Дрождин В.В., Володин В.М. Автономный компонент организации данных. // Проблемы информатики в образовании, управлении, экономике и технике: Сб. стат. VIII Всерос. науч.-техн. конф.– Пенза, 2008.–С. 7–14.

АВТОНОМНЫЙ КОМПОНЕНТ ОРГАНИЗАЦИИ ДАННЫХ

В.В. Дрождин, А.М. Володин

Пензенский государственный педагогический университет
им. В.Г. Белинского,
г. Пенза

Организация и обработка больших объемов информации на современном этапе является очень важной проблемой. Поэтому разработка специализированных компонентов, учитывающих особенности конкретных данных и методов их обработки, является актуальной задачей.

Данные – это информация, представленная в формализованной форме, пригодной для последующей обработки, хранения и передачи.

Структура данных – это взаимосвязанная совокупность элементов данных, между которыми существуют структурные отношения. Элементами данных могут быть как простые данные, так и сложные структуры данных.

Формально структуру данных можно представить в виде:

$S = <D, R, Z>$,

где D – множество элементарных данных; R – множество отношений между элементами данных; Z – ограничения на структуру данных (множество законов композиции).

В качестве примера **элементарных данных D** можно привести элементы массива, хранящие значения данных простого типа (скаляры), а также элементы связного списка, содержащие значения данных и указатели на другие элементы.

Структурные отношения R задают устойчивые (структурные) взаимосвязи между элементами и отражают упорядоченность элементов (отношение непосредственного следования, строгого и нестрогого порядка), направления переходов от одних элементов к другим, отношение иерархии и др. Операции добавления новых элементов, изменения или удаления существующих элементов могут приводить к нарушению этих отношений. Поэтому необходимо либо вызыват методы, восстанавливающие структурные отношения (путем реструктуризации данных), либо запрещаются любые операции, нарушающие отношения.

Законы композиции Z – это ограничения на количество элементов, тип и число связей между элементами структуры, уникальность (наличие или отсутствие дубликатов) и иные логические ограничения. Так, структура «АВЛ-дерево» требует, чтобы для каждого узла высота его правого поддерева отличалась от высоты левого не более чем на единицу. Операции обработки данных также могут приводить к нарушению этих ограничений.

Различают абстрактные и физические структуры данных. **Абстрактные (логические) структуры данных** ориентированы на адекватное отражение реального мира. Эти структуры носят абстрактный характер (не учитывается их реализация на компьютере), хорошо формализованы и, как правило, имеют точное математическое описание.

Физические структуры данных отражают не только логические взаимосвязи между данными, но также определяют способ представления данных в памяти компьютера и реализацию методов их обработки. Поэтому физические структуры данных ориентированы прежде всего на эффективную обработку данных.

Вследствие различия между логическими и физическими структурами данных необходимо наличие методов, позволяющих отображать логические структуры в физические и наоборот.

Фундаментальными физическими структурами данных являются структуры данных, реализованные в языках программирования. Причем из простых структур могут строиться структуры с более сложной организацией данных. К статическим структурам относятся массивы, строки, записи (структуры) и др. Динамические структуры формируются путем использования указателей, позволяющих строить связные списки с различной структурой.

Физические структуры во внешней памяти представлены файлами с различной внутренней организацией данных. К ним относятся файлы данных с простой структурой (содержат записи фиксированной длины), файлы со сложной внутренней организацией (содержат записи разной структуры, могут быть разделены на блоки одинакового или разного размера). По способу организации записей различают файлы с физически смежной организацией данных и связные списки.

При разработке программного обеспечения большую роль играет выбор структур данных. От этого часто зависят сложность реализации и эффективность исполнения программ. Различные виды структур данных имеют свои особенности и подходят для различных приложений. Некоторые из них имеют узкую специализацию для определённых задач. Например, Б-деревья наиболее часто применяются для баз данных, в то время как хеш-таблицы используются повсеместно для создания различного рода словарей, например, для отображения доменных имён в Интернет адреса компьютеров. При выборе структуры данных необходимо ориентироваться на эффективное выполнение тех операций, которые применяются с высокой интенсивностью к подавляющему большинству элементов структуры. Кроме этого, необходимо учитывать такие факторы, как надежность, изменчивость, независимость данных, минимизацию избыточности, способность к восстановлению информации после сбоев и др.

Многие известные структуры данных представлены в стандартных библиотеках языков программирования. Широко используются стандартная библиотека шаблонов STL языка C++, Java Collection Framework, Apache Common Collection, Google Collection, написанные на Java, и др.

Классические подходы к организации данных основываются на предположении, что доступ ко всем элементам структуры данных является равновероятным. Однако часто такое предположение оказывается неверным. Так, одна из операций обработки данных (например, поиск) может применяться с разной интенсивностью к различным элементам некоторого множества данных. Свойства структур данных могут изменяться в процессе их использования. Как показывает практика, физические структуры, являющиеся оптимальными в определенных условиях, являются непригодными в других условиях.

В силу недостаточности знаний и изменчивости среды, в которой будет использоваться структура данных, попытки разработать структуру, эффективную во всех необходимых ситуациях, не представляются возможными.

Существующие реализации структур данных – это пассивные компоненты [1], использующие универсальные алгоритмы обработки данных. Такие компоненты достаточно слабо или вообще не учитывают особенности хранимых данных, их количество и интенсивность выполняемых операций над конкретными элементами данных.

Для повышения эффективности обработки данных целесообразно использовать знания о различных способах реализации одних и тех же логических структур данных разными физическими структурами. Каждая структура имеет свои особенности выполнения операций обработки данных. Для каждой из них можно указать условия, при которых эти операции будут выполняться наиболее эффективно (определенные размер и количество элементов данных, интенсивность их поиска и модификации и др.). Так, неупорядоченный список является эффективным при небольшом объеме данных и низкой интенсивности их использования, а иерархические структуры эффективно используются при достаточно больших объемах данных и достаточно высокой интенсивности их использования. Подобные условия можно назвать **областью эффективного использования** структуры данных [3].

При изменении условий обработки целесообразно использовать именно ту физическую структуру данных, которая максимально полно удовлетворяла бы текущим требованиям. Сложность операций обработки новой структуры должна быть минимальной в этом контексте, а сложность перехода от предшествующей структуры к новой структуре должна быть относительно невысокой. Это позволит эффективно обрабатывать данные и осуществлять преобразование структур данных в реальном времени, т.е. в темпе протекания процесса обработки данных – в течение относительно небольших перерывов между выполнением операций над данными. Такой подход к организации и обработке данных позволит создавать более сложные информационные системы, которые будут обладать механизмом самоорганизации на самом низком уровне – на уровне организации данных. Эти информационные системы следует отнести к **самоорганизующимся информационным системам** (СИС) [4], в которых логическая организация данных осуществляется в рамках эволюционной модели данных [2], а физическая организация данных конструктивно реализуется системой взаимодействующих автономных компонентов.

Автономный компонент организации данных (АКОД) – это программно-информационный модуль, способный самостоятельно функционировать в среде СИС.

Целью АКОД является как можно более длительное существование. Средством существования АКОД является его **способность выполнять** некоторую **работу**, т.е. эффективно хранить и обрабатывать определенную информацию. Для этого АКОД должен обладать определенной структурой и необходимыми ресурсами.

Построение модели АКОД будем осуществлять на основе системного подхода, принятого в [2]. При этом АКОД должен обеспечивать реализацию R-систем 1, 2 и 3-го рода.

В архитектуре АКОД целесообразно выделить следующие подсистемы: информационную, функциональную, оптимизации и адаптации и жизнеобеспечения.

Информационная и функциональная подсистемы реализуют структуры данных и методы их обработки. При этом должны быть реализованы четыре основные операции обработки данных: поиск требуемого элемента, добавление нового элемента, изменение и удаление существующего элемента, а также различные специализированные операции, допустимые для конкретных данных.

Подсистема оптимизации и адаптации осуществляет постоянный процесс оптимизации внутренней организации системы (механизм **внутренней эволюции системы**), который стремится минимизировать систему как по функциям, так и по структуре. Оптимизация АКОД направлена на уменьшение физического объема данных и сложности операций их обработки.

Механизмы оптимизации предполагают организацию данных в зависимости от их важности (по интенсивности использования и необходимости для формирования ответов). Например, можно выделить:

- наиболее важные данные (часто изменяемые) – для наиболее эффективного использования;
- средней важности (условно постоянные данные) – для достаточно эффективного использования;
- невысокой важности (постоянные данные, к которым применяется только операция поиска) – для наиболее эффективного хранения;
- незначительной важности (архивные данные, которые очень редко участвуют в обработке данных) – для эффективного хранения или полного удаления из системы.

Каждое такое подмножество элементов данных реализуется собственной физической структурой данных. Так, для постоянных и архивных данных целесообразно применять различные методы сжатия, а для наиболее часто используемых – методы кеширования.

Среди всех операций обработки данных может существовать некоторая операция, которая применяется с высокой интенсивностью к подавляющему большинству элементов структуры. В этом случае физическая структура данных должна быть ориентирована на эффективное выполнение этой операции.

Если к элементам данных операции обработки данных применяются примерно с одинаковой вероятностью, то в этом случае целесообразно сформировать иерархическую структуру – дерево некоторого вида. Использование деревьев обусловлено предположением равновероятного доступа ко всем элементам данных и незначительным ростом высоты дерева при увеличении количества данных. Как известно, в деревьях время доступа к данным и сложность их обработки имеют логарифмическую сложность, что дает им преимущество по сравнению с линейными структурами данных.

В процессе существования АКОД на него оказывают воздействие другие компоненты СИС, которые часто носят случайный характер. Случайность

внешних воздействий существенно снижает эффективность и надежность АКОД, противодействует процессу оптимизации. Случайные воздействия могут не позволить АКОД своевременно выполнить требуемую обработку данных, а высокая интенсивность таких воздействий может привести даже к разрушению АКОД.

Для повышения эффективности функционирования АКОД как открытой системе, необходим процесс адаптации (приспособления) к изменениям внешней среды (механизм **внешней эволюции системы**). Таким образом, АКОД должен проявлять свойства адаптивной системы, уметь идентифицировать и позиционировать себя во внешней среде. Для достижения собственных целей АКОД должен проявлять внутреннюю активность.

Для существования АКОД необходимо соблюдение **принципа адекватности** (соответствия, согласованности) системы и среды на уровне сложности и организации, которое носит не жесткий характер, а лишь определяет соответствие по параметрам сложности (количеству состояний) и количественной мере организации – организованности.

Поскольку СИС является средой переменной сложности и организованности, то при ее изменении АКОД должен стремиться достичь нового уровня адекватности по сложности и организованности со средой при минимальных затратах ресурсов (времени, памяти). При этом наиболее высокая степень адекватности достигается в том случае, когда одному состоянию (воздействию, стимулу) СИС соответствует несколько состояний (реакций) АКОД.

К механизмам адаптации необходимо отнести следующие методы:

- получение и накопление информации из внешней среды (о внешних объектах, о своем поведении и взаимодействии с внешними объектами);
- анализ накопленной информации;
- прогнозирование (интерполяция собственного поведения и поведения внешних объектов) и оптимизация поведения во внешней среде;
- разработка требований к реорганизации АКОД.

Таким образом, процесс адаптации должен регулярно повторяться во времени, создавая цикл обратной связи между внешней средой и собственным поведением.

Подсистема жизнеобеспечения реализует механизмы, защищающие систему от разрушения. Например, АКОД должен иметь методы валидации (проверки) непротиворечивости и целостности данных. Если операции обработки данных могут привести к нарушению структуры данных, то они либо запрещаются, либо вызываются методы, восстанавливающие структурные отношения.

Следует отметить, что в результате взаимодействия с внешней средой могут нарушаться структурные отношения и законы композиции информационной подсистемы АКОД. АКОД должен накапливать такие отклонения и использовать их в процессе оптимизации и реструктуризации. Способность обрабатывать исключения приводит к повышению устойчивости и надежности системы, снижает степень негативного воздействия внешней среды на АКОД.

Высокая сложность АКОД может приводить к частичной утрате способности системы к восстановлению после сбоев и потере данных. Поэтому крайне важно, чтобы существовала процедура восстановления данных, без которых не-

возможно дальнейшее функционирование АКОД. Для обеспечения такой возможности АКОД может периодически выполнять копирование этих данных и регистрировать все проводимые модификации (вести журнал изменений). Процедуры рестарта и восстановления после сбоев могут предъявлять специальные требования к физическому размещению данных, вызывая, например, необходимость дублирования части данных.

В качестве возможной реализации АКОД можно использоваться привычная структура классов, расширенная дополнительными категориями. В описании класса объектно-ориентированного программирования используются следующие понятия:

- данные (свойства) – описание характеристик объекта;
- методы обработки данных – описание действий с объектом;
- правила наследования методов и свойств объекта.

Это соответствует только информационной и функциональной подсистемам АКОД. Поэтому необходимо дополнить объект новыми структурными элементами: добавить методы оптимизации и адаптации, а также механизмы жизнеобеспечения. Объект должен самостоятельно осуществлять контроль своего состояния и проверку адекватности.

Активность нового объекта будет определяться наличием собственного, постоянно исполняемого процесса функционирования и способностью АКОД полностью управлять внутренней организацией и осуществлять определенные взаимодействия с внешней средой по собственной инициативе.

Библиографический список

1. Дрождин, В.В. Синтаксическая самоорганизация в доменно-ориентированных базах данных // Проблемы информатики в образовании, управлении, экономике и технике: сб. стат. III Всерос. науч.-техн. конф. – Пенза, 2003. – С. 26 – 28..

2. Дрождин, В.В. Системный подход к построению модели эволюционных баз данных. – Программные продукты и системы. – 2007. – № 3. – С. 52 – 55.

3. Дрождин В.В. Построение адаптивных самоорганизующихся простых доменов // Проблемы информатики в образовании, управлении, экономике и технике: сб. матер. Всерос. науч.-техн. конф. – Пенза, 2001. – С. 29 – 31.

4. Дрождин, В.В. Построение системы моделирования на основе самоорганизующейся информационной среды // Модели и алгоритмы для имитации физико-химических процессов: материалы Междунар. науч.-техн. конф. – Таганрог: Изд-во НП «ЦРЛ», 2008. – С. 251 – 255.