УДК 004.65(07)

Ефремов Д.А., Мязин Е.С., Нечаев М.А.,
Романов А.В., Часовников А.С.

ИСПОЛЬЗОВАНИЕ ОБЪЕКТНО-ОРИЕНТИРОВАННЫХ СУБД В ПРАКТИКЕ ПРОЕКТИРОВАНИЯ ЭЛЕКТРОПРИВОДОВ

Возникновение баз данных (БД) обусловлено необходимостью накапливать большие объемы данных, сохраняя их во внешней памяти. Простейший способ – записать данные в файл и извлекать их по мере надобности – неприемлем, если мы имеем дело с большими объемами данных, для которых желательна сортировка и возможность обеспечивать доступ к ним нескольким пользователям.

На протяжении своей истории система управления базами данных (СУБД) постоянно эволюционировали [1, 2]. С 70-х годов наибольшее распространение получили реляционные модели данных, объекты в которых нормируются, то есть разбиваются на составные части и помещаются в реляционные таблицы, что уменьшает производительность, так как при каждом обращении к объекту его снова приходиться “собирать”. Несмотря на это они и сейчас играют главную роль на рынке СУБД. Однако многие пользователи и разработчики информационных программных продуктов выражают недовольство реляционной моделью. Она не удовлетворяет их требованиям по скорости доступа к данным и времени разработки программного обеспечения. Особенно это проявляется при проектировании систем с неструктурированными данными. Признав этот факт, крупнейшие разработчики СУБД начали встраивать в свои продукты поддержку объектно-ориентированного программирования, то есть возник новый объектно-реляционный подход. Собственно объектно-ориентированные базы данных (ООБД), появившись достаточно давно, наибольшее распространение получили только в настоящее время. Толчком для их развития послужило развитие объектно-ориентированных языков программирования и языков с произвольными типами данных. Наибольшее влияние на ООБД оказал язык Smalltalk.

Объектно-ориентированный подход традиционно базируется на следующих концепциях:

- объекта и идентификатора объекта;
- атрибутов и методов;
- классов;
- иерархии и наследования классов.

Каждый объект имеет состояние и поведение. *Состояние* – это набор значений его атрибутов. *Поведение* – это набор методов в виде программного кода, оперирующих состоянием объекта. Множество объектов, имеющих одинаковые атрибуты и методы, объединяются в *класс*. Допускается наличие предопределенных классов, объекты которых не имеют атрибутов. Допускается создание новых классов на основе существующих – подклассов. Также подкласс может создаваться на основе нескольких классов. Созданный подкласс наследует все атрибуты и методы класса. Некоторые объектно-ориентированные системы поддерживают *типы* (Simula, Vbase, O2), с помощью которых обобщаются общие черты множества объектов с одинаковыми характеристиками.

При создании первой общепризнанной реляционной модели данных использовался строгий математический аппарат. В случае объектно-ориентированной модели данных такого аппарата просто не существует. Бери предлагает в общих чертах формальную основу ООБД, которая возможно получит развитие. Кратко ее можно изложить следующим образом. Выделяется два уровня объекта: *структурный* и *поведенческий*. На структурном уровне поддерживаются сложные объекты. Наряду с понятием объекта поддерживается понятие значение. Для поведенческого уровня сформулирован только общий подход к логическому аппарату.

Для примера рассмотрим конкретную модель данных в ООБД O2, в которой поддерживаются объекты и значения. Объект – это пара (идентификатор, значение). Значение объектов доступны только через методы (процедуры). Значения могу быть атомарными и структурными. Структурные значения состоят из значений или объектов и создаются с помощью конструкторов множеств, кортежей и списков. Данные могут быть организованы в классы, экземплярами которых являются объекты и в типы, экземплярами которых являются значения. Каждому классу сопоставляется тип, описывающий структуру экземпляров класса.

Метод – программный код, привязанный к конкретному классу и применимый к объектам этого класса. Определение метода

походит в два этапа. На первом этапе необходимо объявить его имя, типы и классы результата и аргумента. Методы могут быть общими для всех классов (публичные) или доступными только внутри одного класса (приватными). На втором этапе определяется реализация класса на одном из языков программирования.

Языки программирования в ООБД во многом подобны традиционным объектно-ориентированным языкам. Различия в терминологии проявляются в дополнительной характеристике класса как *устойчивого* объекта, то есть обладающего способностью быть сохраненным и инициализированным. Вообще, объектные СУБД, как показано на рис. 1, реализуют как возможности баз данных (целостность данных безопасность, версионность, стабильность, параллелизм, транзакция, запросы, восстанавливаемость, архивирование, запросы), так и возможности объектно-ориентированных языков программирования (наследование, полиморфизм, инкапсуляция данных, идентификация объектов).

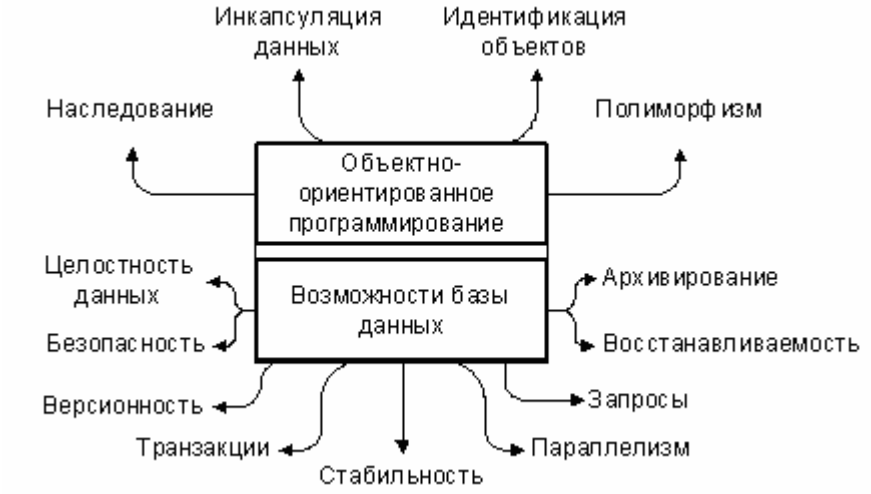

Рис. 1. Объединение возможностей СУБД и объектно-ориентированного программирования

Поясним значения некоторых терминов.

Идентификация объектов – это присвоение объекту уникального идентификатора, таким образом, каждый объект в СУБД уникален. Обычно уникальный идентификатор невидим, его "знает" только СУБД, но он не изменяется ни при каких обстоятельствах, несмотря на модификацию объекта вплоть до его удаления.

Стабильность – это свойство СУБД сохранять данные между сеансами.

Параллелизм – система должна обеспечить бесконфликтное существование пользователей, одновременно работающих с базой данных.

Восстанавливаемость – после программных или аппаратных сбоев система должна возвращаться к первоначальному состоянию данных.

Транзакция – механизм, обеспечивающий фиксацию базы данных до начала критической операции и последующее возвращение к этому состоянию.

Безопасность данных – одно из ключевых понятий в мире СУБД, поскольку часто в базах хранится конфиденциальная, а подчас и секретная информация.

Целостность данных – имеется в виду как структурная, так и логическая целостность базы данных. Кроме того, для объектных СУБД важно поддержание соответствия между объектами в базе и объектами, переданными клиенту.

Запросы – для любой серьезной базы данных обязателен механизм выполнения незапланированных запросов. Причем требуется такой язык запросов, который бы не требовал знания информации о том, как найти объекты в базе данных. В качестве языка запросов широко используются различные реализации OQL – Object Query Language, объектного языка запросов. Это, как правило, расширение SQL, дополненное объектными свойствами, средствами описания типов данных, итерации по объектам в СУБД.

Архивирование – поскольку в базах могут накапливаться огромные объемы информации, надо максимально задействовать средства программного сжатия данных.

Версионность – поддержка на уровне СУБД многих версий одного объекта. Например, данные объекта могут многократно изменяться, а в базе храниться все версии объекта.

Особенности конкретных СУБД, поддерживающих объектно-ориентированный подход, показаны в табл. 1.

Таблица 1.

Компания-разработчик: Objectivity/DB. **Название СУБД:** Objectivity 5.0. **Работа со стабильными объектами:** Набор предопределенных классов, от которых наследуется пользовательский класс. Доступ к объекту через идентификатор, причем, для записи модификации объекта необходимо явно уведомить СУБД. **Поддерживаемые языки:** C++, Java, Smalltalk. **Платформы:** DEC, Sun4/SPARC, VAX, HP 9000, IBM RISC System/6000, Silicon Graphics, NCR System.
Компания-разработчик: O2 Technology **Название СУБД, версия:** O2, 5.0 **Работа со стабильными объектами:** Объект сразу создается стабильным. Доступ к объектам производится через указатель. **Поддерживаемые языки:** C++, Java, Smalltalk. **Платформы:** Нет данных.
Компания-разработчик: ONTOS, Inc. **Название СУБД:** ONTOS DB 2.5. **Работа со стабильными объектами:** Создание стабильного объекта производится наследованием от базового объекта Object. Чтение из базы обеспечивается набором функций, запись измененного объекта производится явно. **Поддерживаемые языки:** C++, VisualBasic, Java. **Платформы:** IBM RISC System/6000, IBM PC, HP 9000, SCO 386, Sun4/SPARC.
Компания-разработчик: Versant Object Technology. **Название СУБД:** Versant, Release 5. **Работа со стабильными объектами:** Создание стабильного объекта производится наследованием от базового объекта PObject. Доступ к объекту через указатель, запись обновленных объектов в базу данных указывается в программе. **Поддерживаемые языки:** C++, Java, Smalltalk. **Платформы:** Sun4/SPARC, IBM RISC System/6000, HP 9000, DEC, Sequent, IBM PC, Silicon Graphics, NeXT.

Продолжение таблицы 1.

Компания-разработчик: Computer Associates , Inc. **Название СУБД:** Jasmine, 1.1. **Работа со стабильными объектами:** Любой объект, созданный в среде разработки JADE (Jsamine Application Developer Environment) считается стабильным. **Поддерживаемые языки:** C++, C, Java, VisualBasic. **Платформы:** Sun Solaris, HP-UX, DEC, IBM RISC System/6000, Windows NT 4.0.
Компания-разработчик: ObjectDesign, Inc. **Название СУБД:** ObjectStore, 5.0. **Работа со стабильными объектами:** Создание стабильного объекта производится после вызова специального оператора new. Доступ к объектам через специальные точки входа в БД. Запись изменений в базу производится автоматически. **Поддерживаемые языки:** C++, Java. **Платформы:** Sun, HP, DEC, NCR, Univel, Olivetti, IBM RISC System/6000, Silicon Graphics, IBM PC.
Компания-разработчик: Gemstone, Inc. **Название СУБД:** Gemstone, 5.0. **Работа со стабильными объектами:** Создание стабильного объекта производится наследованием от базового класса GS_Object и в дальнейшем все ссылки на объект производятся через указатель. **Поддерживаемые языки:** Smalltalk, Java. **Платформы:** Sun4/SPARC, IBM RISC System/6000, HP 9000, DEC, Sequent.
Компания-разработчик: POET Software GmbH. **Название СУБД:** POET, 5.0. **Работа со стабильными объектами:** Создание стабильного объекта производится добавлением в его описание ключевого слова persistent. Доступ к объектам через указатель. **Поддерживаемые языки:** C++, Java, VisualBasic. **Платформы:** Windows NT 3.51, Os/2 Warp, Novell Netware, Macintosh 68K, PowerPC, Sun Solaris, HP-UX, IBM AIX, SCO.

Продолжение таблицы 1.

Компания-разработчик: Ibex Computing, Inc. **Название СУБД:** Itasca, 5.0. **Работа со стабильными объектами:** Любой объект программы считается стабильным. **Поддерживаемые языки:** Lisp, CLOS, C++, C. **Платформы:** HP-UX, Sun Solaris,, Windows NT 4.0, Digital UNIX, SGI, AIX, клиентская часть в Windows 95, Windows NT.
Компания-разработчик: UniSQL, Inc. **Название СУБД:** UniSQ, 3.2. **Работа со стабильными объектами:** Нет данных. **Поддерживаемые языки:** ANSI C, Smalltalk, Microsoft Visual C. **Платформы:** Windows 95, Windows NT.
Компания-разработчик: НПЦ "Интелтек Плюс". **Название СУБД:** ODB-Jupiter, 2.1. **Работа со стабильными объектами:** Создание стабильного объекта производится наследованием от базового объекта TOdbObject и реализацией методов для обмена с базой данных. **Поддерживаемые языки:** C++. **Платформы:** Windows, Windows 95, Windows NT 4.0.

Традиционной областью применения объектных СУБД являются системы автоматизированного проектирования (САПР) и моделирования [1]. Действительно, для таких систем актуален вопрос хранения отдельных объектов, имеющих собственное поведение и обширный набор неструктурированных параметров.

Рассматривая возможность использования объектных СУБД в автоматизированном проектировании электроприводов в рамках системы САПР ЭП [3] можно сказать, что это, несомненно, перспективное направление. Но на данный момент существует достаточно большое количество сдерживающих факторов.

1. На данный момент данные элементов электропривода и основополагающая информация о ходе проектирования хранится в файлах, использующих структуру INI-файла. Это не очень удобно, но данная технология уже реализована и апробирована, поэтому насущной необходимости в специализированных данных-объектах нет.

2. Используемая реляционная СУБД на данный момент удовлетворяет потребностям САПР. Быстродействие обеспечивается локальным расположением файлов БД. Конечно, существуют определенные трудности, например, для определения данных, необходимых для проектирования, но отсутствующих в справочной литературе. В этих случаях они определяются расчетным путем на уровне объектов непосредственно в софт-компонентах САПР.

3. Необходим опыт работы в среде разработки ООБД. Без этого невозможно окончательно решить вопрос с ее использованием.

4. Не все объектно-ориентированные СУБД поддерживают платформу Windows XP (табл. 1), а САПР ЭП функционирует именно в этой операционной системе. Принятая технология разработки делает задачу переноса с сохранением всех функциональных возможностей разработанного программного обеспечения в другую среду очень трудоемкой.

5. Принятый основной язык программирования для разработки софт-компонент САПР ЭП – Object Pascal, интегрированная среда разработки – Delphi 7. Исходя из перечня поддерживаемых языков (табл. 1), очевидно, что придется использовать или СИ++, или VisualBasic, или Java. Естественно, что это также создаст определенные трудности, и будет препятствовать развитию САПР.

ЛИТЕРАТУРА

1. http://www.inteltec.ru/publish/articles/objtech/oodbms_o.shtml / Объектно-ориентированные базы данных: среда разработки программ плюс хранилище объектов. А.М. Андреев, Д.В. Березкин, Ю.А. Кантонистов.

3. http://cs.stu.cn.ua/~vik/semestr/manifodb.html / Манифест систем объектно-ориентированных баз данных. М. Аткинсон, Ф. Бансилон, Д. ДеВитт, К. Диттрих, Д. Майер, С. Здоник.

4. Романов А.В., Бурковский В.Л., Фролов Ю.М. Проблематика автоматизированного проектирования электроприводов. / Вестник ВГТУ. Серия "Вычислительные и информационно-телекоммуникационные системы". Том 2. № 5. Воронеж: ВГТУ, 2006. 148 с. С. 119 – 123.

Аннотация

В материале рассмотрены основные понятия объектно-ориентированного подхода к созданию баз данных, используемые языки программирования, особенности отдельных объектно-ориентированных баз данных. Проанализирована возможность и необходимость использования подобного подхода в системе автоматизированного проектирования электроприводов.