

УДК 004.6

**ЭВОЛЮЦИЯ АРХИТЕКТУРЫ ИНФОРМАЦИОННЫХ СИСТЕМ**

В.В. Дрождин, к.т.н.; Р.Е. Зинченко

(Пензенский государственный педагогический университет им. В.Г. Белинского, drozhdin@yandex.ru, rzinchenko@yandex.ru)

Рассматривается развитие архитектуры информационных систем от систем с организацией данных в виде файлов до самоорганизующихся информационных систем. Приведены требования к основным компонентам информационных систем, отмечены преимущества и недостатки информационных систем для каждого типа архитектуры. Показана целесообразность взаимодействия самоорганизующейся информационной системы с системой иерархически организованных пользователей.

**Ключевые слова:** информационная система, самоорганизующаяся информационная система, архитектура информационной системы, концептуальная модель предметной области, база данных.

История автоматизированных информационных систем (АИС) и БД в их современном понимании началась с середины 1950-х гг. с появлением ПО, поддерживающего модель обработки записей на основе файлов [1, 2]. В этих системах БД для каждого приложения представляла собой независимый набор файлов.

Архитектура таких систем (рис. 1) определяла, что файлами должен управлять вручную непосредственно разработчик программной системы. Основными проблемами АИС с файловой системой являются следующие [3].

1. Сложный доступ к данным вследствие их размещения в отдельных файлах. Обработка данных в АИС такого типа очень трудоемка, так как для извлечения необходимой информации требуется синхронная обработка соответствующего количества файлов.

2. Многократное дублирование данных из-за их децентрализованного размещения и обработки, что приводит к противоречиям в системе и требует дополнительных затрат (времени, финансов, человеческого труда и др.) на ввод, хранение и обработку избыточных данных.

3. Большая зависимость программ от данных, так как физическая структура данных и способ хранения записей в файлах жестко зафиксированы в программном коде приложения. Изменение су-

ществующей структуры данных практически невозможно из-за очень высокой трудоемкости изменения программного кода и может вызвать появление ошибок.

4. Несовместимость форматов файлов, созданных программами на разных языках программирования, так как структура файлов определялась программным кодом приложения.

5. Очень высокая трудоемкость создания ПО вследствие обработки запросов и формирования отчетов (ответов) на низком уровне обработки данных. Это приводило к сокращению возможных типов запросов и увеличению количества файлов и приложений, что существенно снижало адекватность АИС информационным потребностям пользователей.

6. Существенные проблемы с поддержкой целостности и обеспечением безопасности, так как средства восстановления системы после сбоя программного или аппаратного обеспечения были крайне ограничены или вообще отсутствовали.

Использование внешних дисковых устройств с произвольным доступом потребовало разработки специализированных подпрограмм для манипулирования данными (что явилось одним из первых шагов на пути к созданию систем управления данными), однако от разработчиков требовалось знание физической организации данных (организации данных низкого уровня). Это незначительно снизило трудоемкость разработки ПО [4, 5].

Указанные проблемы создания и сопровождения АИС с файловыми системами обусловили появление в конце 1960-х годов СУБД, реализующих единый метод доступа и обработки данных. Это позволило исключить из прикладных программ определение организации данных и их низкоуровневую обработку в БД. Однако в специализированных АИС файловые системы применяются до настоящего времени.

Эра информационных систем с использованием СУБД началась в середине 1960-х гг., когда корпорация IBM совместно с фирмой NAA (North American Aviation, в настоящее время – Rockwell International) разработали СУБД IMS (Information Management System), поддерживающую БД в рам-

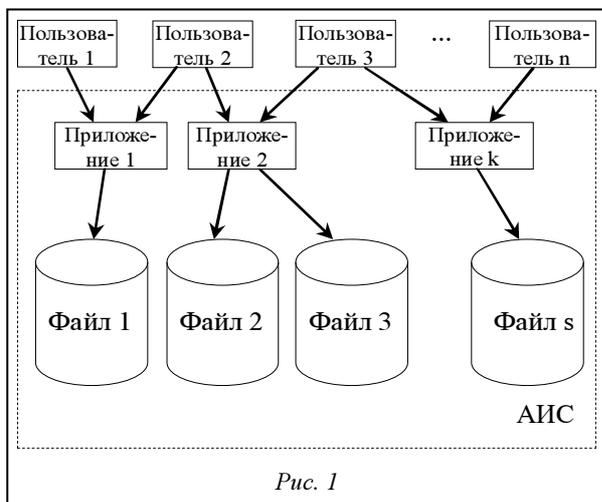


Рис. 1

как иерархической модели данных. *IMS* является самой первой коммерческой СУБД и до сих пор остается основной иерархической СУБД, используемой на большинстве крупных мейнфреймов.

Другим заметным достижением середины 60-х годов было появление СУБД *IDS (Integrated Data Store)* фирмы *General Electric*, поддерживающей БД в рамках сетевой модели данных. СУБД *IDS* создавалась для представления и обработки данных с более сложными взаимосвязями, чем в иерархической модели данных, и послужила основой для разработки первых стандартов БД. Для создания таких стандартов в 1965 г. на конференции *CODASYL (Conference on Data Systems Languages)* была сформирована рабочая группа *List Processing Task Force*, переименованная в 1967 г. в группу *Data Base Task Group (DBTG)*, в компетенцию которой входило определение спецификаций среды, допускающей разработку БД и управление данными.

Однако *DBTG*-системы имели существенный недостаток: даже для выполнения простых запросов с использованием переходов и доступом к определенным записям необходимо было создавать достаточно сложные программы, что обеспечивало их слабую независимость от данных [6].

Для его преодоления в середине 1970-х гг. комитет *ANSI/X3/SPARC* разработал архитектуру АИС с трехуровневой организацией данных, в которой концептуальный уровень является единой, логически непротиворечивой информационной моделью предметной области (ПрО) (архитектура *ANSI-SPARC* приведена на рисунке 2) [7, 8].

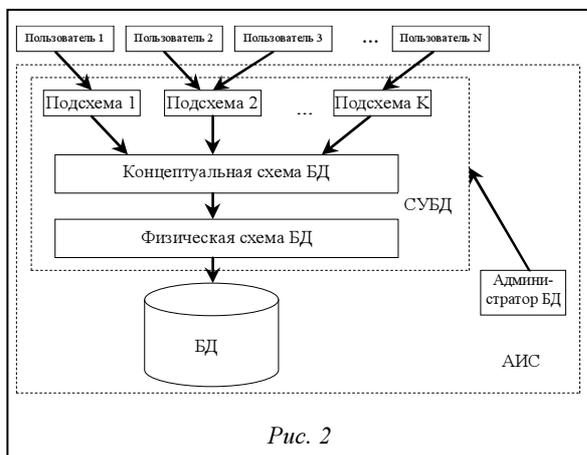


Рис. 2

АИС с такой архитектурой способны поддерживать предметные БД, являющиеся универсальным хранилищем данных, не зависящим от решаемых задач и ориентированным на точное и полное описание ПрО. Достоинствами предметных БД являются их стабильность и простота использования данных внешними системами, минимальная избыточность данных, минимальные затраты на ведение БД и синхронность изменения данных для всех пользователей. К недостаткам

относится необходимость создания службы администратора БД, обеспечение физической и логической целостности БД, защита БД от несанкционированного доступа, необходимость разрешения коллизий при одновременном обращении нескольких пользователей к одним и тем же данным, а также необходимость проектирования БД.

Процесс проектирования АИС с данной архитектурой заключался в разработке концептуальной модели ПрО и ее реализации в виде подсхем пользователей и концептуальной схемы БД в рамках некоторой даталогической модели данных, поддерживаемой СУБД. Однако оказалось, что концептуальная модель ПрО шире, чем схема БД, так как отражает не только объектную структуру ПрО, но и систему бизнес-процессов, протекающих на ее основе.

Поэтому архитектура АИС была усовершенствована путем замены пользовательского уровня приложением, поддерживающим представления пользователей и бизнес-процессы ПрО, и переходом к двухуровневой организации данных, поддерживаемой СУБД [9]. Эта архитектура практически без изменений существует до настоящего времени (рис. 3).

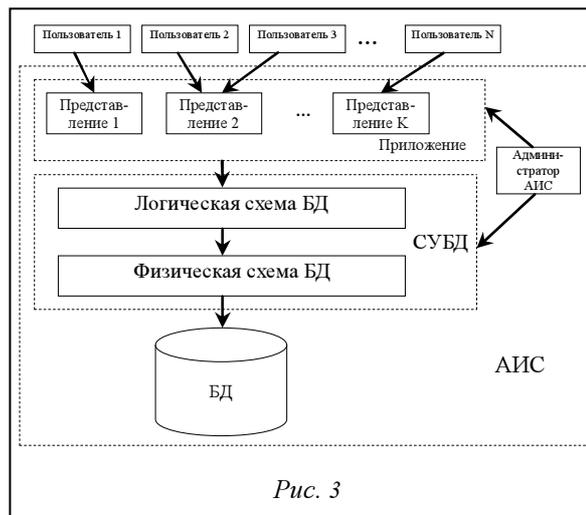


Рис. 3

Наличие приложения делает АИС проблемно-ориентированной системой, что обеспечивает ей достаточно высокую адекватность ПрО, однако это требует сложного процесса проектирования АИС и разработки приложения. Причем концептуальное моделирование ПрО часто осуществляется с использованием *CASE*-систем, не связанных непосредственно с создаваемой АИС, а при изменении ПрО необходимо перепроектирование АИС.

Реализация уровня приложения может быть выполнена по различным технологиям (клиент-серверной, трехзвенной и др.), но это принципиально не влияет на архитектуру АИС [6, 10].

Таким образом, создание современных АИС предполагает наличие большого и сложного этапа

проектирования системы, а формализованная концептуальная модель ПрО не поддерживается создаваемой АИС, а представляет собой однократно созданный самостоятельный объект (документ, диаграмму и т.д.).

Для снижения сложности и трудоемкости проектирования и создания АИС целесообразно возложить существенную часть данного процесса непосредственно на пользователей, предоставив им средства концептуального моделирования ПрО. При этом система должна обладать способностью создавать и поддерживать БД, адекватную концептуальной модели ПрО.

Поэтому предлагается изменить архитектуру АИС таким образом, чтобы она непосредственно поддерживала концептуальную модель ПрО и автоматически обеспечивала соответствие этой модели логической схеме БД (рис. 4).

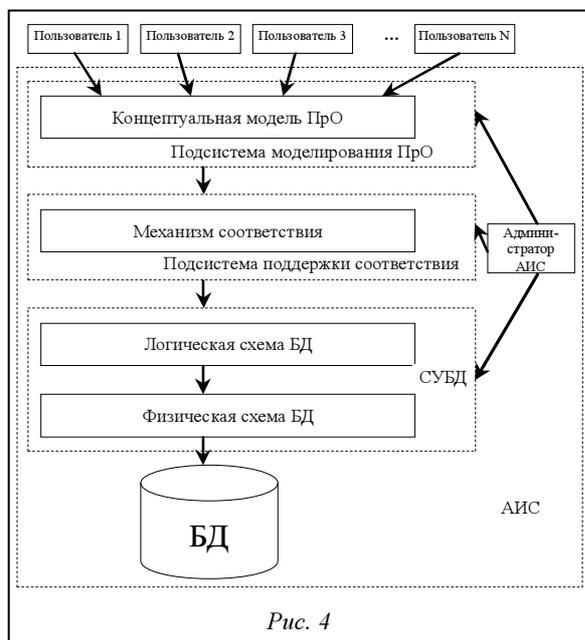


Рис. 4

Наличие концептуальной модели ПрО непосредственно в АИС имеет следующие преимущества:

- обеспечение максимальной адекватности АИС ПрО;
- существенное участие пользователей АИС в ее создании;
- взаимодействие пользователей с АИС на семантическом уровне;
- обеспечение автоматической поддержки соответствия концептуальной модели ПрО и схемы БД, позволяющее без привлечения разработчиков модифицировать АИС в широких пределах при изменении ПрО или организации данных в процессе функционирования системы;
- многократное увеличение срока эксплуатации АИС.

Авторы предлагают принципиально новую архитектуру АИС с независимой эволюцией кон-

цептуальной модели ПрО и БД (рис. 5). Независимость эволюции модели ПрО и БД основана на методе системно-изоморфного динамического соответствия концептуальной модели ПрО и схемы БД [11]. При этом предполагается, что БД и СУБД реализованы по наиболее распространенной в настоящее время реляционной технологии организации и обработки данных.

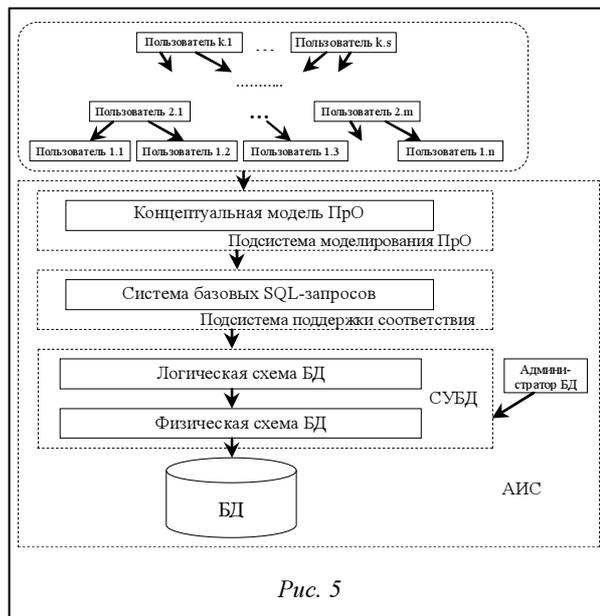


Рис. 5

В предлагаемой архитектуре АИС пользователи представляются не множеством изолированных однородных объектов, а системой объектов, организованных преимущественно в сетевую структуру. Наличие иерархических отношений между пользователями позволяет распределять обязанности администрирования системы среди достаточно большого числа иерархически привилегированных пользователей, что приведет к существенному снижению затрат на эксплуатацию АИС [12]. Именно этим объясняется отсутствие администратора системы на уровне концептуального моделирования ПрО, хотя администратор БД остается для выполнения технических работ по поддержанию аппаратного и программного обеспечения организации и обработки данных.

В соответствии с новой архитектурой АИС пользователи получают возможность формировать собственное представление о ПрО и решаемых задачах в форме концептуальной модели ПрО и формулируют информационные потребности путем оперирования понятиями, а проектирование и ведение БД, соответствующей модели ПрО, осуществляется самой АИС [13].

При этом пользователи получают возможность формировать модель ПрО, начав с очень простого представления реального мира, включающего только наиболее существенные объекты, их свойства, состояния и отношения между ними, а АИС сразу же создает рабочую версию системы,

соответствующую этому представлению. Пользователь, освоив работу с такой системой, будет совершенствовать модель ПрО, уточняя существующие и включая в нее новые объекты, изменяя систему отношений между ними, а система, соответственно, будет изменять БД и методы ее обработки.

Таким образом, несмотря на различные пользовательские представления ПрО и изменяющееся в процессе функционирования системы представление ПрО конкретного пользователя, АИС всегда будет способна поддерживать БД, адекватную полному представлению ПрО всех пользователей.

Взаимодействие АИС с внешней средой, представленной системой иерархически связанных пользователей, дает возможность осуществлять управление и контроль вышестоящими пользователями деятельности подчиненных пользователей. При возникновении сложных ситуаций в процессе взаимодействия с конкретным пользователем АИС всегда знает, к кому ей необходимо обратиться за разъяснениями и разрешением исполнения соответствующих действий конкретного пользователя, может регулярно или по требованию вышестоящих пользователей информировать их о деятельности подчиненных пользователей.

Таким образом, между АИС и системой пользователей будет осуществляться более интенсивное взаимодействие, чем с существующими системами, что создаст реальные условия для возникновения системы принципиально нового типа, являющейся суперсистемой и включающей АИС и систему пользователей в качестве равноправных взаимодействующих подсистем.

Возможность формирования пользователями собственной системы понятий и наличие иерархических отношений между пользователями позволяют естественным образом реализовать механизм областей видимости для каждого пользователя АИС, что существенно повышает защищенность системы без применения специальных средств защиты, а приобретение АИС свойств темпоральной системы делает ее очень надежной. Эти свойства АИС обеспечивают широкие возможности их использования в управлении предприятиями (организациями).

АИС с новой архитектурой, обладая большими возможностями по адаптации к потребностям пользователей, не ограничивает область деятельности как отдельных пользователей, так и предприятия в целом, что позволяет использовать такие АИС на предприятиях любого типа. Более того, построение системы управления предприятием на основе АИС с данной архитектурой запускает процесс коэволюции АИС и предприятия, существенно ускоряющий их совместное развитие [14]. Это позволит предприятию заранее предвидеть возникновение различных ситуаций как внутри

предприятия, так и во внешнем мире и своевременно адекватно реагировать на них, что обеспечит устойчивое развитие и более длительное существование предприятия, а также высокую эффективность и продолжительную эксплуатацию АИС.

Однако в отличие от эволюции концептуальной модели ПрО механизм системно-изоморфного динамического соответствия концептуальной модели ПрО и схемы БД предъявляет более высокие требования к эволюционной организации данных.

Это связано с тем, что организация БД в рамках реляционной модели данных является пассивной системой и не позволяет механизму соответствия функционировать с максимальной эффективностью и поддерживать БД, наиболее адекватную концептуальной модели ПрО и требуемой обработке данных. Например, снижение адекватности происходит при включении в концептуальную модель ПрО нового понятия-класса для существующих понятий-подклассов и включении нового конкретного понятия для существующего понятия-образа, а снижение эффективности обработки данных наблюдается при росте объема БД и усложнении логической организации данных.

Поэтому дальнейшее развитие архитектуры АИС должно происходить путем совершенствования организации БД, которую предлагается создавать в рамках конструктивной эволюционной модели данных [15, 16] (рис. 6).

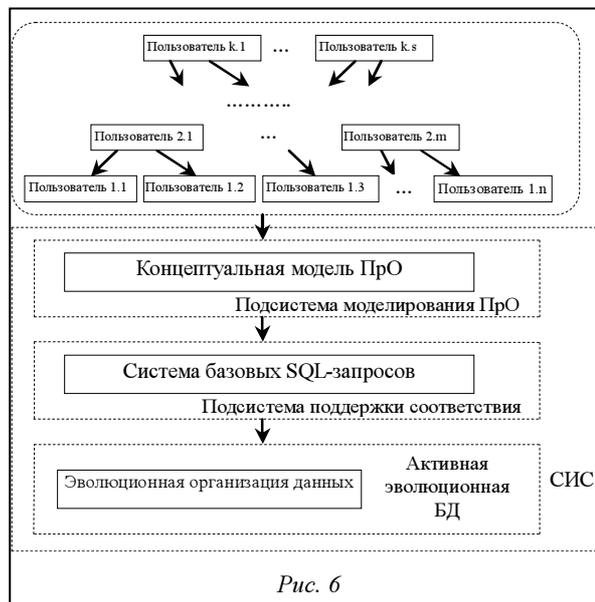


Рис. 6

Эволюционная БД имеет следующие преимущества.

– БД организуется в виде системы взаимодействующих программно-информационных компонентов, самостоятельно осуществляющих организацию и обработку данных на основе собственных знаний и методов обработки данных. Это

позволит снять задачу организации данных с механизма соответствия концептуальной модели ПрО и схемы БД, а алгоритмы ведения данных существенно упростить.

– Отображение понятий концептуальной модели ПрО в БД преимущественно будет осуществляться операцией проекции, что приведет к еще большему упрощению этого механизма без снижения его функциональных возможностей.

– Активность эволюционной БД, с одной стороны, будет устранять снижение адекватности БД концептуальной модели ПрО, а с другой – вследствие ориентации на требуемую обработку данных будет многократно повышать надежность и эффективность функционирования системы.

– Способность эволюционной БД к выявлению логических закономерностей (в том числе функциональных зависимостей между данными) обеспечит формирование и ведение БД, более точно соответствующей предметной области, чем концептуальная модель ПрО, создаваемая пользователями. Это позволит осуществлять контроль за деятельностью пользователей и обеспечивать очень высокую надежность функционирования системы.

– Выявление эволюционной БД различных конструктивных закономерностей (например, устойчивых иерархических зависимостей) обеспечит максимально эффективную организацию данных при текущем уровне знаний системы, что еще больше будет способствовать повышению эффективности обработки данных.

– Взаимодействие информационных компонентов эволюционной БД с концептуальной моделью ПрО и между собой с помощью реляционных отношений в первой нормальной форме обеспечивает высокую универсальность и устойчивость взаимодействия при изменении внутренней организации и обработки данных в БД в широких пределах.

– Эволюционная БД способна самостоятельно формироваться и функционировать без проектирования и разработки.

Таким образом, архитектура информационной системы (рис. 6) вследствие полного автоматизма всех ее компонентов является архитектурой *самоорганизующихся информационных систем* (СИС).

Функционирование СИС заключается в формировании и постоянном совершенствовании организации данных, соответствующей концептуальной модели ПрО, выступающей для системы в качестве неспецифических (неуправляющих) воздействий, а минимальные затраты, максимальная адекватность, высокая эффективность, живучесть и предоставление более высокой полезной функциональности по сравнению с существующими

АИС будут интенсифицировать взаимодействие системы пользователей с СИС, формируя систему более высокого иерархического уровня, в которой процесс коэволюции будет многократно ускорять оптимизацию и развитие каждой подсистемы и всей системы в целом. Это приведет к существенным организационным преимуществам предприятий, использующих СИС, их большей адекватности внешней среде и, следовательно, к их более высокой конкурентоспособности и выживанию в изменяющихся условиях.

### Литература

1. Gray J. Data Management: Past, Present, and Future // IEEE Computer 29(10). 1996. Vol. 29, No. 10, pp. 38–46.
2. Haigh T. How Data Got its Base: Information Storage Software in the 1950s and 1960s // IEEE Annals of the History of Computing. 2009, No. 4.
3. Connolly T.M., Begg C.E. Database Systems: A Practical Approach to Design, Implementation and Management. Addison Wesley, 2009, p. 1400.
4. Elmasri R., Navath Sh.B. Fundamentals of Database Systems. Addison Wesley, 2010, p. 1200.
5. Singh P.K. Database Management System Concepts. V.K. (India) Enterprises, 2009.
6. История развития СУБД: матер. с сервера дистанционного обучения Бийского технолог. ин-та. URL: [http://do.bti.secna.ru/lib/book\\_it/istor Razv.html](http://do.bti.secna.ru/lib/book_it/istor Razv.html) (дата обращения: 24.06.2010).
7. ANSI/X3/SPARC Study Group on Data Base Management Systems: Interim Report. FDT, ACM SIGMOD bulletin. 1975. Vol. 7, No. 2.
8. Jardine D.A. The ANSI/SPARC DBMS Model. North-Holland Pub. Co. 1977.
9. Härder T. DBMS Architecture – Still an Open Problem // In Datenbanksysteme in Business, Technologie und Web. Gesellschaft für Informatik (GI), 2005, pp. 2–28.
10. Vasconcelos A., Sousa P., Tribolet J. Information System Architecture Evaluation: From Software to Enterprise Level Approaches // 12th European Conference On Information Technology Evaluation (ECITE 2005), Turku, Finland, September 2005.
11. Зинченко Р.Е. Системно-изоморфное динамическое соответствие концептуальной модели предметной области и схемы базы данных // Программные продукты и системы. 2010. № 1. С. 71–75.
12. Дрождин В.В., Зинченко Р.Е. Системный подход к концептуальному моделированию предметной области в самоорганизующейся информационной системе // Программные продукты и системы. 2009. № 4. С. 73–79.
13. Дрождин В.В., Зинченко Р.Е. Информатизация предприятия на основе самоорганизующейся информационной системы // Инновации в управлении и образовании: технико-технологические и методические аспекты: матер. II Междунар. науч.-практ. конф. Тула, 2009. С. 91–93.
14. Князева Е.Н., Курдюмов С.П. Синергетика. Нелинейность времени и ландшафты коэволюции. М.: КомКнига, 2007. 272 с.
15. Дрождин В.В. Системный подход к построению модели данных эволюционных баз данных // Программные продукты и системы. 2007. № 3. С. 52–55.
16. Дрождин В.В. Открытость структур в эволюционной модели данных // Программные продукты и системы. 2009. № 2. С. 135–137.