

[3]. ТП экспонируется построчно. Длина строки определяется лингвистическими переменными (ЛгП) “многословность” и “многозначность”, которые определяются числом слов и знаков в строке. Время экспозиции строки рассчитывается по набору *решающих правил* (РП), определенных для указанной пары ЛгП. На расчетное время накладывается медленное постоянное сокращение и небольшие циклически изменяющиеся (положительные и отрицательные) приращения времени экспозиции, аналогично описанному в [4]. Указанные процессы определяются собственными (вложенными) контурами нечеткого управления, построенными на своих ЛгП и наборах РП. На совокупности трех контуров строится система с обратной связью. ЧО при некомфортной скорости подачи строк воздействует на ТС (через клавиатуру компьютера), чем сокращает темпы нарастания подачи ТП.

При надлежащем подборе параметров трех указанных нечетких процессов, рассмотренная система обеспечивает компромиссное решение для звена (ТС«ЧО»). За счет обратной связи ТС поддерживает комфортную для ЧО скорость подачи ТП; в то же время ЧО постоянно испытывает со стороны ТС позитивное тренирующее воздействие, благодаря которому наращивает темп чтения.

Выводы

Звено взаимодействия “техническая система – человек оператор” представляет интерес, как среда моделирования взаимодействия интеллектуальных технических

систем. Человек - оператор является интеллектуальным элементом априорно; параметры технической системы могут варьироваться в процессе эксперимента; наблюдением и сопоставлением могут быть получены данные об эффективности информационного взаимодействия. В качестве прикладной области рассмотрена задача из области автоматизированного компьютерного обучения: тренинг оператора по скорочтению. Привлекательность данной задачи состоит в гарантированной положительной мотивации оператора, что важно для однородности проведения экспериментов.

Литература

1. Л.А. Заде. Понятие лингвистической переменной и его применение к принятию приближенных решений. М.: Мир, 1976.
2. Д.А. Поспелов. Нечеткие множества в моделях управления и искусственного интеллекта. – М.: Наука, 1986.
3. О.Ф. Михаль. Принципы организации системы дистанционного обучения скорочтению на локально-параллельных нечетких алгоритмах. // Образование и виртуальность. - 2002. Харьков - Ялта: УАДО, 2002, с. 112-116.
4. А.Г. Зырянов, С.С. Котегов и др. Скорочтение и ЭВА. // Микропроцессорные средства и системы.- 1990.- №16.- с. 90 - 92.

УДК 519.687

ТЕХНОЛОГИЯ ДЕДУКТИВНЫХ БАЗ ДАННЫХ В СИСТЕМАХ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ

Н.Н. Буслик

Доктор технических наук, профессор кафедры искусственного интеллекта, Харьковский национальный университет радиоэлектроники. пр.Ленина, 14, г.Харьков, Украина.

E-mail: buslik@datalab.kharkov.ua

Д.А. Летучий

Магистр-системный аналитик, младший научный сотрудник кафедры искусственного интеллекта, Харьковский национальный университет радиоэлектроники. пр.Ленина, 14, г.Харьков, Украина.

E-mail: letuchy@datalab.kharkov.ua

Проанализирована возможность применения технологии дедуктивных баз данных для реализации задач принятия решений. Рассмотрены характерные прикладные задачи, в которых использование традиционных методов теории принятия решений затруднено. Описаны компоненты дедуктивной базы данных. Предложены принципы создания систем управления дедуктивными базами данных.

Анализ проблемы

Многие современные информационные системы, обслуживающие бизнес-процессы и процессы управления организационно-техническими комплексами, характеризуются как системы поддержки принятия решений (СППР). Основными функциями такой системы являются изучение ситуаций, возникающих в предметной об-

ласти, выработка и оценка вариантов поведения на основе статистического и интеллектуального анализа данных, отражающих результаты управляемой деятельности на протяжении длительных периодов времени. Важное место в таких системах отводится применению передовых технологий баз данных и, прежде всего, - технологии интерактивной аналитической обработки данных (OLAP) [1]. Наряду с традиционными требованиями, предъявляемыми к СУБД, такими, как поддержка архитектуры

«клиент/сервер», распределенная обработка транзакций, индустриализация процессов сбора, накопления и первичной обработки данных, в СППР выдвигаются повышенные требования к анализу полноты и достоверности данных, возможностью составления и обработки нерегламентированных запросов поискового и аналитического характера. Другими словами, СУБД должна обеспечивать возможность реализации задачи углубленного («интеллектуального») анализа данных и принятия решений индустриальными методами, сравнимыми по своей технологичности с методами традиционной обработки данных.

Следует констатировать, что сегодняшний уровень развития технологий БД не в полной мере обеспечивает выполнение перечисленных требований. Так, наиболее распространенные в настоящее время СУБД, основанные на реляционном подходе (РСУБД), эффективно решают задачи ведения БД, статистического анализа данных, поиска данных по простым условиям, формирования отчетов. Подчеркнем, что для описания этих задач в большинстве случаев достаточным оказывается применение декларативных средств языка баз данных SQL-2, что, собственно, и обеспечивает удобство работы пользователей с БД, относительно низкую трудоемкость прикладного программирования, оперативность и эффективность реализации задач.

Однако для более тонкого анализа предметной области, описания семантики взаимосвязанных данных и соответствующих задач их интеллектуальной обработки такие средства оказываются уже недостаточными. Дополнительные осложнения привносят условия неполноты и нечеткости исходной информации, в которых часто приходится решать задачи формирования и оценки альтернатив. Обычно в этих случаях прибегают к применению процедурных расширений SQL или включающих процедурных языков, что негативно сказывается как на затратах, необходимых для разработки и реализации задач, так и на оперативности решения задач. Более того, в связи с тем, что многие задачи, которые потенциально могли бы решаться в среде БД, слабо формализуемы, не имеют четкого алгоритма решения или требуют глубокой математической проработки, они или вообще не решаются, или решаются с применением специализированных пакетов или экспертных систем, не интегрированных в среду функционирующей БД.

Приведем несколько примеров подобных задач.

1. Коммутация участков сети энергоснабжения [2]. В случае аварийного либо планового отключения фидерных устройств необходимо так замкнуть секционные выключатели, чтобы все потребители энергосети были запитаны. Очевидно, что альтернативами в этой задаче выступают различные топологии замкнутой сети энергоснабжения. Выбор лучшей топологии может вестись упрощенно по одному критерию: количество замкнутых секционных выключателей (оно должно быть минимальным).

2. Оптимизация обслуживания сети точек радиологического мониторинга. Как правило, при проектировании сети закладываются избыточные точки контроля (например, для мониторинга подземных вод, – дополнительные скважины). Эксплуатация таких точек приводит, с одной стороны, к дублированию информации, а с другой, – к временным и финансовым потерям. В то же время, указанная избыточность не покрывает потребностей мониторинга в случае аварийных ситуаций. Таким образом, задача состоит в нахождении избыточного

множества точек X ; все точки вне X должны быть «законсервированы», чтобы в нештатной ситуации их можно было оперативно ввести в эксплуатацию.

3. Выбор оптимальной стратегии игры [3]. Ряд задач принятия решений можно определить в терминах теории игр. Так, например, процесс взаимодействия заказчика и исполнителя можно рассмотреть как поочередную последовательность шагов, в каждом из которых заказчик пытается снизить для себя стоимость создания изделия, а исполнитель, наоборот, пытается повысить стоимость изделия (подразумевается, что изделие изготавливается в несколько этапов; каждый из этапов может быть реализован согласно одной технологии из некоторого их множества). Выбор лучшей стратегии заключается в определении таких действий (шагов) первого игрока, которые при любых действиях второго игрока не приведут к ухудшению позиций первого. Формально это означает, что необходимо найти ядро графового представления задачи, включающее конечные вершины графа, т.е. различные варианты исполнения изделия.

Отметим две характерные особенности рассматриваемого класса задач:

1. Правило порождения альтернатив достаточно просто формулируется на естественном языке, однако выбор и применение конкретной математической модели для формализации этого правила затруднено.

2. Комбинаторная природа задач в общем случае не позволяет решать их непосредственно, не прибегая к оптимизационным процедурам сужения области перебора.

Очевидно, что технология БД, ориентированная на эффективное применение в СППР, должна, прежде всего, основываться на использовании языков обработки данных высокого уровня, выразительные возможности которых превышают возможности SQL-2. Ближайшей перспективой развития РСУБД в данном направлении является внедрение стандарта SQL-3 [1], главное отличие которого от своих предшественников состоит в использовании так называемых рекурсивных запросов. Рекурсия на уровне запросов – мощный инструмент программирования БД, открывающий новые горизонты в индустриализации обработки данных и подготовке управленческих решений. Однако, по нашему мнению, SQL-3 – лишь промежуточный и кратковременный этап в развитии технологий СППР, на смену которому должны прийти технологии дедуктивных баз данных (ДБД), основанные на языках пятого поколения (языках логического программирования). (Доказательством этому служит хотя бы тот факт, что разработка систем SQL-3 опирается уже не столько на реляционную модель данных, сколько на логико-предикативную модель Datalog – абстрактного языка логического программирования БД).

Преимуществами Datalog-систем (СУДБД) являются лаконичность языка, близость его конструкций к естественному языковому, высокая выразительная мощность, покрывающая возможности SQL-3 (в отношении описания задач анализа данных). Вместе с тем, создание и применение СУДБД сопряжено с необходимостью решения целого ряда проблем как теоретического, так и практического характера, которые мы проанализируем ниже.

Постановка задачи.

Мы рассматриваем СУДБД, прежде всего, как средство реализации задач в СППР. Формально, общая задача принятия решений может быть представлена короткем:

$$\langle T, P, C, I \rangle,$$

где T – цель принятия решения (выбор наилучшей альтернативы, упорядочивание множества альтернатив, выделение групп альтернатив);

P – правило порождения альтернатив;

C – решающее правило относительно T ;

$I = \{I_{\text{вх}}, I_{\text{вых}}, I_{\text{реш}}\}$ – предметная область задачи (исходные данные), множество порожденных альтернатив и решение задачи согласно T соответственно [4].

ОЗПР относится к слабоструктурированным задачам, а ее компоненты в общем случае слабоформализуемы. Данный факт объясняет практическую нецелесообразность использования традиционного модельного подхода для реализации ОЗПР. Более того, традиционно в теории принятия решений основное внимание уделяется правилу выбора; подразумевается, что множество альтернатив заранее задано, или может быть легко найдено [5]. Однако, как следует из вышеперечисленных примеров, на практике встречаются задачи, в которых правило порождения альтернатив представляет особую сложность, в то время как правило выбора тривиально.

Прежде чем приступить к описанию аппарата ДБД, обратимся к главному тезису теоретико-доказательного (декларативного) решения задач: решить задачу – значит суметь ее описать. Таким образом, декларативное решение задачи подразумевает наличие логического языка с развитыми выразительными средствами и метод вычисления, который на основе аксиоматического представления предметной области с помощью правил вывода и унификации производит поиск решения.

ДБД представляет собой триплет $\langle EDB, IDB, IC \rangle$,

где EDB – экстенциональная база данных,

IDB – интенциональная (вычисляемая) база данных,

IC – множество ограничений целостности.

Компоненты ДБД задаются на Пролог подобном языке *Datalog*. Вычисление запроса в ДБД означает выполнение программы ДБД, в ходе которого все выведенные факты, релевантные запросу (целевому предикату), образуют модель программы ДБД, или, иначе, определяют решение ЗПР. Процесс вычисления имеет итеративный характер и прекращается в том случае, если невозможно вывести ни одного нового факта. Отметим, что в ряде случаев программа ДБД может иметь более одной модели [6]. Мы не заостряем внимание на различных аспектах программирования ДБД, с ними можно детально познакомиться, например, в [7, 8, 9].

Как уже отмечалось, решение ЗПР может быть осуществлено на основе некоторой ДБД, а, следовательно, СППР может быть выполнена на основе СУДБД, либо включать СУДБД в качестве своей подсистемы. СУДБД S можно представить в виде триплета:

$$S(L) : \langle In_{\text{ЗПР}}(L), F(L), Out_{\{A_i\}}(L) \rangle,$$

где L – язык ДБД, выступающий в качестве параметра системы;

$In_{\text{ЗПР}}(L)$ – подсистема описания ЗПР на языке L ;

$F(L)$ – подсистема анализа и интерпретации логической программы ДБД;

$Out_{\{A_i\}}(L)$ – подсистема вывода результатов, т.е. множества моделей A_i .

Нашей целью является, обобщив собственный опыт и опыт многих других исследователей [6, 7], предложить ряд принципов, следование которым предопределяет успешное решение сложных проблем, возникающих в ходе разработок СУДБД и их внедрения в СППР.

СУДБД как технологическая оболочка СППР.

1. Связывание ДБД с РБД. Факты предикатов могут храниться в таблицах некоторой РБД, соответственно подсистема $In_{\text{ЗПР}}$ должна содержать средства, обеспечивающие функции доступа к этим таблицам. Эффективность вычислений в ДБД напрямую зависит от степени интеграции СУДБД в реляционную СУБД. В самом лучшем случае, СУДБД выступает в качестве подсистемы РСУБД. Однако ввиду большой практической сложности расширения РСУБД функциями организации дедуктивных баз данных, на практике применяют менее сильные типы связываний. Развиваемый нами подход состоит в трансляции логической программы ДБД в предложение *SQL*, что позволяет, с одной стороны, отойти от реализации низкоуровневых функций хранения информации в ДБД, а, с другой, – использовать средства РСУБД для эффективного выполнения логической программы, такие, например, как оптимизатор запросов.

2. Единство средств описания задачи и управления данными. Традиционно ДБД рассматривается как логическая программа, подлежащая выполнению, а не как объект управления, т.е. собственно база данных. Это приводит к возникновению проблемы ведения ДБД: изменение *EDB*, а также задание новой цели вычисления, заключается в переписывании соответствующего файла логической программы. Рассматриваемый принцип, во-первых, предусматривает вынесение целевых запросов за рамки ДБД, а во-вторых, создание командного языка, расширяющего язык ДБД конструктами создания, модификации и удаления как самой ДБД, так и ее компонентов (например, пользователей, параметров связи ДБД с РБД, и т.д.).

3. Разделение среды формирования и реализации запросов. Данный принцип фактически развивает предыдущий принцип дополнительным требованием клиент-серверной организации СУДБД.

4. Открытость языковых средств. Так как класс решаемых задач непосредственно зависит от выразительных возможностей логического языка, необходимо обеспечить преемственность СУДБД новым языковым конструктам. Обычно новые версии систем программирования, адекватные соответствующим версиям языка, строятся таким образом, чтобы обеспечить возможность интерпретации существующих приложений без их модификации (преемственность версий). В то же время, это не означает, что сама система программирования как программный продукт не требует существенных переделок. Учитывая высокую динамику в развитии языковых средств СУДБД и широкие возможности их совершенствования, мы выдвигаем более жесткие требования в отношении открытости проекта системы: расширение языка новыми конструктами не должно приводить к переработке подсистемы трансляции. Этого можно достичь за счет использования средств метатрансляции, способных автоматически осуществлять перенастройку ядра транслятора при внесении изменений в синтаксис языка [10].

5. Управление вычислениями. Учитывая экспоненциальную сложность поиска решений в пространстве состояний ЗПР (в нашем случае речь идет о пространстве состояний или экземпляров *IDB*), необходимо обеспечить минимальность вычислений в СУДБД. Это подразумевает исключение повторного получения фактов в последовательности шагов вывода (*принцип минимальности итеративных вычислений*), а также устранение из процесса

вычисления правил, которые не могут дать новых фактов. Указанный принцип является основой для разработки эффективных методов вычисления в СУДБД. Более того, без его применения невозможно решение целого класса задач, связанных с проблемами комбинаторного перечисления. Например, в ряде случаев необходимо учитывать вывод внешне одинаковых фактов, полученных на разных шагах вывода, т.е. имеющих различную внутреннюю структуру (если из вершины а в вершину b существуют несколько путей, стандартная процедура вывода определит только единственный путь). Таким образом, принцип минимальности итеративных вычислений позволяет работать с мультимножествами фактов.

6. Декларативность языковых расширений. Как отмечалось, проблема расширения языка новыми конструкциями не перестает быть актуальной для разработчиков СУДБД. Есть два способа расширения языка: декларативное и процедурное. Ко второму, например, можно отнести реализацию функции backtracking и предикатов явного изменения (удаления и добавления) фактов из базы знаний. Несмотря на относительную простоту реализации процедурных расширений в логическом языке, их внедрение, с одной стороны, может привести к неоднозначной интерпретации программ, а с другой стороны, противоречит идее декларативного решения задач. Рассматриваемый принцип предполагает использование декларативных расширений языка, либо таких процедурных расширений, которые имеют прозрачную естественно-языковую трактовку и однозначную интерпретацию.

Выводы.

С позиций системного подхода проанализирована проблема применения методов и технологий искусственного интеллекта в информационных системах типа СППР, сформулированы принципы организации взаимодействия традиционных СУБД с системами дедуктивного вывода. Изложенные принципы нашли отражение в разработанном авторами

действующем прототипе СУДБД «Datalog server», ориентированном на выполнение задач в реляционной среде. Опытно-промышленная эксплуатация сервера показала принципиальную возможность и практическую целесообразность его применения для решения как традиционных, так и слабоформализованных задач анализа данных в организационно-технических системах.

Литература

1. Гарсиа-Молина Г., Ульман Дж., Уидом Дж. Системы баз данных. Полный курс. : Пер. с англ. – М.: «Вильямс», 2003. – 1088 с.
2. Говоров Ф. П., Говоров В. Ф., Верещук Б. М. Оптимизация режимов систем электроснабжения городов // ЕЕJET, № 2, 2003. – С. 28-33.
3. Басакер Р., Саати Т. Конечные графы и сети. : Пер. с англ. – М.: Наука, 1973. – 368 с.
4. Анфилатов В. С. и др. Системный анализ в управлении / В.С. Анфилатов, А.А. Емельянов, А.А. Кукушкин; под ред. А.А. Емельянова. – М.: Финансы и статистика, 2003. – 368 с.
5. Ларичев О. И. Теория и методы принятия решений. – М.: Логос, 2002. – 392 с.
6. Фернандес Х., Минкер Дж. Теория и алгоритмы дизъюнктивных дедуктивных баз данных // Программирование, 1993, № 3. – С. 4-38.
7. Чери и др. Логическое программирование и базы данных. : Пер. с англ. – М.: Мир, 1992. – 352 с.
8. Nilsson U., Maluszynski J. Logic, Programming and Prolog. John Wiley & Sons Ltd., 1995. – 294 p.
9. Буслик Н.Н., Летучий Д.А. Отрицательные литералы и вычислительная парадигма Дейталога // Проблемы бионики. Харьков, 2001, № 55. – С. 3-8.
10. Буслик Н.Н., Летучий Д.А. Организация объектно-ориентированного предикативного синтаксического анализатора // Проблемы бионики. Харьков, 2002, № 56. – С. 21 – 25.

УДК 621.314

ГРАФІЧНИЙ РЕДАКТОР ЕЛЕКТРИЧНИХ СХЕМ ДЛЯ ОБ'ЄКТНО-ОРІЄНТОВАНОГО МОДЕЛЮВАННЯ ПЕРЕТВОРЮВАЧІВ ЕЛЕКТРИЧНОЇ ЕНЕРГІЇ

В.В. Коломієць

Аспірант кафедри Автоматики та радіоелектроніки Української інженерно-педагогічної академії, вул. Університетська 16, м. Харків, Україна, 61003.

Контактний телефон: +38(0572) 32-80-43.

E-mail: VladKolomiets@rambler.ru

В.Г. Ягуп

Доктор технічних наук. Професор кафедри Автоматики та радіоелектроніки Української інженерно-педагогічної академії, вул. Університетська 16, м. Харків, Україна, 61003.

Контактний телефон: +38(0572) 20-64-87.

Проектування та аналіз електромагнітних процесів у схемах електронних пристроїв за допомогою комп'ютера, як правило, супроводжується кропітким введенням

Розглянуто питання, пов'язані з розробкою графічного редактора обробки схем тиристорних перетворювачів. Розробка здійснюється для дослідження електромагнітних процесів в тиристорних перетворювачах за допомогою об'єктно-орієнтованої моделі тиристорного перетворювача, що заснована на методах діакоптики. Обґрунтовано застосування і представлена реалізація об'єктно-орієнтованої моделі графічного редактора тиристорного перетворювача. Працездатність редактора підтверджено прикладом.

Вступ