или по заказу пользователя, см.1.4.) с целью минимизации числа записей в областях переполнения. Следовательно физически на диске для каждого дефинированного пользователем каталога создается и поддерживается только один индексно-последовательный файл, а в программе - 3 или 4 логических файлов. При заказе связанном с каталогом к данной базе, автоматически генерируется имя файла. Этот файл открывается (т.е. осуществляется связь между логическим и физическим файлами) и потом производится соответствующая обработка. Например, название базы - ABCABC. Пользователь требует поиска при помощи каталога 3. Программа генерирует название ABCABE3, закрывает, используя заданную этикетную информацию, логический файл, которым пользовалась до этого момента, открывает логический файл, предназначенный для прямого поиска, заменяет в его ДФ-таблице название ABCABE3 и начинает работу с файлом. Если необходимый файл уже открыт, то вторая и третья операции пропускаются.

2.2. СТРУКТУРА КАТАЛОГА

Запись в каждом каталоге (в индексно-последовательном файле) имеет следующую структуру: ключ (ведущий элемент инвертированного файла), 3 байтовый номер документа и признак актуальности длиной 1 байта. Первые 2 элемента составляют ключ в смысле индексно-последовательного файла. Признак актуальности необходим, потому что в ДОС не поддерживается функции удаления записей.

Из всего этого следует, что ДФ-таблицы должны служить для обработки индексно-последовательных файлов с разной длиной записи (из-за разной длины ключа). Это требует настройки соответствующей ДФ-таблицы в зависимости от длины записи обрабатываемого каталога. Эта настройка связана с исчислением и с загрузкой нескольких констант и адресов в ДФ-таблице. При ее реализации были использованы части из /5/.

После оценки факторов, которые связаны с требованиями пользователя, с емкостью внешних носителей, с требованиями, связанные с буферной памяти и т.д., допустимая максимальная длина ключа фиксирована на 106. Записи блокируются так, чтобы длина блока была не больше 1680 байтов, но пользователь не обязан знать об этом.

2.3. ПОДДЕРЖКА КАТАЛОГА

Для поддержки так описанных файлов (каталогов) необходима информация, которая сохраняется в виде таблиц вместе с описаниями основной базы. Эта информация:

- а) длина таблицы каталога;
- б) длина ключа;
- в) число актуальных элементов (записей) в каталоге;
- г) признаки, указывающие состояние каталога. Различаются следующие состояния: 1) каталог в готовности; 2) в базе сделано не менее одного изменения, которое связано с элементами каталога; 3) в базе сделано по крайней мере одно массовое изменение. В последнем случае сохраняется маска первого массового изменения.

д) максимальный ключ. Он позволяет медленнее заполнять области переполнения. Если поступающая запись имеет ключ, больше максимального до этого момента, он записывается при помощи индексно-последовательной функции "загрузка" и так сохраняется в основной области. В противном случае (по необходимости), ключ записывается при помощи индексно-последователь-

ной функции "добавление" и следовательно записывается в области переполнения.

е) информация о структуре ключа. Эта информация позволяет формировать ключ каталога из элементов или из их частей так, как дефинирован ключ при открытии каталога (см.1.1.).

Вероятно самая тяжелая проблема - поддержка каталога в актуальном состоянии, т.е. в состоянии, в котором все наступившие изменения приняты во внимание и отражены. Дело в том, что некоторые из функций системы БИСЕС могут дать изменения, относящиеся к одному или к нескольким каталогам. Возникает вопрос, целесообразно ли в момент изменения актуализировать соответствующий каталог? Поддержка каталога включает две части:

а) каждая функция системы БИСЕС при изменении элемента в основной базе обращается к специальной подпрограмме. Эта подпрограмма проверяет изменяемый элемент (изменяемые элементы) является или не является частью ключа каталога (каталогов). Если это так, номер измененного документа записывается в списке А. Каждый элемент этого списка содержит номер и 8-битовую маску (по одному биту на каждый каталог). Если к-тый бит равен 1, это значит, что для документа с данным номером в основной базе было сделано изменение, касающее к-того каталога (к от 1 до 8). Одновременно делается отметка в признаках (см.2.2.).

б) в момент заказа на поиск для данного каталога или заказа С (см.1.5.) проверяются признаки. Если сделаны только индивидуальные изменения (т.е. для отдельных документов), просматривается каталог. Для всех записей каталога, содержащих номер документа, который участвует в списке А, документ из базы читается, записывается новая запись в каталоге, а существующая запись становится неактуальной. После этого соответствующий бит маски в А анулируется. В случае массовых изменений или когда длина списка А недостаточна, чтобы отразить все изменения, каталог актуализируется путем просмотра базы.

Аналогичен подход и в случае первоначального создания каталога.

ЛИТЕРАТУРА

- /1/ Добрев Д., Киркова Р., Първанов П. Подсистемы обработки информации системы БИСЕС. Конференция по системам информационного обслуживания коллективов профессионально-связанных потребителей. Сборник докладов, 1977.
- /2/ Мартин Дж. Организация баз данных в вычислительных системах, Москва, 1978.
- /3/ Ескенази А., Петрова В., Манева Н. Подсистема генерирования системы БИСЕС. см. /1/.
- /4/ Ескенази А. Усъвършенствуване използване на индексно-последователния метод в ДОС/ЕС. сп.АСУ, 2/1979.
- /5/ Димитров Е. и др. Единна система от ЕИМ. Модел 1020. Част II. Техника, 1976.

A TANULMÁNSOROZATBAN 1981–BEN MEGJELENTEK:

- 116/1981 Siegler András: Egy 6 szabadságfokú antropomorf manipulátor kinematikája számítógépes vezérlése
- 117/1981 Knuth Előd – Radó Péter: Principles of Computer Aided System Description
- 118/1981 Demetrovics János – Gyepesi György: Általános függőségek és lekérdezéssel kapcsolatos algoritmusok relációs adatmodellekben
- 119/1981 Sztanó Tamás: REAL–TIME programrendszerek eseményvezérelt szervezése
- 120/1981 Szentgyörgyi Zsuzsa: A számítástechnika műszaki fejlődése és társadalmi hatásai
- 121/1981 Vicsek Tamásné (Strehó Mária) : Vizsgálatok a kezdeti érték problémák numerikus megoldásával kapcsolatban
- 122/1981 Andó Györgyi – Lipcsey Zsolt: Sztochasztikus Ljapunov módszerek és alkalmazásaik
- 123/1981 Márkus Zsuzsanna: Intelligens interaktív rendszerek elvi problémái
- 124/1981 Márkus Zsuzsanna: Logikai alapú programozási módszerek és alkalmazásaik számítógéppel segített építészeti tervezési feladatok megoldásához
- 125/1981 Fabók Julianna: Software implementációs nyelvek
- 126/1981 Várszegi Sándor: Multimikroszámítógépes-rendszerek
- 127/1981 Lipcsey Zsolt: N-személyes minőségi differenciáljátékok késleltetéssel és késleltetés nélkül
- 128/1981 Böszörményi László: Multa-task rendszerek fejlesztése magasszintű nyelven
- 129/1981 Tóth János: A formális reakciókinetika globális determinisztikus és sztochasztikus modelljéről és néhány alkalmazásáról

