Лекция 9

Информационное обеспечение САПР

Управление данными в САПР

В большинстве автоматизированных информационных систем применяют системы управления базами данных (СУБД), поддерживающие реляционные модели данных.

Общие требования к СУБД:

1) обеспечение целостности данных (их полноты и достоверности);
2) защита данных от несанкционированного доступа и от искажений из-за сбоев аппаратуры;
3) удобство пользовательского интерфейса;
4) в большинстве случаев важна возможность распределенной обработки в сетях ЭВМ.

Первые два требования обеспечиваются

- ограничением прав доступа,
- запрещением одновременного использования одних и тех же обрабатываемых данных (при возможности их модификации),
- введением контрольных точек (checkpoints) для защиты от сбоев
- и т.п.

Банк данных (БнД) в САПР является важной обслуживающей подсистемой, он выполняет функции информационного обеспечения и имеет ряд особенностей. В нем хранятся как редко изменяемые данные (архивы, справочные данные, типовые проектные решения), так и сведения о текущем состоянии различных версий выполняемых проектов.

БнД работает в многопользовательском режиме, с его помощью осуществляется информационный интерфейс (взаимодействие) различных подсистем САПР.

Построение БнД САПР — сложная задача, что обусловлено следующими особенностями САПР:

1

Разнообразие проектных данных, фигурирующих в процессах обмена как по своей семантике (многоаспектность), так и по формам представления. В частности, значительна доля графических данных.

2

Нередко обмены должны производиться с высокой частотой, что предъявляет жесткие требования к быстродействию средств обмена (полагают, что СУБД должна работать со скоростью обработки тысяч сущностей в секунду).

3

В САПР проблема целостности данных оказывается более трудной для решения, чем в большинстве других систем, поскольку проектирование является процессом взаимодействия многих проектировщиков, которые не только считывают данные, но и изменяют их, причем в значительной мере работают параллельно.

Вследствие этого:

во-первых, итерационный характер проектирования определяет наличие нескольких равноценных версий всех частей проекта, поэтому возникает необходимость сохранения всех версий с возможностью возврата к любой из них;

во-вторых, нельзя допускать использования неутвержденных данных, поэтому проектировщики должны иметь свое рабочее пространство в памяти и работать в нем автономно, а моменты внесения изменений в общую БД должны быть согласованными и не порождать для других пользователей неопределенности данных.

4

Транзакции могут быть длительными и трудоемкими.

Транзакцией называют последовательность операций по удовлетворению запроса.

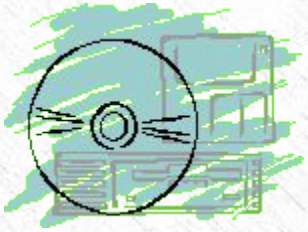
В САПР внесение изменений в некоторую часть проекта может вызвать довольно длинную и разветвленную сеть изменений в других его частях из-за существенной взаимозависимости компонентов проекта (многошаговость реализации запросов). В частности, транзакции могут включать в себя такие трудоемкие операции, как верификация проектного решения с помощью математического моделирования.

В результате транзакции могут длиться даже несколько часов и более. Одна из трудностей заключается в отображении взаимозависимости (ассоциативности) данных. При хранении компонентов проекта во внешней памяти затраты времени на обработку запросов оказываются значительно выше, чем в большинстве других автоматизированных систем, с менее выраженными взаимозависимостями данных.

5

Иерархическая структура проектных данных и, следовательно, отражение наследования в целях сокращения объема базы данных.

В определенной мере названные особенности учитываются в СУБД третьего поколения, в которых стали применяться черты объектно-ориентированных (объектных) СУБД. В них наборы данных, характеризующих состояние предметной области (состояние проекта в случае САПР), помещаются в отдельные файлы.



Интерпретация семантики данных осуществляется с помощью специальных процедур (методов), сопровождающих наборы. Наследование свойств объектов предметной области выражается с помощью введения категорий класса, надкласса, подкласса. Информационные модели приложений для таких СУБД разрабатываются на основе методик типа IDEF1X.

Объектные БД выгодны,

во-первых, тем, что данные по конкретным объектам проектирования не разбросаны по множеству таблиц, как это имеет место в реляционных БД, а сосредоточены в определенных местах.

Во-вторых, для каждого объекта могут быть назначены свои типы данных.

В результате проще решаются задачи управления и удовлетворения за[illegible]з.

Наряду с чисто объектными СУБД (pure ODBMS), применяют ***СУБД объектно-реляционные***.

В последних происходит объединение свойств реляционных и объектно-ориентированных СУБД: объектно-ориентированная СУБД снабжается непроцедурным языком запросов или в реляционную СУБД вводятся наследование свойств и классы.

Непроцедурность входного языка обеспечивается использованием языка SQL.

Его операторы непосредственно включаются в программы на языке С.

Возможно написание дополнительных программ, интерпретирующих SQL-запросы.

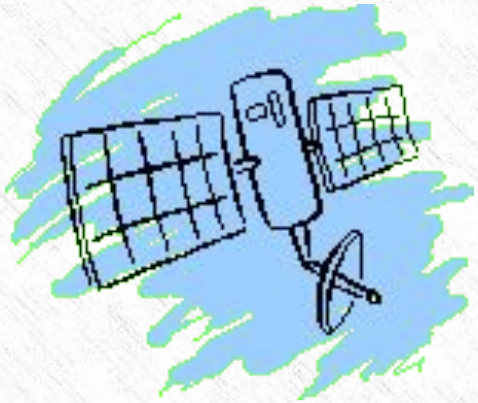
Отличительные особенности

СУБД третьего поколения:

- расширенный набор возможных типов данных (это абстрактные типы, массивы, множества, записи, композиции разных типов, отображение величин с значениями разных типов),
- открытость (доступность из разных языков программирования, возможность обращения к прикладным системам из СУБД),
- непроцедурность языка (общепринятым становится язык запросов SQL),
- управление асинхронными параллельными процессами, состояние которых отражает БД.

Управление асинхронными параллельными процессами, состояние которых отражает БД, позволяет говорить о тесной взаимосвязи СУБД и подсистемы управления проектами DesPM.

Названные особенности управления данными в САПР нашли свое выражение в современных подсистемах управления проектными данными PDM.



В PDM разнообразие типов проектных данных поддерживается их классификацией и соответствующим выделением групп с характерными множествами атрибутов.

Такими группами данных являются описания изделий с различных точек зрения (аспекты).

Для большинства САПР машиностроения характерными аспектами являются:

- свойства компонентов и сборок (эти сведения называют Bill of materials — BOM),
- модели и их документальное выражение (основными примерами могут служить чертежи, 3*D* модели визуализации, сеточные представления для конечно-элементого анализа, текстовые описания),
- структура изделий, отражающая взаимосвязи между компонентами и сборками и их описаниями в разных группах.

Вследствие большого объема проектных данных и наличия ряда версий проектов PDM должна обладать развитой системой поиска нужных данных по различным критериям.

Рассмотренные особенности банков данных в САПР позволяют квалифицировать их как системы Data Warehouse (DW), т.е. хранилища данных.

Для хранилищ данных *характерен* ряд особенностей:

1) длительное хранение информации, отражающей историю разработок;
2) частота операций чтения данных выше частоты операций обновления данных;
3) использование единых форматов для однотипных данных, полученных из различных источников (например, от разных программно-методических комплексов).

Эти особенности позволяют управлять конфигурацией проектов, что означает хранение в САПР всех версий проекта и данных по проектам предыдущих разработок, удовлетворение сложных запросов, для ответа на которые требуется извлечение и обработка данных из различных частей хранилища (многомерная обработка).

Модели данных в DW отличаются от реляционных моделей (RM).

В RM стремятся максимально уменьшить избыточность данных использованием нормальных форм, что приводит к увеличению числа таблиц, но уменьшенных размеров.

Однако многомерный поиск, требующийся в DW, в множестве таблиц затруднен.

Поэтому в DW чаще используется модель данных “звезда”, в которой имеется общая таблица фактов (Fact Table) и каждому факту ставится в соответствие несколько таблиц с необходимыми атрибутами.

Целостность данных в DW обеспечивается:

- ✔ проверкой и трансформацией данных (data cleaning), вводимых из внешних источников,
- ✔ наличием дисциплины обновления данных,
- ✔ централизованным хранением основной базы;

при этом достаточное быстродействие поддерживается

передачей копий определенных частей базы в локальные базы, называемые киосками данных (Data Mart) и ориентированные на отдельные группы пользователей.

Примером СУБД, учитывающей требования, предъявляемые со стороны САПР, является система IMAN фирмы EDS Unigraphics.

Это система управления объектно-ориентированными базами данных, ее можно также назвать системой интеграции данных.

Она выполняет функции подсистемы PDM, которые являются функциями хранения данных, управления доступом к ним, контроля вносимых изменений, создания спецификаций изделий, интегрирования прикладных подсистем.

Внутри IMAN используется реляционная модель данных, а на интерфейсном уровне — объектно-ориентированная информационная модель. Для синхронизации изменений предусматривается блокировка доступа пользователей, если с БД уже начал работу некоторый пользователь.

Другими *примерами* подсистем управления проектными данными могут служить системы

- Optegra (фирма Computervision),
- Euclid Design Manager (Matra Datavision),
- ProPDM в составе САПР Pro/Engineer (PTC),
- TechnoDOCS (Российская фирма “Весть”).

Ряд фирм разрабатывает системы PDM, которые можно использовать как самостоятельные продукты и как подсистемы в автоматизированных системах проектирования и управления.

Примером может служить система PartY (фирма Лоция Софт), в которой предусмотрены функции управления конфигурацией изделий, управления проектными данными и документооборотом, графический пользовательский интерфейс, реализация архитектуры клиент-сервер.

Интеллектуальные серверы БД

Особенности СУБД в САПР определяют их квалификацию
как *интеллектуальных*
(СУБД третьего поколения).

Признаки интеллектуальной СУБД (дополнительно к вышеуказанным):

- *реализация* в СУБД части прикладных процедур, что характерно для структуры DBS,
- *оповещение* пользователей (прикладных программ) об интересующих их изменениях состояния БД,
- *синхронизация событий* в БД,
- *способность* обслуживать прикладные программы, первоначально ориентированные на разные типы СУБД (*многопротокольность)*.

Оповещение заключается в информировании программы *A* о совершении события, вызванного программой *B* и влияющего на работу программы *A*.

Примером события может быть выход значения некоторого параметра в БД за допустимые пределы.

Наиболее просто ***информирование*** можно организовать периодическим опросом состояния БД со стороны *К*. Однако это усложняет ПО и неэффективно по затратам времени и загрузке сети.

Лучше возложить функцию оповещения на СУБД,
что и присуще интеллектуальным СУБД.

Но для этого нужно иметь обратные *ссылки* на программы, обращающиеся к БД, *правила* (*триггеры*), фиксирующие наступления событий, и *процедуры* обработки событий.

Удобный вариант ***оповещения*** — информирование программы *A* о происшедших событиях во время ее активизации.

Распределенные базы данных

В крупных АС,
построенных на основе корпоративных сетей, не всегда удается организовать централизованное размещение всех баз данных и СУБД на одном узле сети.

Поэтому появляются *распределенные базы данных*.

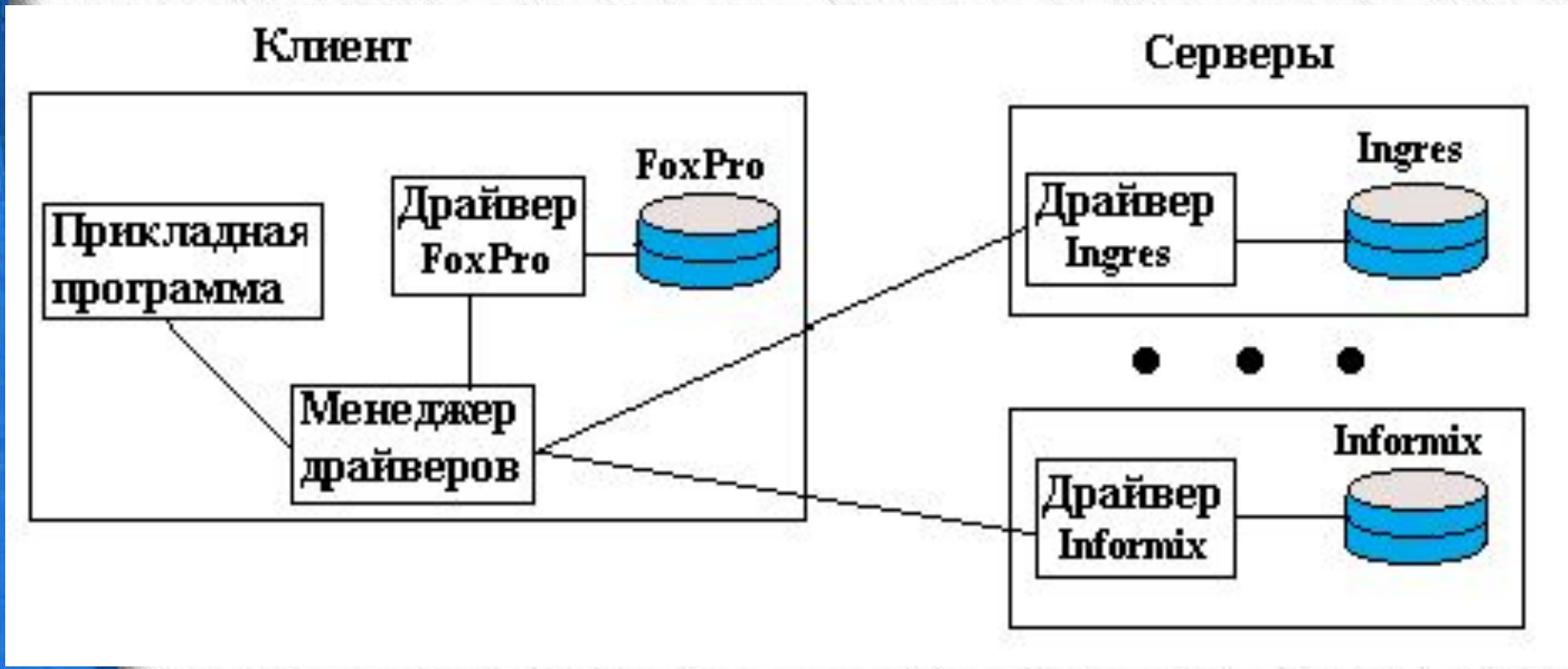
При построении РБД приходится решать ряд сложных проблем, связанных с *минимизацией* трафика, *обеспечением* интероперабельности *обработки* данных и *целостности* данных.

Минимизация трафика нужна в связи с необходимостью при обслуживании запроса данных из многих узлов, пересылаемые по сети.
Целесообразна однократная пересылка таблиц (причем таблиц именно меньшего размера) на один узел, на котором и будет обрабатываться запрос.

Интероперабельность выражает способность взаимодействия программ, работающих в гетерогенных сетях (в разных операционных средах или с разными СУБД).

Интероперабельность обеспечивается или с помощью программ-шлюзов (конверторов) для каждой пары взаимодействующих сред, или с помощью единого унифицированного языка взаимодействия.

Таким языком для доступа к БД является язык SQL, интероперабельность на его основе имеет место в системе ODBC (Open Data Base Connectivity), *пример* реализации которой показан на рисунке.



В примере СУБД FoxPro находится в локальном узле, а СУБД Ingres и Informix – в удаленных узлах. Прикладная программа имеет ODBC-интерфейс, не зависимый от особенностей различных СУБД. Менеджер драйверов реализует на базе унифицированного языка SQL все нюансы доступа к БД, общие для разных СУБД. Драйвер конкретной СУБД преобразует инвариантные к СУБД запросы в форму, принятую в данной СУБД. В трехзвенной структуре менеджер драйверов может быть размещен на промежуточном сервере.

Обеспечение целостности в РБД намного сложнее, чем в одноузловых БД. Различают два подхода ***к построению РБД***:

1) *тиражирование* (репликация), при котором на нескольких серверах (узлах) сети расположены копии БД;
2) *полномасштабная распределенность*, при которой разные части БД находятся на разных серверах сети (классическая распределенность).

Применяют два способа тиражирования.

Способ, называемый ***репликацией первой копии***, основан на выделении среди серверов с копиями БД одного первичного сервера (репликатора). Внесение изменений пользователями возможно только в БД первичного сервера, который в дальнейшем осуществляет тиражирование.

Тиражирование — это перенос изменений БД из первичного сервера во все вторичные (локальные) серверы, которые используются клиентами только для чтения данных.

Репликатор реагирует на события, фиксируемые триггерами, периодически пересылает обновленные данные в копии БД.

Недостаток способа — невысокая надежность, присущая любым централизованным структурам.

Надежность повышается при использовании ***способа голосования***.

Здесь изменения посылаются не в один первичный, а в некоторые *N* серверов. При этом любой запрос на чтение направляется к некоторым *M* серверам, причем $N+M > K$, где *K* — общее число серверов. Принимается последняя по времени обновления версия ответа.

Тиражирование вносит *избыточность* в хранимые данные, появляются *трудности* с разрешением конфликтов из-за возможных несогласованных изменений в локальных БД.

Однако по сравнению с классическими РБД, в которых данные не дублируются, *заметно уменьшается* трафик, *надежнее и проще работа* с локальными БД.

Обеспечение надежности и удобства работы особенно *актуально* в случае ненадежных и медленных каналов связи, что имеет место во многих сетях в России.

Спасибо за внимание!