ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА И ПРИКЛАДНАЯ МАТЕМАТИКА

УДК 004.383.3

В.Н. Ручкин, В.А. Романчук

РАЗРАБОТКА ПРОГРАММНЫХ СРЕДСТВ АНАЛИЗА НЕЙРОПРОЦЕССОРНЫХ СИСТЕМ

Рассматриваются вопросы моделирования и анализа эффективности нейропроцессорных структур для заданного класса алгоритмов. Приведена адаптация общей методики анализа многопроцессорных устройств для нейропроцессоров семейства NM640x для заданного алгоритма обработки информации. Предложены структуры нейропроцессорных систем и получены оценки эффективности для каждой структуры. Описана структура программного комплекса в соответствии с методикой анализа. Описан программный комплекс "НейроКС" для моделирования и анализа систем обработки информации на базе нейропроцессоров семейства NM640x.

Ключевые слова: *нейропроцессор, анализ многопроцессорных систем, обработка информации.*

Введение. В настоящее время для процессоров наступил так называемый "технологический предел", означающий что они достигли максимального уровня повышения быстродействия. Все разработки в данное время направлены на повышение числа процессоров на кристалле. Одним из выходов из данной ситуации является новая элементная база, например использование нейрокомпьютеров. Также необходимо отметить, что в области нейрокомпьютеров в настоящее время ведутся разработки с использованием новых технологий, перспективными можно назвать технологии создания оптических нейрокомпьютеров, нейрокомпьютеров на пластине, молекулярных и нанонейрокомпьютеров [1].

Специфика нейропроцессорных устройств заключается в том, что это устройства, обладающие специальными возможностями (функции активации, взвешенного суммирования и др.), ориентированные на эмуляцию и работу с нейронными сетями. Но данные возможности также делают их особенно эффективными для специализированной обработки информации (обработки изображений, распознавания, криптографии). Кроме этого, в таких устройствах реализовано не обычное, а матричное умножение, реализована переменная разрядность операндов, реализован встроенный конвейер команд, что делает их высокопараллельными устройствами.

Но для дальнейшего развития в области нейропроцессорных технологий существует ряд проблем, основными из которых являются [2]:

1. Невысокая производительность нейропроцессорных устройств в связи с небольшой частотой нейрочипов (30-150 МГц).
2. Малое количество программного обеспечения для нейропроцессоров по сравнению с общераспространенными процессорами.
3. Секретность информационных материалов в данной области.
4. Слишком большая цена перехода от существующих процессоров к нейропроцессорам (изменение не только аппаратных, но и программных средств).

Одним из способов решения первой проблемы является организация многопроцессорных систем. В настоящее время в области нейропроцессорных технологий ведутся исследования в части многопроцессорности, уже разработаны модули, включающие несколько процессоров с различными связями [плата ВМ1, плата МЦ4.04, плата МЦ4.13 (мезонин), МЦ9.01, разработан-

ные в НТЦ ”Модуль”; вычислительные модули SMT302, SMT344, SMT313, SMT315, SMT316 на базе 1,2 и 4 модулей семейства TMS320C4x, разработанные фирмой Sundance].

Но имеются проблемы, мешающие созданию эффективных мультимикропроцессорных структур на базе нейропроцессоров.

1. Нейропроцессоры являются пока дорогим и штучным товаром, и не каждая организация может их приобрести в нужном количестве. Кроме того, необходимо отметить, что для реализации какой-либо задачи необходимы эксперименты с различным количеством процессорных модулей, что также может позволить себе лишь крупная организация.

2. Проектирование и анализ специализированных многопроцессорных систем на базе нейропроцессоров являются очень трудоемким и сложным процессом, так как, в отличие от обычных процессоров, для нейропроцессоров нет необходимой теории, методов и алгоритмов и программных средств моделирования и анализа.

Целью работы является исследование работы многопроцессорных систем на базе нейропроцессоров.

Постановка задачи. Была поставлена задача разработки методики, моделей и алгоритмов с целью анализа эффективности нейропроцессорных систем (НПС) для заданного класса алгоритмов и создания программного комплекса, имеющего функциональные возможности для обеспечения всего цикла анализа эффективности систем обработки информации. Входной информацией является определенный класс алгоритмов, выходной – оценки эффективности той или иной нейропроцессорной системы для заданного класса алгоритмов.

Теоретические исследования. За основу методики анализа была взята общая методика анализа многопроцессорных систем для заданного класса алгоритмов. В дальнейшем она была адаптирована для нейропроцессорных устройств и систем обработки информации. Поэтапная схема анализа и последующей оптимизации НПС приведена на рисунке 1.

Рассмотрим каждый этап данной схемы.

1-й этап.

Первым этапом является выбор элементной базы. Будем считать, что в результате исследования предметной области и представленного класса алгоритмов наиболее рациональным выбором является семейство процессоров NM640x.

2-й этап.

На втором этапе требуется алгоритму обработки информации поставить в соответствие некоторую программу, написанную на внутреннем языке выбранного процессора, т.е. процессора семейства NM640x. То есть определить некоторое отображение φ, представленное в виде:

$$\varphi : A^{(j)} \to PR^{(j)}, j = 1, N,$$

где $A^{(j)}$ - некоторый j-й алгоритм обработки информации;

$PR^{(j)}$ - некоторая j-я программа для однопроцессорного варианта.

Важными выходными характеристиками данного этапа являются:

- длина программы $/PR^{(j)}/$, определяемая как число команд, входящих в программу;
- число одиночных команд, т.е. команд за исключением макрокоманд $/PR_i^{(j)}/$ (под макрокомандой будем понимать команду процессора, представляющую отдельный блок, повторяемый более одного раза);
- частота повторения макрокоманд $X_m{}^{(j)}, m = 2, M$;
- время выполнения программы $T(j) = \sum_{m=1}^{M} X_m{}^{(j)} t_m + \sum_{i=1}^{I} t_i$, где t_m - время выполнения m-й макрокоманды, t_i - время выполнения i-й одиночной команды.

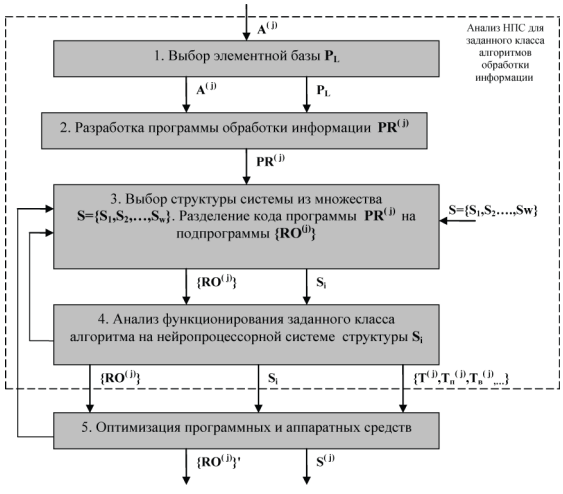


Рисунок 1 – Схема анализа и последующей оптимзации нейропроцессорных систем

3-й и 4-й этап

На третьем этапе необходимо рассмотреть множество возможных структур $S_w \in S$; $w = 1, W$, позволяющих некоторой программе поставить в соответствие множество подпрограмм. На четвертом этапе в результате анализа каждой структуры S_w определяются основные выходные характеристики, необходимые для

выбора наиболее рациональной структуры:

- время выполнения программы T;
- время выигрыша $T_в$ - это время, представляющее собой разницу времени обработки программы на нейропроцессорной системе относительно однопроцессорного варианта;
- время проигрыша системы T_n - это суммарное для всех процессорных модулей время, в течение которого i-й процессорный блок не был занят обработкой информации;
- время простоев в процессоре $T_n^{'}$ - суммарное время, в течение которого некоторые параллельные устройства процессора не были заняты обработкой информации.

Основными проблемами на данном этапе являются отсутствие единого стандарта нейропроцессорных архитектур и отсутствие критериев оценки эффективности рассматриваемых архитектур.

В связи с этим предлагается следующая методика:

1. Анализ программы $PR^{(j)}$.

Исследование программы $PR^{(j)}$ основано на понятии равенства подпрограмм $RO_j = RO_k$ (под которым понимаются равенство длин подпрограмм $/RO_j/ = /RO_k/$ и полное совпадение подпрограмм с точностью до микрокоманды $MK_l^{(j)} = MK_l^{(k)}, \forall l = 1,|RO_j|$). Такие подпрограммы можно объединить в отдельный класс, называемый классом эквивалентности, а затем можно определить вид параллельной системы (конвейерная, векторная, конвейерно-векторная, векторно-конвейерная) обработки исходя из числа найденных классов $L^{(j)}$ и порядка каждого класса $|a_l|$.

2. Выбор архитектуры в соответствии с числом классов эквивалентности $L^{(j)}$ и порядком класса $|a_l|$.

Рассмотрим следующие архитектуры и оценки эффективности:

а) если число классов эквивалентности равно L и порядок каждого класса равен единице $|a_l|$=1, тогда рациональным выбором будет конвейерная структура обработки информации.

- Время простоя в данном случае будет равно

$$T_n^{(j)} = L^{(j)} * TO^{(j)} - \sum_{l=1}^{L} TO_l^{(j)},$$

где $TO_i^{(j)}$ - время выполнения i-й подпрограммы.

- Время выигрыша:

$$T_в^{(j)} = \sum_{l=1}^{L} TO_l^{(j)} - \max_{l \in L} TO_l^{(j)};$$

б) если число классов эквивалентности L равно единице и порядок класса $|a_q|$равен q, то выбирается векторная структура.

- Время выполнения:

$$TO = \max_{l \in L} TO_l.$$

- Время проигрыша:

$$T_n^{(j)} = \sum_{i=1}^{q} (\max_{l \in L} TO_l^{(j)} - TO_i^{(j)}).$$

- Время выигрыша:

$$T_в^{(j)} = \sum_{l=1}^{L} TO_l^{(j)} - \max_{l \in L} TO_l^{(j)};$$

в) если число классов эквивалентности равно L и порядок каждого l-го класса равен $|a_l|$ и в случае, когда информация не требуется одновременно для всех представителей классов, тогда рационально использовать так называемую конвейерно-векторную структуру обработки информации.

- Время обработки:

$$T_R = \sum_{l=1}^{L} \sum_{r \in RO_l} t_i^{(l)}.$$

- Время проигрыша:

$$T_n = \sum_{l=1}^{L-1} (|a_l| - |a_{l+1}|) \sum_{r \in RO_l} t_i^{(l)}.$$

- Время выигрыша:

$$T_в = \sum_{l=1}^{L} |a_1| \sum_{r \in RO_l} t_i^{(l)} - \sum_{l=1}^{L-1} (|a_l| - |a_{l+1}|) \sum_{r \in RO_l} t_i^{(l)};$$

г) если информация требуется одновременно для всех представителей классов. Тогда, если подпрограммы внутри l-го класса обмениваются последовательно, назначим число процессорных модулей No для случая $|a_l| = q, \forall l = 1, L$, равное $No = L * q$, и для общего случая $No = L * \max_{l \in L} |a_l|$. В противном случае рационально использовать конвейерно-векторную структуру.

Получаемая структура обработки является векторно-конвейерной структурой обработки информации.

- Время выполнения программы:

$$T_R = \sum_{m=1}(\sum_{l=1}^{L}\sum_{r\in RO_l} t_i^{(l)} + \sum_{l=1}^{L}\sum_{r\in RO_l\setminus RO_1} t_i^{l}).$$

- Время проигрыша:

$$T_n = \sum_{l=1}^{L}|a_l| \sum_{r\in RO\setminus RO} t_i^{(l)} + \sum_{l=1}^{L}|a_{l+1}| \sum_{r\in RO_l\setminus RO_1} t_i^{(l)}.$$

Особенностью нейропроцессора является то, что он сам является векторным устройством, способным обрабатывать параллельно несколько потоков данных. Следовательно, он может быть рассмотрен с точки зрения системы с жесткой структурой и некоторыми особенностями.

В процессоре NM6403 существует два вида команд: скалярные и векторные.

Скалярную команду можно представить в виде совокупности двух операций: правой части и левой части (рисунок 2).

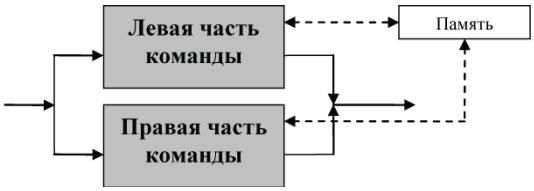


Рисунок 2 - Схема выполнения скалярной команды на процессоре NM6403

Следовательно, если рассматривать процессор, обрабатывающий скалярную команду, то можно сказать, что процессор представляет собой систему, параллельно обрабатывающую два потока данных.

Тогда:

время простоя будет равно

$$T_{kn}^{(j)} = 2 * \max_{l\in L} TO_{kl}^{(j)} - TO_{k1}^{(j)} - TO_{k2}^{(j)},$$

время выигрыша:

$$T_{kв}^{(j)} = TO_{k1}^{(j)} + TO_{k2}^{(j)} - \max_{l\in L} TO_{kl}^{(j)}.$$

Векторная команда также состоит из левой и правой части, но в правой части из-за переменной разрядности операндов возможно выполнение до 64 операций одновременно (рисунок 3).

Время простоя в этом случае равно:

$$T_{kn}^{(j)} = \sum_{i=1}^{L}(\max_{l\in L} TO_{kl}^{(j)} - TO_{ki}^{(j)}) + TO_{kn1}^{(j)} + TO_{kn2}^{(j)},$$

где $L \in \{2,...,64\}$.

Время выигрыша:

$$T_{kв}^{(j)} = \sum_{i=1}^{L} TO_{ki}^{(j)} - \max_{l\in L} TO_{kl}^{(j)} - TO_{kn1}^{(j)} - TO_{kn2}^{(j)},$$

где $L \in \{2,...,64\}$.

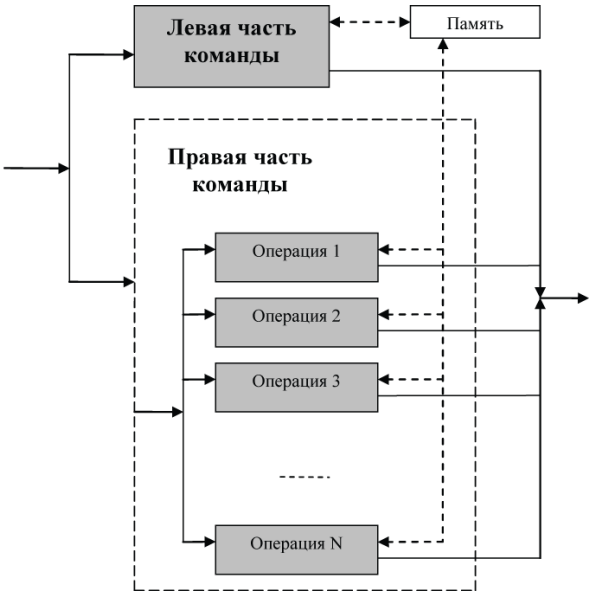


Рисунок 3 - Схема выполнения векторной команды на процессоре NM6403

5-й этап

Исходя из этого, на пятом этапе необходимо минимизировать время простоев и время проигрыша и, следовательно, повысить время выигрыша исходя из полученных на четвертом этапе оценок эффективности НПС для реализации данного алгоритма. Выполняется данный этап либо с помощью специальных программных продуктов, либо методами подбора наиболее рациональной конфигурации системы (для минимизации аппаратных средств), либо методами оптимизации программного кода (для минимизации программных средств), либо другими методами. На данном этапе также возможен возврат на третий этап по причине нерациональной структуры $S_i^{(j)}$ или неэффективного программного кода $PR^{(j)}$.

В соответствии с разработанной адаптированной методикой анализа НПС была разработана схема для анализа НПС на базе семейства нейропроцессоров NM640x для заданного класса алгоритмов.

Практические исследования. В соответствии с данной схемой была разработана структура программного комплекса.

После этого в соответствии со структурой был разработан программный комплекс. Интерфейс программного комплекса является многодокументным. Рассмотрим подробнее каждый модуль.

1. Анализатор программного кода процессора.

Для данного модуля была разработана модель процессора на основе структур, представленных на рисунках 2 и 3. Анализатор

программного кода разделен на анализатор программного кода скалярного процессора и анализатор программного кода векторного сопроцессора [4].

2. Текстовой редактор для языка нейроассемблера.

В данной части пользователь может ввести программу на языке нейроассемблера. Реализованы подсветка синтаксиса, специальные функции и отправка кода во внешние программы компиляции, отладки и др.

3. Текстовой редактор для языка C++.

В данной части пользователь может ввести программу на языке C++. Реализована отправка кода в компилятор Microsoft Visual C++.

4. Конструктор систем.

В данной части формируется система из процессоров (векторная, конвейерная, векторно-конвейерная и конвейерно-векторная). Реализовано аналитическое и графическое представление структуры.

5. Анализатор нейропроцессорной системы обработки информации.

В данном модуле реализованы моделирование и последующий анализ работы программы на одном или системе нейропроцессоров, для чего были разработаны модели в соответствии с конвейерной, векторной, конвейерно-векторной и векторно-конвейерной обработкой. Входными данными для процесса моделирования являются файлы кода на языках C++ и нейроассемблера и файл конструктора систем, разработанного в модуле ”Конструктор систем”. Выходными данными являются оценки эффективности выполнения заданного алгоритма обработки информации для системы заданной структуры. Эти оценки могут быть представлены в аналитическом и графическом виде.

В соответствии со схемой анализа эффективности НПС, приведенной на рисунке 1, была рассмотрена реализация алгоритма шифрования информации по методу ГОСТ 28147-89 с использованием разработанного программного комплекса.

Шаг 1**.** Элементной базой был выбран процессор NM6403.

Шаг 2**.** На основе алгоритма с помощью экземпляров подсистем ”Текстовой редактор”, ”Текстовой редактор для языка C++”, ”Анализатор программного кода” программного комплекса ”НейроКС” был разработан програмный код для однопроцессорного варианта системы.

Шаг 3**.** Данный код был протестирован и отлажен с помощью данного программного комплекса и исследования работы программного кода на эмуляторе процессора и реальном процессоре NM6403 (плата MC4.31).

Шаг 4**.** В соответствии с алгоритмом и кодом программы программу шифрования можно разбить на классы эквивалентности следующим образом: основной шаг криптопреобразования представляет собой самостоятельный класс эквивалентности с порядком $|a_1| = 32$. Но данный класс может быть также разбит на 5 классов эквивалентности. Тогда программа разбивается на 32*5=160 подпрограмм. Обмен данными между подпрограммами осуществляется последовательно, поэтому наиболее рациональной структурой является конвейерная. Следовательно, для реализации данной программы шифрования информации необходимо иметь 160 процессорных модулей (ПМ). На практике данная реализация является очень трудоемкой и дорогостоящей. Поэтому рассмотрим конвейерные структуры, включающие 1-8 процессорных модулей, для чего необходимо объединить некоторые классы эквивалентности в один и не рассматривать классы эквивалентности внутри основного шага криптопреобразования.

Шаг 5. В соответствии с выбранной структурой программный код был разделен на подпрограммы с помощью директив и экземпляра подсистемы ”Текстовой редактор”. Результатом данного шага является файл, разделенный директивами на подпрограммы, который может компилироваться как самостоятельная программа.

Шаг 6. С помощью функций отправки кода во внешние программы экзмепляра подсистемы ”Текстовой редактор” каждая подпрограмма была подготовлена для запуска на процессоре NM6403 (файлы являются абсолютными исполняемыми и имеют расширение .abs).

Шаг 7. Анализ выполнения данного программного кода на структуре, выбранной на шаге 4, производился с помощью экземпляра подсистемы ”Анализатор нейропроцессорной системы”.

Поэтапно были произведены моделирование и анализ следующих структур:

- конвейерная структура НПС для одного процессора (ПМ №1 выполняет все 32 цикла основного шага);
- конвейерная структура НПС для двух процессоров (каждый ПМ выполняет 16 циклов основного шага);
- конвейерная структура НПС для трех процессоров (ПМ №1 выполняет 10 циклов основного шага, ПМ №2 – 11 циклов, ПМ №3 – 11 циклов);
- конвейерная структура НПС для четырех

процессоров (каждый ПМ выполняет по 8 циклов основного шага);

- конвейерная структура НПС для пяти процессоров (ПМ №1 выполняет 6 циклов основного шага, ПМ №2 – 6 циклов, ПМ №3 – 6 циклов, ПМ №4 – 7 циклов, ПМ №5 – 7 циклов);
- конвейерная структура НПС для шести процессоров (ПМ №1 выполняет 5 циклов основного шага, ПМ №2 – 5 циклов, ПМ №3 – 5 циклов, ПМ №4 – 5 циклов, ПМ №5 – 6 циклов, ПМ №6 – 6 циклов);
- конвейерная структура НПС для семи процессоров (ПМ №1 выполняет 4 цикла основного шага, ПМ №2 – 4 цикла, ПМ №3 – 4 цикла, ПМ №4 – 5 циклов, ПМ №5 – 5 циклов, ПМ №6 – 5 циклов, ПМ №7 – 5 циклов);
- конвейерная структура НПС для восьми процессоров (каждый ПМ выполняет по 4 цикла основного шага).

На основе полученных результатов можно составить диаграммы изменения времени работы системы, коэффициента эффективности, времени выигрыша, времени проигрыша, зависимости коэффициента загруженности системы и времени проигрыша из-за внутреннего параллелизма процессора от числа процессоров.

Зависимость времени выигрыша от числа ПМ приведена в таблице.

Количество ПМ	Увеличение времени выигрыша
2	на 49.7 %
3	на 96.9 %
4	на 150.7 %
5	на 202.1 %
6	на 257.1 %
7	на 297.2 %
8	на 359 %

Время проигрыша менялось от 0 до 35000 тактов в связи с тем, что процессорные модули не обрабатывали информацию на протяжении первого цикла обработки, так как информация еще не была обработана предыдущим процессорным модулем.

Время простоев из-за внутреннего параллелизма процессора менялось от 300 до 440 тактов из-за необходимости добавления новых команд передачи данных, не всегда оптимизированных.

Шаг 8. Программная оптимизация производилась с использованием программного комплекса – рекомендаций в справочной системе в подсистеме ”Анализатор программного кода”. Аппаратная оптимизация – с помощью подсистемы ”Анализатор НПС”. В данной работе этот шаг не рассматривается.

Шаг 9. Запуск на процессоре в целью отладки и тестирования производился с использованием подсистемы ”Анализатор программного кода” и функций доступа к процессору и эмулятору.

Таким образом, полностью был рассмотрен процесс анализа многопроцессорной системы шифрования и дешифрования информации по методу ГОСТ 28147-89 с помощью программного комплекса ”НейроКС”. Кроме этого, программный комплекс также был использован при проектировании системы сжатия изображений фрактальным методом на базе нейропроцессора NM6403, что позволило выбрать наиболее рациональную структуру нейропроцессорной системы и оптимизировать программный код [3, 4].

Заключение. Таким образом, поставленные задачи в ходе исследования были выполнены и основными результатами стали:

- методика анализа средств обработки информации, адаптированная для нейропроцессоров семейства NM640x;
- для реализации данной методики произведен системный анализ, результатом которого стала классификация систем обработки информации;
- для получения оценок эффективности каждого процессорного модуля НПС нейропроцессор NM6403 был рассмотрен с точки зрения системы параллельной обработки данных векторной структуры;
- в соответствии с адаптированной методикой анализа определена структура и разработан программный комплекс;
- проведено экспериментальное исследование комплекса, для чего был рассмотрен процесс анализа нейропроцессорной системы шифрования информации по методу ГОСТ 28147-89. Показано, что в данном программном комплексе реализованы все возможности для обеспечения полного цикла анализа НПС и последующей обработки результатов: оптимизации, запуска и других этапов проектирования.

Библиографический список

1. *Галушкин А.И.*, Нейрокомпьютеры. Кн.3. – М: ИПРЖР, 2000. - 528 с.

2. *Галушкин А.И., Судариков В.А., Шабанов Е.В.* Нейроматематика: Методы решения задач на нейрокомпьютерах.- М: Препринт, 1990. - 440 с.

3. Нейрокомпьютеры в системах обработки изо-

бражений. Кн. 7: Коллективная монография / общ. ред. А. И. Галушкина. // Радиотехника, 2003.- 192 с.

4. *Ручкин В.Н., Романчук В.А., Колмыков М.В.* Возможности программного комплекса NM Model для разработки и отладки программ обработки изображений // Вестник РГРТУ. 2008. №2. Выпуск 24.- С.83-85

УДК 681.39

В.А. Антипов, Р.Е. Гузенко

ГРАФОВАЯ МОДЕЛЬ ДОМЕНА КОНТРОЛЯ И ДИАГНОСТИКИ ПРОЦЕССА ПРОИЗВОДСТВА РАДИОЭЛЕКТРОННОЙ АППАРАТУРЫ МЕДИЦИНСКОГО НАЗНАЧЕНИЯ

Рассматриваются вопросы постороения домена (DoG), который отображает семантику предметной области в терминах понятий и знаний.

Ключевые слова: *домен, графовая модель домена, проверка правильности графа.*

Введение. При семантическом моделировании домена контроля и диагностики были определены основные информационные объекты, которые соответствуют сообщениям Автоматизированного тестового и инспекционного оборудования о техническом состоянии контролируемых изделий РЭА медицинского назначения в процессе производства [1]. Теоретико-множественный подход [2] позволил формально отобразить семантику домена контроля и диагностики на структуру XML сообщений о состоянии диагностируемых изделий.

Цель работы: проиллюстрировать процесс определения графовой модели домена контроля и диагностики.

Графовая модель домена. Предложенную трёхуровневую метамодель графа типа [2], включающую формальный метод отображения домена предметной области в терминах понятий и знаний правил предметной области, будем называть графом домена (*DoG*).

Проиллюстрируем, как определяется *DoG*, как осуществляется его проверка правильности и как эти три уровня метамодели (*пример-уровень, тип-уровень, мета-тип-уровень*) связаны. Для этого введём простой пример домена контроля и диагностики процесса производства электронной аппаратуры медицинского назначения.

Будем моделировать контрольно-измерительные операции, которые идентифицируем как тип тестовой операции (ТипШагаТестирования), каждый шаг занимает определённое место в тестовой последовательности, идентифицированное номером (ШагТестирования). Для простоты будем моделировать единственную операцию контроля активного сопротивления с именем *R*, связанную с шагом номер "1".

Определим различные типы ограничений, выражающие знания предметной области.

Ограничения будут покрывать:

- ограничения связи начальных и конечных типов узлов;
- ограничения кардинальных чисел связей;
- ограничения промежуточных последовательностей связей в терминах промежуточных типов связей.

На рисунке 1 изображён пример модели.

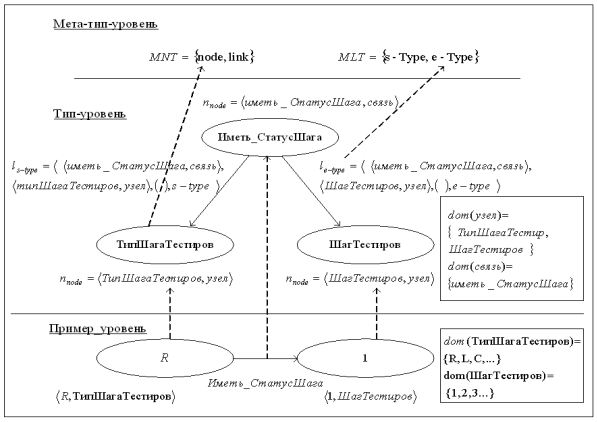


Рисунок 1

На *тип-уровне* домен смоделирован следующим образом.

- Понятия домена представлены как узлы *мета-типа* узла. В нашем случае это типШагаТестиров и ШагТестиров.
- Отношения между понятиями домена представлены как узлы *мета-типа* связи. В нашем случае – Иметь_СтатусШага.
- Структурные ограничения представлены как узлы и/или связи *тип-уровня*.

➢ Семантика ограничений определена посредством *мета-типа* связи ограничения. *Мета-тип s-type* ограничивает тип узла начала связи, *e-type* ограничивает тип узла конца связи. В нашем случае тип узла начала – Иметь_СтатусШага является типом контрольно-измерительной операции (ТипШагаТестиров), а тип узла конца – номер в тестовой последовательности (ШагТестиров).

Для введения ограничений на количество элементов и кардинальные числа (кардЧисла) будем предполагать, что число однотипных испытаний ограничено числом 5, но каждый шаг тестирования соответствует единственному контролируемому элементу. Это изображено на рисунке 2, причём изображена только уместная часть примера модели.

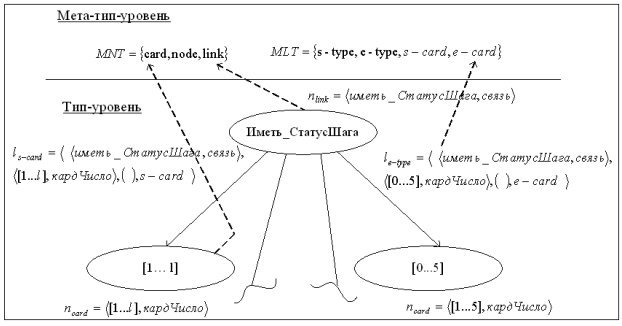


Рисунок 2

Диапазон возможных значений (домен) для ограничений кардинальных чисел определяется на основе *мета-типа*:

$$dom(card) \subseteq \underline{N} \times \underline{N};$$

$$dom(card) = \{(min,\ max\} | min,max \in \underline{N} \wedge \wedge min \le max\}.$$

Семантика ограничений определяется посредством *мета-типа* связи ограничения, который может быть *s-card* и *e-card*.

s-card ограничивает число примеров типов связей, которые соединяют один пример начального узла типа с примерами конечных узлов типов, которые лежат внутри диапазона, определяемого *min*, *max.*

e-card ограничивает число примеров связей типов, которые соединяют примеры начальных узлов типа с одним примером конечного узла типа, включённым в диапазон определения (*min*, *max*) кардинальных чисел.

Последовательность типов связей (узлов *мета-типа* связей) ограничивает последовательность промежуточных примеров связей косвенной связи.

Определим это как ограничение последовательности и введём тип мета-узла:

$$sequ \in MNT.$$

Расширим пример так, чтобы тип выполняемой тестовой операции принадлежал конкретному экземпляру тестируемого электронного узла (ЭУ). Следовательно, необходимо моделировать связи между типом тестовой операции и ЭУ. Дополнительно определим косвенную связь, отражающую то, что шаг тестирования принадлежит тестовой последовательности, а контролируемый элемент принадлежит конкретному экземпляру ЭУ (рисунок 3). Связь *ЭлементПринадлежит* изображена пунктирной линией. Определим домен возможных значений для ограничения последовательности (*sequ*). Для этого сначала определим две функции *s-node, e-node* на узлах gtn_{link} *тип-уровня*.

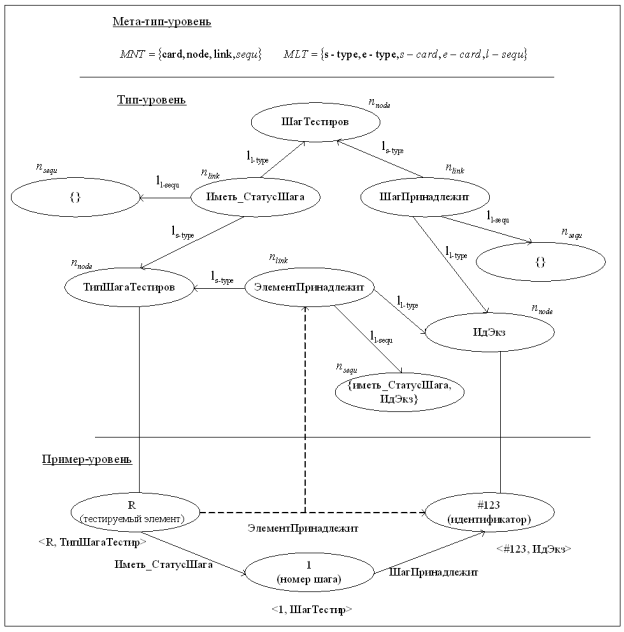


Рисунок 3

Набор GTN_{link} элементов gtn_{link} определяется как

$$GTN_{link} \subseteq GTN; \quad GTN_{link} = \{gtn \mid type: :(gtn) = link\}.$$

Узлы gtn_{link} представляют тип связи. Функция *s-node* отображает gtn_{link} на узлы *gtn*, которые определяют начальные тип-узлы связи, где

$$type(l_s\text{-}type) = s\text{-}type:$$

$$\forall\ gtn_{link}, gtn_s \in GTN\ \forall\ l_s\text{-}type \in GTL : [s\text{-}node(gtn_{link}) - gtn_s] \Rightarrow \exists\ l_{s\text{-}type} : s(l_s\text{-}type) = tn_{link} \wedge t(l_s\text{-}type) = gtn_s.$$

Таким же образом формально выражается функция *e-node*, которая отображает gtn_{link} на конечные узлы типов $gtn_{e.}$.

Область допустимых значений (домен) *dom*(*sequ*) определяется

$$dom(sequ) \subseteq \prod_{n \in \underline{N}} GTN_{link},$$

$$dom(sequ) =$$

$$= \{(gtn_{link1}, gtn_{link2}, \ldots, gtn_{link\,k}) | k \geq 1 \wedge \forall I, 1 \leq I \leq (k\text{-}1) : e\text{-}node(gtn_{linki}) = s\text{-}node(gtn_{linki+1})\}.$$

Пустая последовательность ограничений связи будет ребром графа.

Дополнительно определим субдомены, введя обозначение

$dom(sequ, gtn_{link})$ так, что

$dom(sequ, gtn_{link}) \subseteq dom(sequ)$;

$dom(sequ, gtn_{link}) = \{(gtn_{link1}, gtn_{link2}, \ldots, gtn_{link\,k}) | k \geq 1 : s\text{-}node(gtn_{link1}) = s\text{-}node(gtn_{link}) \wedge e\text{-}node(gtn_{link\,k}) = e\text{-}node(gtn_{link})\}$.

Субдомен является специфичным для определённого узла gtn_{link} и содержит только последовательности типов связей, которые начинаются в узле начала gtn_{link} и оканчиваются в узле окончания gtn_{link}.

Введённая композиция связей является мощным средством абстракции. Граф типов в первую очередь определяется посредством его узлов и прямых *тип-связей* tl^0, тогда как косвенные связи являются неявными для графа. Косвенные связи получаются прохождением вдоль соединяющих последовательностей связей, показывая неявную информацию, предоставляемую графом. Эта неявная информация делается явной посредством отображения таких последовательностей связей на явные связи определённого типа. Это, с одной стороны, предусматривает возможность извлечения соответствующей информации домена из сложного графа, а с другой - позволяет поднять сложный граф на более абстрактный уровень, сосредотачиваясь на понятиях домена и отношениях между ними.

Проверка правильности типового графа. Введём понятие правильности типового графа *TG*: *типовой граф считается правильным тогда и только тогда, когда он удовлетворяет структурным ограничениям, определённым его графом типа* (*GT*).

Формализация проверки является важной проблемой, так как она строит основу для разработки специфической алгебры, которая, в свою очередь, предусматривает доказательство разумности и законченности преобразований, определённых на типовом графе.

В разделе, посвящённом введению типов узлов и типов связей, было подчёркнуто преимущество и значение введённых иерархий типа. На графах иерархий типов определены множества, называемые *Субтипами* и *Супертипами.* Множества, содержащие субтипы и супертипы, дают элемент *gtn* (тип узла на *тип-уровне*). В зависимости от типа узлов (*gtn*), которые могут быть или узлами, или связями, определяемыми на *мета-тип-уровне*, мы рассматриваем иерархию типов узлов или иерархию типов связи.

Определим формально субтип $sub\text{-}type \in MLT$ для того, чтобы представить связь иерархий типа:

$$Sub\text{-}Types_{gtn} = \{gtn_i \mid type(gtn) = (node \vee \vee link) \wedge type(gtn_i) = type(gtn) \wedge \exists gtls : : [s(gtls) = gtn \wedge t(gtls) = gtn_i \wedge \forall \forall\, gtl \in gtls : type(gtl = sub\text{-}type]\},$$

где *gtls* является соединяющей последовательностью связей $gtl \in GTL$

$$Sub\text{-}Types^{+}_{gtn} = \{gtn\} \cup Sub\text{-}Types_{gtn};$$
$$Super\text{-}Types_{gtn} = \{gtn_i \mid gtn \in Sub\text{-}Types_{gtni}\};$$
$$Super\text{-}Types^{+}_{gtn} = \{gtn\} \cup Super\text{-}Types_{gtn}.$$

Ранее была определена функция на типах элементов *type*(.), определим дополнительную функцию на типах элементов *instance-of*. Функция *instance-of* отображает кортеж, состоящий из типа элемента и типа множества булевых значений {*true*, *false*} так, что *TN* – множество типов узлов, *GTN* – множество типов:

$instance\text{-}of : (TN \times GTN) \rightarrow \{true, false\}$,

запишем

$instance\text{-}of\ (tn, gtn) \in \{true, false\}$, $tn \in TN \wedge \wedge gtn \in GTN$,

отображение определяется

$$\forall\ tn \in TN\ \forall\ gtn \in GTN : [instance\text{-}of\ (tn, gtn) = = true \Rightarrow type\ (tn) \in Super\text{-}Types^{+}_{gtn}].$$

Для того чтобы формализовать проверку правильности, определим функцию *valid* на типовом графе. Функция *valid* отображает типовой граф на множество булевых значений {*true*, *false*}

$$valid: TG \rightarrow \{true, false\},$$
$$valid(TG) \in \{true, false\}.$$

В случае если $valid(TG) = true$, типовой граф является правильным, в противном случае он является неправильным.

Правильность типового графа определяется в терминах правильности узлов и связей графа:

$$TG = (TN, TL, dom(TG)):[\ valid(TG) = true \Rightarrow \forall\ tn \in TN : valid(tl) = true].$$

Типовой граф является правильным тогда и

только тогда, когда все типы узлов и все типы связей графа являются правильными.

Дополнительные функции были определены для выражения правильности единственного узла и единственной связи относительно введённых ограничений.

Для типа узла *tn* на его ассоциированном типе $t_{node} = type(tn)$ выбираем множество выходящих типов связей t_{link}, где

$$t_{node} \in GTN,\ GTN_{node,\ out} \subseteq GTN;$$

$$GTN_{node,\ out} = \{\ t_{link} \mid \text{s-}node(t_{link}) \in Super\text{-}Types^{+}(\ t_{node})\}.$$

Множество $GTN_{node,\ out}$ содержит типы связей с данным типом узла t_{node} или одним из его супертипов, определённых как начальный тип узла. Это рассматривает концепцию полиморфизма.

Рассмотрим множество входящих типов связей:

$$GTN_{node,\ int} \subseteq GTN;$$

$$GTN_{node,int} = \{\ t_{link} \mid e\text{-}node(t_{link}) \in Super\text{-}Types^{+}(t_{node})\}.$$

Построим подмножества из типов связей, содержащие входящие связи узлов типа *tn*, являющиеся частным случаем определённого типа связи:

$$TL_{tn,\ tlink,\ out} \subseteq TL;$$

$$TL_{tn,\ tlink,\ out} = \{tl \mid \text{s}(tl) = tn \wedge instance\text{-}of(tl, t_{link})\},$$

где $tlink \in GTNnode,\ out \wedge tnode = type(tn)$.

Определим подмножества, содержащие входные связи *tn*, являющиеся частным случаем определенного типа связи:

$$TL_{tn,\ tlink,in} \subseteq TL;$$

$$TL_{tn,\ tlink,int} = \{tl \mid s(tl) = tn \wedge instance\text{-}of(tl, t_{link})\},$$

где $t_{link} \in GTN_{node,int} \wedge t_{node} = type(tn)$.

Так как связь типа является частным случаем её типа связи, то она также является частным случаем супертипа этого типа связи, поэтому тип связи может содержаться более чем в одном множестве $TL_{tn,\ tlink}$, а именно, в любом множестве, определённом от имени супертипа t_{link}. Это обеспечивает уверенность в том, что тип связи (частный случай связи типа) является правильным, несмотря на ограничения, определённые посредством его связи типа, а также, несмотря на ограничения, определённые всеми супертипами его типа связи.

Определим функцию *valid-card*(*tn*), которая отображает тип узла в значениях *true* или *false*:

$$tn \in TN,\ t_{node} = type\ (tn),\ \forall\ t_{link\ 0} \in GTN_{tnode\ ,out},$$

$$\forall\ t_{link\ i} \in GTN_{tnode\ in}: [valid\text{-}card(tn) = true \Rightarrow$$

$$\Rightarrow |TL_{tn},\ t_{link\ 0,\ out}| \geq \min(s\text{-}sard(t_{link\ 0})) \wedge$$

$$\wedge |TL_{tn},\ t_{tlink\ 0,\ out}| \leq \max(s\text{-}sard(t_{link\ 0})) \wedge$$

$$\wedge |TL_{tn},\ t_{link\ i,int}| \geq \min(\text{e-card}(t_{link\ i})) \wedge$$

$$\wedge |TL_{in},\ t_{link\ i,\ \text{in}}| \leq \max(e\text{-}sard(t_{link\ i}))].$$

Тип узла является действительным тогда и только тогда, когда все ограничения кардинальности, определённые для выходящих и входящих связей, с учётом понятия полиморфизма, удовлетворены.

Теперь можно определить, как функция *valid*(*tn*) для элементов $tn \in TN$ и функция *valid*(*tl*) для элементов $tl \in TL$ отображает тип узла или тип связи на значения *true* или *fals* соответственно:

$$\forall\ tn \in TN: [valid(tn) = true \Rightarrow valid\text{-}card(tn)];$$

$$\forall\ tl \in TL: [valid(tl) = true \Rightarrow$$

$$\Rightarrow instance\text{-}of(s(tl), s\text{-}node(type(tl))) \wedge$$

$$\wedge instance\text{-}of(t(tl), e\text{-}node(type(tl)))].$$

Подобным же способом правильность графа типа может быть выражена относительно ограничений, определённых на *мета-тип-уровне*.

Определение DoG. В модели *DoG* семантика предметной области определена на *тип-уровне* посредством отображения понятий области и отношений между понятиями на типы узлов и типы связей. Знания предметной области представлены определёнными структурными ограничениями на *тип-уровне*. Дополнительно можно определить иерархии типов для того, чтобы описать модель домена на различных уровнях абстракции. Приведём пример определения конкретного *DoG* (рисунок 4).

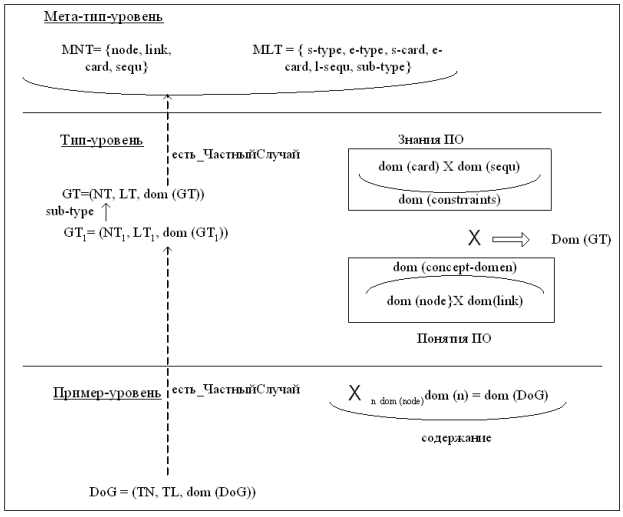


Рисунок 4

Определим домены для типов узлов и типов связей:

dom(*node*) = {ТипШагаТестирования, ШагТестирования, Экземпляр} – понятия предметной области;

dom(*link*) = {ИметьСтатусШага, ЭлементПринадлежит, ШагПринадлежит} – отношения между понятиями предметной области.

Чтобы определить структурные ограничения, определим ограничения для каждого типа связи.

Для каждого элемента множества определим сокращённое понятие:

$$Spec_{(type|gtl))} = \{\langle value(gtn_1), value(gtn_2) \mid \\ \mid gtn_1 = s(\text{gtl}) \wedge gtn_2 = t(gtl)\},$$

где $gtn_1, gtn_2 \in GTN \wedge gtl \in GTL \wedge type(gtl) \in MLT \wedge type(gtn_1) \in MNT, type(gtn_2) \in MNT$.

Приведём структурные ограничения:

$Spec_{(s\text{-}type)}$ = {<ИметьСтатусШага, ТипШагаИспытания>, <ШагТестирования>};

$Spec_{(e\text{-}type)}$ = {<ИметьСтатусШага, ШагаТестирования>, <ШагПринадлежит, Экземпляр>};

$Spec_{(s\text{-}card)}$ = {<ИметьСтатусШага, [1…1]>, <ШагПринадлежит, [1…n]>};

$Spec_{(e\text{-}tcard)}$ = {<ИметьСтатусШага, [1…5]>, <ШагПринадлежит, [1…n]>};

$Spec_{(l\text{-}sequ)}$ = {<ЭлементПринадлежит, {ИметьСтатусШага, ШагПринадлежит}>}.

Отображение семантики XML домена на DoG. Для того чтобы отобразить семантику XML домена на *DoG*, необходимо следовать формальному определению *DoG*. Во-первых, должны быть определены понятия XML домена и отношения между ними. XML определяет сообщения – структуры данных, которые будут составлены из элементов, атрибутов и секций данных, содержащих текст. Элемент может быть связан с другими элементами, которые находятся с ним в отношении предок/потомок и элемент может ссылаться на другие элементы посредством механизма идентификатор/ссылка (ID/IDREF). Дополнительно элемент может иметь некоторое текстовое содержание и атрибуты.

Основными понятиями XML домена являются: элемент, атрибут, текст.

Язык Связывания XML определяет простые и расширенные элементы связи, которые ссылаются на другие элементы, называемые источником и целью, и которые могут быть представленны XPath выражениями (XML языком пути) [3]. При этом язык Связывания XML используется для определения связи.

Заключение. Выбранный подход состоит в определении *DoG* как направленного графа. В этом случае становится возможным объединение конструкции XML Языка Связи в соответствующее определение типа. Понятия XML домена и соответствующих структурных ограничений, выраженных в форме графа типа *DoG*, делают возможным формальное определение произвольных XML структур и утверждение их правильности посредством предложенного теоретико-множественного подхода.

Библиографический список

1. *Антипов В.А., Соколов В.П.* Отображение семантики домена контроля и диагностики на структуру XML-сообщений // Биомедицинские технологии и радиоэлектроника. – 2007. № 7. – С. 59–68.

2. *Антипов В.А., Соколов В.П.* Трёхуровневая метамодель отображения семантики предметной области // Биомедицинские технологии и радиоэлектроника. – 2008, №7.

3. W3C XPath, *XML Path Language Recommendation,* http://www.w3c.org/TR/xpath, Nov. 1999.

УДК 681.3

А.В. Крошилин

ПРИМЕНЕНИЕ НЕЧЕТКОЙ КЛАСТЕРИЗАЦИИ ДЛЯ ЭФФЕКТИВНОГО МОНИТОРИНГА СТАТИСТИЧЕСКОЙ ИНФОРМАЦИИ В СИСТЕМАХ НЕОПРЕДЕЛЕННОСТИ

Предложен усовершенствованный алгоритм кластеризации на базе нечетких отношений равнозначности, порождаемых из свойств исследуемых данных в нечетких аналитических системах. Получена нечеткая оценочная функция качества кластеризации для построения нечеткой системы. Разработана методика нечеткой кластеризации с использованием

оценочной функции качества кластеризации, позволяющая полностью формализовать решение задачи кластеризации.

***Ключевые слова**: кластеризация, оценочная функция, аналитическая система.*

Введение. За последние годы в результате использования автоматизированных информационных систем, имеющих в своей структуре хранилище данных, во многих организациях скопились большие объемы данных, в которых заключено громадное количество дополнительной невыявленной и потенциально полезной информации [1]. В частности, в медицинских учреждениях работа с данными сводится к накоплению статистики и формированию отчетов по ней.

Эффективный мониторинг накопленной статистической информации позволяет, например, определить статистические показатели для выявления и оценки существующих и потенциальных угроз неблагоприятных эпидемиологических ситуаций и подготовить мотивационную базу для принятия управленческих решений, направленных на повышение эффективности мероприятий по устранению таких угроз. Эффективный мониторинг данных достигается путем применения методов интеллектуального анализа, особое место в которых занимают методы нечеткой кластеризации, при которой выбор наилучшего решения осуществляется по заданным критериям с использованием нечеткой функции принадлежности.

Цель работы - разработка нечеткой аналитической системы, осуществляющей сбор и эффективный мониторинг данных, путем применения методов интеллектуального анализа и метода нечеткой кластеризации.

Описание эффективного мониторинга данных для нечеткой аналитической системы. Интеллектуальный анализ данных – область знаний, относящаяся к обработке данных, изучающая поиск и описание скрытых, нетривиальных и практически полезных закономерностей в исследуемых данных. В основу современной технологии интеллектуального анализа данных положена концепция шаблонов (паттернов), отражающих фрагменты многоаспектных взаимоотношений в данных. Эти шаблоны представляют собой закономерности, свойственные подвыборкам данных, которые могут быть компактно выражены в понятной человеку форме. Поиск шаблонов производится методами, не ограниченными рамками априорных предположений о структуре выборки и виде распределений значений анализируемых показателей. При таком подходе решаются такие задачи как классификация, регрессии, поиска ассоциативных правил, а также задачи кластеризации [2].

В задачах *классификации и построения регрессионной модели* данных требуется на основании значений других переменных, характеризующих данный объект, определить значение зависимой переменной. Пусть дано конечное множество объектов (под объектами будем понимать исследуемые данные) предметной области (ПрО):

$$G = \{g_1, g_2, \ldots, g_i, \ldots, g_n\}. \quad (1)$$

Каждый из объектов ПрО характеризуется некоторым набором атрибутов:

$$g_i\,(x_{i1}, x_{i2}, \ldots, x_{ik}, \ldots, x_{im}, x_{im+1}). \quad (2)$$

Пусть значения атрибутов объекта ПрО g_i $(x_{i1}, x_{i2}, \ldots, x_{ik}, \ldots, x_{im})$ известны, а задача заключается в определении неизвестного атрибута x_{im+1}. При классификации множество значений x_{im+1} конечно, а при регрессии множество значений может иметь мощность континуума или быть конечным.

Решение задачи *поиска ассоциативных правил* заключается в выявлении закономерностей часто встречающихся наборов (групп) объектов ПрО в большом множестве таких наборов. Пусть каждому наблюдаемому событию соответствует некоторое подмножество U_h множества G, которое назовем событием:

$$G = \{\, U_h \,\}.$$

Рассмотрим множество P всех наблюдаемых событий. Пусть кардинальное число множества P равно m. Тогда

$$P_i = \{U_k: g_j \in U_k, j = 1 \ldots n, k = 1 \ldots m\} \subseteq P, \quad (3)$$

где P_i - множество событий, в которых наблюдался объект ПрО g_j,

$$P_E = \{U_k: P \subseteq U_k, k = 1 \ldots m\} \subseteq P \quad (4)$$

где P_E - множество событий, в которые входит набор объектов ПрО E.

Обозначим *Supp(E)* - поддержку набора E, которая определяется как отношение количества событий, при которых наблюдался набор объектов ПрО E, к общему количеству событий. Нахождение наборов E, поддержка которых превышает некоторое минимальное пороговое значение *Smin*, является задачей поиска

ассоциативных правил и записывается в виде:

$$M = \{E: Supp(E) > Smin\}. \quad (5)$$

Кластеризация имеет различные способы решения [2], но при любом подходе сложность заключается в отсутствии на момент начала анализа какой-либо дополнительной информации о данных, при этом возможное множество решений по кардинальному числу сопоставимо с входным множеством, что невозможно реализовать на практике. Методики выбора наилучших решений необходимы для качественного и быстрого решения задачи нечеткой кластеризации, при этом выбор наилучшего решения осуществляется формально, по заданным критериям и предварительной информации о кластерах не требуется.

Задача кластеризации заключается в разбиении конечного множества объектов ПрО $G = \{g_1, g_2, \ldots, g_i, \ldots, g_n\}$ на группы (кластеры) по некоторым атрибутам. Каждый из объектов ПрО g_i характеризуется m-компонентным атрибутным описанием $g_i(x_{i1}, x_{i2}, \ldots, x_{ik}, \ldots, x_{im})$, где $x_{ik} \in X_{ik}$, X_{ik} – допустимое множество значений атрибута. Необходимо построить множество кластеров K и отображение E: $G \rightarrow K$. Структура кластера:

$$k_h = \{g_j, g_p: g_j, g_p \in G, \mathrm{d}(g_j, g_p) < \psi\}, \quad (6)$$

где $k_h \in K$, k_h – кластер.

Таким образом, кластер состоит из объектов ПрО, находящихся в пространстве атрибутов X в пределах метрики d (оценочная функция), и ограничивается величиной ψ.

При осуществлении эффективного мониторинга накопленной статистической информации нашли свое применение методы нечеткой кластеризации, в которых элементы входного множества относят к тому или иному кластеру на основании значения нечеткой функции принадлежности, но на слабоструктурированных данных традиционные методы нечеткой кластеризации не дают адекватных решений. В эти методы закладывается ряд допущений: кластеры имеют особую внутреннюю точку – центр кластера и заданную форму; разбиение определяется с учетом взаимосвязей между данными и центрами кластеров. В общем случае кластеры могут быть произвольной формы и не иметь центров, поэтому был разработан метод кластеризации, свободный от указанных допущений и обеспечивающий разбиение только на базе отношений в имеющихся статистических данных. Таким образом, для качественного и быстрого решения задачи кластеризации необходимы методики выбора наилучших решений [3].

Методы решения задач интеллектуального анализа данных опираются на математический аппарат классической теории множеств, теории нечетких множеств, математической статистики, теории семантических сетей, а также аппарат универсальной алгебры и т.п. Алгоритмическое решение формализованной задачи связано с поиском экстремума оценочной функции.

Эффективный анализ данных в условиях неполноты, нечеткости исходной информации имеет нечеткий характер. Для их формализации в настоящее время успешно применяется аппарат теории нечетких множеств и нечеткой логики. Теория нечетких множеств имеет неоспоримое преимущество над вероятностными подходами, которое заключается в том, что экспертные системы, построенные на ее основе, обладают повышенной степенью обоснованности принимаемых решений. Это связано с тем, что в расчет попадают все возможные сценарии развития событий, что несвойственно вероятностным методам, рассчитанным на конечное (дискретное) множество сценариев. Нечеткие исходные данные в рассматриваемом случае формализуются в виде нечетких и лингвистических переменных, а нечеткость действий в процессе принятия решения – в виде нечетких алгоритмов [4]. Аналитические системы, способные формализовывать нечеткую информацию и обрабатывать ее в рамках нечетких алгоритмов, будем называть нечеткими аналитическими системами (НАС).

В разработанной интеллектуальной аналитической системе мониторинга пациентов на основе нечеткой кластеризации для медицинских учреждений «Диспансер» с помощью кластеризации решена задача первичного анализа информации, когда о внутренних зависимостях в данных ничего неизвестно. Благодаря кластерир кластеризации можно формулировать более детальные задачи о поиске зависимостей, влияющих на группировку данных в исходном множестве, и проводить эффективный мониторинг информации.

Существует большое число методов кластеризации [3], которые делятся на иерархические и неиерархические, среди которых наибольшую популярность получили методы разбиения (например: методы k-средних, Fuzzy C-Means и кластеризация по Гюстафсону-Кесселю и др.). В этих методах имеются недостатки: использование в решении понятия центра кластера (хотя он может отсутствовать); извлечение кластеров только формой, определенной алгоритмом (часть кластеров может быть пропущена); получение кластеров с учетом отношений между элементами данных и центрами кластеров. Преодолеть

эти недостатки возможно с помощью привлечения аппарата нечетких отношений, а связь атрибутов исследуемых данных рассматривать как нечеткие объектные связи.

Нормальная мера подобия по расстоянию $\mu_y(x)$ порождает нечеткие множества точек, близких к y, и имеет вид:

$$\mu_y(x) = 1 - \frac{d(y,x)}{\max\limits_{z \in X}(d(y,z))}, \quad (7)$$

где $x,y,z \in X$. При этом $\mu_y(x) = 0$, если образец данных максимально отличается от x, и $\mu_y(x) = 1$, если образцы данных абсолютно подобны с x для $x \in X$.

Определим *относительную меру подобия двух образцов данных* относительно третьего как $\tau_y(x,z)$ и запишем:

$$\tau_y(x,z) = 1 - |\mu_y(x) - \mu_y(z)|, \quad (8)$$

где $x,y,z \in X$, а μ_y – нормальная мера подобия.

В данном семействе отношений каждое отношение является нечетким отношением толерантности. Через $\tau_y(x,z)$ можно определить *меру подобия двух образцов данных на всем множестве* X как:

$$\tau(x,z) = \mathrm{T}(\tau y_1(x,z), \tau y_2(x,z) \ldots, \tau y_{|X|}(x,z)), \quad (9)$$

где T – t-норма, $\tau y_i(x,z)$ - относительная мера подобия, $y_i \in X$, $i = 1,\ldots,|X|$, $x,z \in X$. Таким образом, если два образца подобны относительно y_1 и y_2 и подобны относительно $y_{|X|}$, то два образца данных подобны относительно всего множества X. Полученное выражение, которое объективным образом показывает сходство между объектами из множества X, является *нечетким отношением толерантности.*

При вычислении транзитивного замыкания нечеткого отношения толерантности получается *нечеткое отношение равнозначности*. Для подтверждения этого доказан ряд утверждений и положений.

Утверждение 1. Задание уровня нечеткой равнозначности порождает разбиение множества X на группы равнозначных элементов таким образом, что каждый элемент X принадлежит точно одной группе равнозначности.

Утверждение 2. Транзитивное замыкание отношения нечеткой толерантности порождает отношение нечеткой равнозначности на множестве X.

Утверждение 3. Объединение отношений нечеткой толерантности также является отношением нечеткой толерантности.

Положение 1. Если определено отношение нечеткой толерантности R, то справедливо следующее утверждение: $R \subseteq R^2 \subseteq \ldots \subseteq R^n \subseteq \ldots$

Положение 2. Транзитивное замыкание $\hat{R}$, вычисляемое как наименьшая верхняя граница объединения отношений R^i, для отношения нечеткой толерантности R на множестве X равно отношению $R^{|X|}$.

С учетом всех рассмотренных аспектов был предложен алгоритм нечеткой кластеризации (рисунок 1), использующий нечеткое отношение равнозначности.

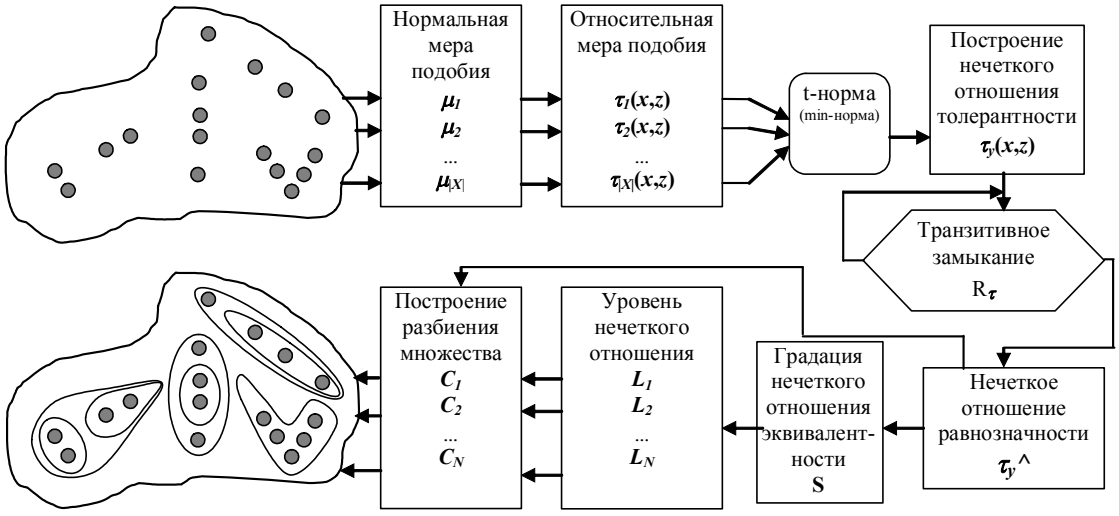


Рисунок 1 - Схема работы алгоритма

Алгоритм состоит из следующей последовательности действий.

1. Создать для каждого образца данных нормальную меру подобия по расстоянию $\mu_y(x)$.

2. Создать относительно каждого образца данных на основании нормальной меры подобия по расстоянию относительную меру подобия для двух образцов данных $\tau_y(x,z)$.

3. Создать меру подобия образцов данных на множестве X, при этом в качестве t-нормы используем min-норму:

$$\tau_y(x,z) = \min(\tau_{y1}(x,z),\ldots,\ \tau_y|X|(x,z)) \qquad (10)$$

- это нечеткое отношение толерантности на множестве X, $\forall x,y_i,z \in X$, i = 1,…,|X|.

4. Вычислить в цикле транзитивное замыкание отношения нечеткой толерантности:

$$R_\tau^i = R_\tau^{i-1} \circ R_\tau, \qquad (11)$$

где $R_\tau = \tau(x,z)$, $\forall x,z \in X$, $i = 2,\ldots,|X|$. В результате получим $R_\tau^{|X|}$, которое по утверждению 2 является отношением нечеткой равнозначности.

5. Создать для отношения нечеткой равнозначности градацию в виде упорядоченного по возрастанию множества различных элементов матрицы этого отношения.

Градация отношения нечеткой равнозначности порождает семейство отношений равнозначности в классическом смысле, каждое из которых разбивает исходное множество исследуемых данных на классы равнозначности. Чем больше уровень отношения, тем более детально разбиение множества *X*.

Предложенный алгоритм кластеризации на базе нечеткого отношения равнозначности позволяет эффективно выявлять в обрабатываемых данных кластеры; приведенная в работе нечеткая оценочная функция позволяет оценить качество проведенной кластеризации; описанная в работе методика нечеткой кластеризации и рекомендации по ее применению позволяют произвести эффективный мониторинг данных и сократить затрачиваемые на него ресурсы.

Построение нечетких аналитических систем. Практически решение можно свести к созданию комплексных нечетких аналитических систем различных типов и уровней сложности. В частности, разработана интеллектуальная аналитическая система мониторинга пациентов на основе нечеткой кластеризации для медицинских учреждений «Диспансер» ver. 4.0. Система позволяет определить статистические показатели для выявления и оценки существующих и потенциальных угроз неблагоприятных эпидемиологических ситуаций и подготовить мотивационную базу для принятия управленческих решений, направленных на повышение эффективности мероприятий по устранению таких угроз.

Аналитическая часть системы состоит из четырех основных блоков: блок начальной подготовки данных для анализа, блок настроек условий и способов анализа, блок нечеткой кластеризации, блок анализа результатов и вывода (рисунок 2).

Рисунок 2 - Укрупненная схема НАС

Блок начальной подготовки данных для анализа отвечает за подготовку множества данных для анализа и выделения атрибутов, по которым будет производиться анализ. Технологический этап перевода исследуемых данных в числовые и нормирования числовых данных в диапазоне [0,1] производится путем их взвешивания или упорядочивания. Взвешивание производится экспертом ПрО путем присваивания числовых значений категориальным атрибутам. При отсутствии эксперта можно произвести упорядочивание данных - каждому из значений категориального атрибута приписывается порядковый номер. Атрибут исключается из рассмотрения, если невозможно применить упорядочивание, а эксперт затрудняется с оценкой. Исследуемые числовые данные необходимо нормировать, чтобы каждый из атрибутов имел равный вес при сравнении. Также необходимо учитывать и вес атрибута относительно других атрибутов для правильного нормирования.

В *блоке настроек условий и способов анализа* задаются количество и состав кластеров, а также указываются наборы данных, которые исключаются из рассмотрения, - так называемые аномалии. Средствами анализа могут выступать различные программные системы и процедуры, как включенные в состав разработанной системы, так и другие среды, такие как Matlab. Ограничения могут накладываться на значение числа кластеров, получаемых в результате кластеризации. Выбор нечеткой оценочной

функции для оценки качества решения задачи может быть сделан на основании рекомендаций эксперта.

Блок нечеткой кластеризации - главный блок, который отвечает за выполнение алгоритма нечеткой кластеризации. В нем последовательно перебираются значения количества кластеров из заданного диапазона, осуществляется кластеризация и вычисляются значения выбранных атрибутов. Анализ завершается, когда достигнуты результаты, удовлетворяющие настройкам в предыдущем блоке, иначе производится новая итерация.

Блок анализа результатов и вывода отвечает за способ представления результатов. Результаты могут быть представлены в перечислении кластеров с указанием их элементов либо в виде матрицы принадлежности - таблицы, где строки соответствуют элементам, а столбцы - кластерам; в ячейках таблицы содержатся значения функций принадлежности.

На основе эффективного мониторинга данных и достижения целевых значений показателей выявляются причины отклонения показателей от плановых нормативов, связанные как с внешней средой, так и с внутренними проблемами организации.

Разработанная НАС, в состав которой входит предложенный алгоритм, позволяет получить дополнительные сведения для анализа информации по группам пациентов, диагнозам, заболеваемости, методам лечения и т.д. Это дает возможность осуществлять эффективную терапию, составлять отчеты, графики, диаграммы и документы на основе постоянно динамически изменяющейся информации.

Выводы. Достоинством нечеткомножественного подхода является его близость к естественному языку, что дает эксперту возможность формализовать свои нечеткие представления, трансформировав их в язык количественных оценок.

Для реализации эффективного мониторинга статистической информации с применением нечеткой кластеризации был выполнен ряд задач.

1. Предложен усовершенствованный алгоритм кластеризации на базе нечетких отношений равнозначности, порождаемых из свойств исследуемых данных и без использования дополнительных сведений о кластерах.

2. Получена нечеткая оценочная функция качества кластеризации для построения нечеткой системы.

3. Разработана методика нечеткой кластеризации с использованием оценочной функции качества кластеризации, позволяющая полностью формализовать решение задачи кластеризации, при этом оценивается качество каждого разбиения и выбирается наилучшее из них. Кроме того, данный подход позволяет производить эффективный мониторинг данных при значительном сокращении затрачиваемых ресурсов.

Таким образом, этот подход позволяет увеличить объем обрабатываемой статистической информации, что в значительной степени повышает эффективность мониторинга в медицинских учреждениях. Кроме того, система позволяет работать с несколькими моделями предметных областей, взаимосвязанными или не связанными между собой.

Библиографический список

1. *Киселёв М.В., Соломатин Е.А.* Средства добычи знаний в бизнесе и финансах // Открытые системы. №12. 2009. С.19-21.

2. *Дюк В.А.* Интеллектуальный анализ данных. - СПб.: Питер, 2008.

3. *Rhee F.C.*-H. Uncertain fuzzy clustering: insights and recommendations // IEEE Computational intelligence magazine. Vol. 2. № 1. 2007. P. 44-56.

4. *Крошилин А.В.* Разработка и анализ интеллектуальных поисковых программ в вычислительных сетях на основе универсальных алгебр: диссертация на соискание ученой степени кандидата технических наук. – Рязань: РГРТА, 2003. – 167 с.

УДК 621.396.969

О.Р. Никитин, А.В. Никитин, А.А. Шулятьев

МОДЕЛИРОВАНИЕ РАДИОТЕПЛОВОГО ЭЛЕКТРОМАГНИТНОГО ПОЛЯ В СЛОИСТЫХ СРЕДАХ

Разработаны компьютерные модели радиотеплового излучения некоторых видов слоистых сред. Проведена экспериментальная проверка

достоверности результатов компьютерного моделирования. Показано, что разработанные модели могут быть использованы для моделирования радиотеплового излучения природных сред.

Ключевые слова: *радиотепловое излучение, слоистые среды.*

Введение. Состояние окружающей среды является неотъемлемой характеристикой качества жизни и уровня благосостояния населения. Уязвимость экосистемы напрямую зависит от степени промышленного техногенеза территории, поэтому вопросы минимизации техногенного воздействия на окружающую среду имеют глобальную значимость.

При реализации программ мониторинга земной поверхности и разведки природных ресурсов наиболее часто используются аэрокосмические носители, оснащенные оптическими (видимого и инфракрасного диапазона длин волн), радиометрическими, магнитометрическими средствами.

Развитие средств обзора земной поверхности с помощью летательных аппаратов, в том числе беспилотных, выдвигает ряд задач в области радиофизического дистанционного зондирования (ДЗ), связанных с исследованием электрофизических, геометрических и статистических характеристик поверхности Земли. Это задачи повышения точности измерений и разрешающей способности, достоверности интерпретации полученных данных; проблемы выбора оптимальных условий проведения экспериментов при их планировании; проблемы, связанные с расширением функциональных возможностей средств зондирования. Решение этих задач является необходимым этапом в развитии стратегических направлений народно-хозяйственного комплекса России, включающим изучение природных ресурсов, мониторинг и экологическую защиту земных покровов и водной поверхности. Возможность решения указанных задач связана с тем обстоятельством, что искомые параметры и статистические характеристики поверхности отображаются в характеристиках рассеянных электромагнитных волн и собственного радиотеплового излучения земной поверхности.

В комплексной амплитуде отраженного сигнала при активном зондировании заключена информация о диэлектрической проницаемости поверхности, ее проводимости, рельефе, физико-химических свойствах. Сильная изменчивость радиотеплового излучения земной поверхности в зависимости от ее состояния является предпосылкой возможности дистанционной оценки некоторых параметров, характеризующих природные ресурсы (влажность, биомасса сельскохозяйственных культур, засоленность почв и др.), а малое по сравнению с водной поверхностью поглощение радиоволн почвой позволяет получать информацию не только о состоянии самой поверхности, но и о физических характеристиках протяженного поверхностного слоя [1, 2].

Для осуществления специальных радиофизических методов и аппаратуры дистанционного зондирования необходимо располагать сведениями о структуре и характеристиках электромагнитного поля (ЭМП) в природных средах и биологических объектах, что наиболее удобно выполнить с помощью средств моделирования. Электродинамические модели являются исходной априорной информацией, дающей аналитическое описание принимаемых полей или их статистических эквивалентов и определяющей связь регистрируемых амплитуд или мощностей сигналов с электрофизическими параметрами и статистическими характеристиками природных сред (поверхностей, атмосферы, слоистых покровов и др.).

Целью настоящей работы были создание компьютерных моделей некоторых природных сред, описанных эквивалентной слоистой структурой, и экспериментальная проверка их адекватности.

Теоретические сведения. Многие варианты подстилающей поверхности могут быть представлены слоисто-неоднородными средами с различными электродинамическими параметрами. Преимуществом подобных моделей являются отсутствие ограничений на число учитываемых слоёв и их электродинамические параметры, возможность введения локально распределённых неоднородностей в каждом слое.

В работах [3, 4] представлено математическое описание многочисленных вариантов радиотеплового излучения слоисто-неоднородных сред. Некоторые из них послужили основой для разработки пакета прикладных программ для компьютерного моделирования.

Простейшей из моделей слоисто-неоднородных сред является однослойная модель [5]. Данная модель является практически применимой при выполнении критерия Рэлея: $h < \frac{\lambda}{16\sin\theta}$, где h – высота неровностей на границе среды [6]. Коэффициенты отражения Френеля для случая излучения горизонтальной и

вертикальной поляризаций могут быть вычислены по формулам:

$$\dot{K}_{fГ} = \frac{\sqrt{\dot{\varepsilon}_1}\cos\theta_1 - \sqrt{\dot{\varepsilon}_2 - \dot{\varepsilon}_1\sin^2\theta_1}}{\sqrt{\dot{\varepsilon}_2}\cos\theta_1 + \sqrt{\dot{\varepsilon}_2 - \dot{\varepsilon}_1\sin^2\theta_1}}, \quad (1)$$

$$\dot{K}_{fВ} = \frac{\dot{\varepsilon}_2\cos\theta_1 - \sqrt{\dot{\varepsilon}_1}\sqrt{\dot{\varepsilon}_2 - \dot{\varepsilon}_1\sin^2\theta_1}}{\dot{\varepsilon}_2\cos\theta_1 + \sqrt{\dot{\varepsilon}_1}\sqrt{\dot{\varepsilon}_2 - \dot{\varepsilon}_1\sin^2\theta_1}}.$$

Для слоисто-неоднородных сред, состоящих из двух слоёв, коэффициенты Френеля при значениях магнитных проницаемостей $\mu_1 = \mu_2 = \mu_3 = 1$ и диэлектрической проницаемости верхней среды $\varepsilon_1 = 1$ могут быть найдены по формуле:

$$\dot{K}_{f2B,Г} = \frac{\dot{K}_{f21} + \dot{K}_{f32}e^{-2ik\Delta h\sqrt{\dot{\varepsilon}_2 - \sin^2\theta}}}{1 + \dot{K}_{f21}\dot{K}_{f32}e^{-2ik\Delta h\sqrt{\dot{\varepsilon}_2 - \sin^2\theta}}},$$

где $\dot{K}_{fMN}$ обозначает коэффициенты Френеля, рассчитанные по формулам (1) для однослойной модели для *M*-го и *N*-го слоёв в предположении, что наблюдение ведётся из *N*-го слоя, Δh – толщина верхнего слоя.

Для трёхслойных сред расчётные формулы при тех же допущениях (равенство магнитной проницаемости всех сред и диэлектрической проницаемости верхней среды единице) имеют вид:

$$\dot{K}_{f3B,Г} = \frac{\dot{K}_{f12} + \dot{W}\dot{S}^2}{1 + \dot{K}_{f12}\dot{S}^2},$$

где

$$\dot{W} = \frac{\dot{K}_{f23} + \dot{K}_{f34}\dot{F}^2}{1 + \dot{K}_{f23}\dot{K}_{f34}\dot{F}^3};$$

$$\dot{S} = \exp\left(-ik\Delta h_1\sqrt{\varepsilon_2 - \sin^2\theta}\right);$$

$$\dot{F} = \exp\left(-ik\Delta h_2\sqrt{\varepsilon_3 - \sin^2\theta}\right).$$

Во всех рассмотренных случаях радиояркостная температура излучения поверхности может быть вычислена по формуле [7]:

$$T_Я = \left(1 - \left|\dot{K}_f\right|^2\right)T_0,$$

где T_0 – температура всей слоистой среды (предполагается, что среда изотермическая).

В ходе работы была создана программа, в которой алгоритмически реализованы три упомянутые выше модели плоских поверхностей со ступенчатым законом изменения показателя преломления (однослойная, двухслойная и трёхслойная). В качестве среды программирования использовался математический пакет Matlab, хорошо подходящий для такого рода вычислений. Программа построена в соответствии с парадигмой структурно-ориентированного программирования, что позволило сократить время разработки и оставило большой запас гибкости для дальнейшего расширения программы, которое в будущем планируется.

Экспериментальные исследования. Для проверки достоверности расчётов была поставлена серия экспериментов. Схема экспериментальной установки показана на рисунке 1. Здесь А – антенное устройство, P_1, P_2 – радиометры, ПК – персональный компьютер. В качестве радиометров использовались специализированные радиометры, работающие в диапазоне 8 мм и 3 мм.

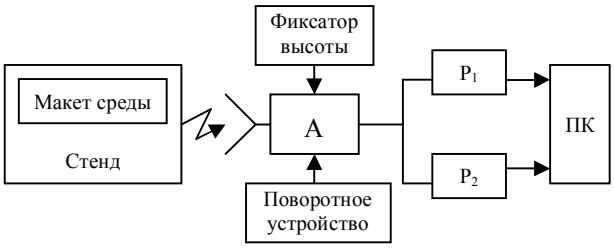


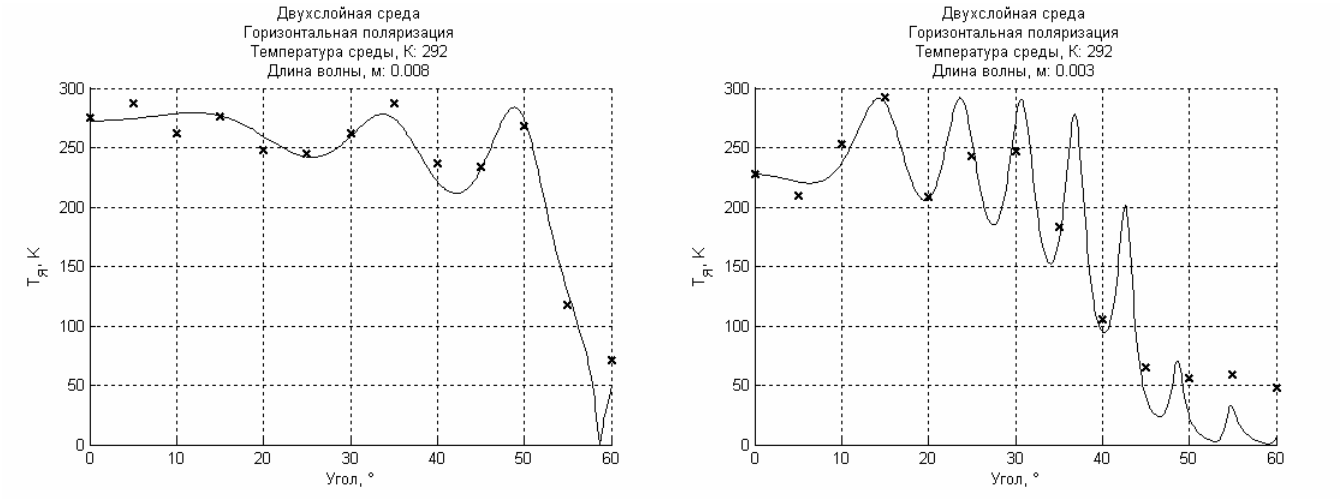
Рисунок 1 – Структурная схема экспериментальной установки

Сведения о параметрах сред во всех экспериментах представлены в таблице 1.

Таблица 1

Вид среды	Материал	Толщина, см	$\dot{\varepsilon}$	
			λ = 8 мм	λ = 3 мм
Однослойная	воздух	–	1	1
	песок	–	4 + 0,4*i*	2,5 + 0,6*i*
Двухслойная	воздух	–	1	1
	бетон	7	5,5 + 0,5*i*	5,5 + 0,36*i*
	песок	–	4 + 0,4*i*	2,5 + 0,6*i*
Трёхслойная	воздух	–	1	1
	асфальт	5	2,5 + 0,6*i*	2,25 + 0,18*i*
	бетон	7	5,5 + 0,5*i*	5,5 + 0,36*i*
	песок	–	4 + 0,4*i*	2,5 + 0,6*i*

На рисунке 2 представлены результаты компьютерного моделирования радиотеплового излучения двухслойной среды горизонтальной поляризации и соответствующие экспериментальные данные для двух длин волн (*а* – 8 мм, *б* – 3 мм).



а *б*

Рисунок 2 – Теоретические и экспериментальные кривые зависимости интенсивности радиотеплового излучения от угла обзора

В таблице 2 приведены значения среднеквадратического отклонения результатов компьютерного моделирования от экспериментально измеренных значений.

Таблица 2

Вид среды	Поляризация			
	горизонтальная		вертикальная	
	$\lambda = 8$ мм	$\lambda = 3$ мм	$\lambda = 8$ мм	$\lambda = 3$ мм
однослойная	2,1%	3,9%	2,5%	4,7%
двухслойная	5,2%	8,8%	6,1%	9,3%
трёхслойная	7,9%	10,3%	9,1%	10,7%

Заключение. На основании экспериментов сделаны следующие выводы. Во-первых, погрешность измерения имеет тенденцию возрастать с увеличением числа слоёв и уменьшением длины волны. Во-вторых, качественно наблюдаемое согласие результатов эксперимента с теорией позволяет сделать вывод о достоверности разработанных моделей. Подытоживая всё вышесказанное, следует заключить, что разработанные компьютерные модели могут быть использованы для моделирования радиотеплового излучения природных сред. Погрешность моделирования при этом не превышает 10 % для данного типа задачи.

Пользуясь данными моделями, можно составить банки данных параметров радиотеплового электромагнитного поля в природных средах с различными радиофизическими и геофизическими параметрами.

Библиографический список

1. *Андреев Г.А.* Тепловое излучение миллиметровых волн земными покровами // Зарубежная радиоэлектроника. 1982. № 12. С. 3-39.

2. *Шульгина Е.М.* Радиотепловое зондирование земных покровов // Зарубежная радиоэлектроника. 1993. № 4. С. 59-68.

3. *Волосюк В.К.* и др. Математические методы моделирования физических процессов в задачах дистанционного зондирования Земли / В.К. Волосюк, В.Ф. Кравченко // Успехи современной радиоэлектроники. 2000. №8. С. 3-75.

4. *Волосюк В.К.* и др. Математические методы моделирования физических процессов в задачах дистанционного зондирования Земли / В.К. Волосюк, В.Ф. Кравченко, В.И. Пономарёв // Успехи современной радиоэлектроники. 2000. №12. С. 3-74.

5. Под ред. Никитина О.Р. Решение экологических задач наземно-дистанционными радиофизическими методами: монография / О.Р. Никитин, В.М. Гаврилов. — Муром: Изд. Полиграфический центр МИ ВлГУ. — 2009. – 105 с.: ил.

6. *Голунов В.А.* Влияние атмосферы и шероховатости поверхности на радиометрические характеристики естественных покровов / В.А. Голунов, А.Ю. Зражевский, А.Г. Павельев // Радиотехника и электроника. 1988. № 12. С. 2544-2550.

7. *Гуляев Ю.В.* Использование миллиметровых волн для диагностики поверхности и атмосферы Земли и в фундаментальных исследованиях / Ю.В. Гуляев, В.Ф. Кравченко, И.А. Струков // Радиотехника. 1995. № 4-5. С. 83-87.

УДК 004.622

А.Н. Андреев

ПРОЕКТИРОВАНИЕ ОНТОЛОГИИ ВЕРХНЕГО УРОВНЯ ДЛЯ УНИФИЦИРОВАННОГО ОПИСАНИЯ OLAP КУБОВ

Рассматривается процесс проектирования онтологии верхнего уровня для унифицированного описания OLAP кубов, а также формализация понятий OLAP кубов на основе технологии Semantic Web.

Ключевые слова: *многомерная модель, ETL, OLAP куб, онтология, Semantic Web, OWL.*

Введение. Технологии Semantic Web являются мощной инфраструктурой для построения и распространения знаний в Web. Данная технология использует XML (eXtensible Markup Language – расширяемый язык разметки), RDF (Resource Description Framework – язык представления информации в Web), RDFS (RDF Schema – расширение RDF) и OWL (Web ontology language – язык описания онтологий) для описания web контента. Основой технологии Semantic Web для обработки формализованных знаний являются онтологии. Согласно Тому Груберу: «Онтология – это точная спецификация концептуализации» [1]. Развитие технологии Semantic Web, введение новых типов данных и новых подходов к их хранению и обработке требует и новых подходов к их бизнес-анализу.

В последние годы были предложены различные подходы для объединения технологий Semantic Web и технологий хранилищ данных с целью увеличения семантической поддержки в распределенных хранилищах. Данные подходы можно условно разделить на две группы:

- использование технологий Semantic Web в системах ETL;
- использование технологий Semantic Web в OLAP системах.

Использование технологий Semantic Web в системах ETL (Extract, Transform, Load – процесс в управлении хранилищами данных, включающий извлечение данных из внешних источников, их трансформацию и очистку, а также загрузку их в хранилище данных) предусматривает использование словарей с наименованием схем, аннотаций и описаний источников данных и хранилищ данных для выбора релевантной информации, генерирования онтологий приложения, трансформации атрибутов. Это позволяет полуавтоматически генерировать потоки данных ETL, сокращать усилия в разработке приложений хранилищ данных.

Технологии Semantic Web используются для процесса разработки OLAP приложений (OnLine Analytical Processing – оперативная аналитическая обработка), устранения конфликтов при построении кубов данных. Онтологический подход к проектированию процесса интеграции данных предполагает использование онтологий и при проектировании OLAP систем.

Описания понятий OLAP систем не достаточно формализованы. Некоторые авторы рассматривают понятия OLAP систем с точки зрения хранилищ данных, что не является корректным. Например, таблица фактов может иметь множество измерений времени. В этом случае достаточно только одного измерения времени на уровне хранилища данных для поддержки множества измерений времени на уровне куба. В связи с этим возникает задача унифицированного описания OLAP систем и концептов вне зависимости от предметной области.

Цель работы: унификация описаний OLAP кубов, которые не будут зависеть от конкретной предметной области, а также формализация понятий OLAP кубов на основе технологии Semantic Web.

Основные понятия систем класса OLAP. Традиционно OLAP приложения основываются на многомерном моделировании, которое позволяет интуитивно рассматривать данные согласно метафоре куба. Куб – визуальное представление таблиц хранилища данных. Ячейки куба представляют собой события, которые происходят в бизнес-среде. Каждое событие определяется набором мер, каждое ребро куба соответствует измерению [2].

Рассмотрим базовую (фактическую) таблицу *r*, на основе которой строится OLAP куб. Множество атрибутов *r* условно делится на 2 группы

1. Набор измерений, которые служат критериями для анализа и определяют многомерное пространство OLAP куба. За счет фиксации значений измерений получаются срезы (гиперплоскости) куба. Каждый срез представляет

собой запрос к данным, включающий агрегации.

2. Набор мер – функции, которые каждой точке пространства ставят в соответствие данные.

Многомерное пространство для r определяется следующим образом:

$$Space(r) = \left\{ \times_{A \in D} \left(Dim[A] \cup ALL \right) \cup (0,0,...,0) \right\},$$

где $\times$ – декартово произведение, $Dim[A]$ – проекция r на атрибут A, ALL – множество всех возможных значений в иерархии измерения, $(0,0,...,0)$ – нулевой кортеж. $\forall s \in \mathrm{Space(r)}$, s – многомерный кортеж [3,4].

Основные подходы к работе с онтологиями. Существуют несколько подходов к классификации онтологий. Одним из них является классификация по уровню общности [5]. Согласно данному подходу выделяют:

- онтологии верхнего уровня (top-level ontologies) – те, которые определяют только общие характеристики понятий и не связаны ни с какими конкретными областями;
- предметные онтологии (domain ontologies) – те, которые определяют специфичные для области характеристики понятий;
- прикладные онтологии (application ontologies) – те, которые содержат все необходимые знания для моделирования конкретной задачи.

Для унифицированного описания OLAP куба рассмотрим онтологию верхнего уровня, так как она не привязана ни к одной прикладной области.

Язык OWL. Web ontology language (OWL) или язык описания онтологий – это XML словарь, который используется для определения классов, их свойств, а также свойств отношений [6].

Примеры словаря OWL:

- subClassOf – декларирование одного класса элементов подмножеством другого класса элементов;
- equivalentProperty – предполагает, что одно свойство является эквивалентом другого;
- sameIndividualAs – предполагает, что один экземпляр является таким же, как другой экземпляр;
- maxCardinality – определяет максимальное число объектов, удовлетворяющих свойству.

Проектирование онтологии верхнего уровня для описания OLAP кубов. Онтология верхнего уровня определяет общие концепты OLAP куба и не зависит от прикладной области. Чтобы определить прикладную область, должна существовать предметная онтология или ее необходимо построить на основе онтологии верхнего уровня. Реальный OLAP куб описывается на основе предметной онтологии. В свою очередь, семантической поддержкой разработки предметной и прикладной онтологий является использование общих концептов предметной области.

Наивысшим уровнем онтологии верхнего уровня является сущность «OLAP куб» – далее Cube. Данная сущность на языке OWL описывается как класс.

```
<owl: Class rdf: ID="Cube">
<rdfs: SubClassOf rdf:
resource="http://www.w3.org/2002/07/owl#
Class"/>
</owl: Class>
```

Сущность Cube включает не менее одного элемента «Строка фактов» – далее FactRow. Сущность FactRow содержит атрибуты измерений и атрибуты мер [4], а также может содержать дополнительные атрибуты, которые не относятся ни к измерениям, ни к мерам, но представляют интерес при анализе. Данные атрибуты можно выносить в отдельные измерения, но это может привести к существенному увеличению объема данных из-за дублирования и усложнению проекта [7]. Атрибуты измерений относятся к измерениям OLAP куба и являются их ключами.

Сущность FactRow можно представить картежом (d1, ..., dn, m1, ..., mn, a1, …, an), где d1, ..., dn – члены измерений D1, ..., Dn, m1, ..., mn – члены мер M1, .., Mn, a1,…, an – дополнительные атрибуты, которые теоретически можно вынести в другие измерения.

Сущность FactRow на языке OWL описывается классом:

```
<owl: Class rdf: ID="FactRow">
<rdfs: SubClassOf rdf:
resource="http://www.w3.org/2002/07/owl#
Class"/>
</owl: Class>
```

Сущность Cube включает не менее одной сущности «Измерение» – далее Dimension. Сущность Dimension на языке OWL описывается классом:

```
<owl: Class rdf: ID="Dimension">
<rdfs: SubClassOf rdf:
resource="http://www.w3.org/2002/07/owl#
Class"/>
</owl: Class>
```

Необходимо отметить сущности, присущие любому OLAP кубу. Сущность «Иметь измерение» – далее hasDimension, сущность «Иметь строку фактов» – далее hasFactRow, сущность «Иметь меру» – далее hasMeasure определяют свойства, которыми обладает любой

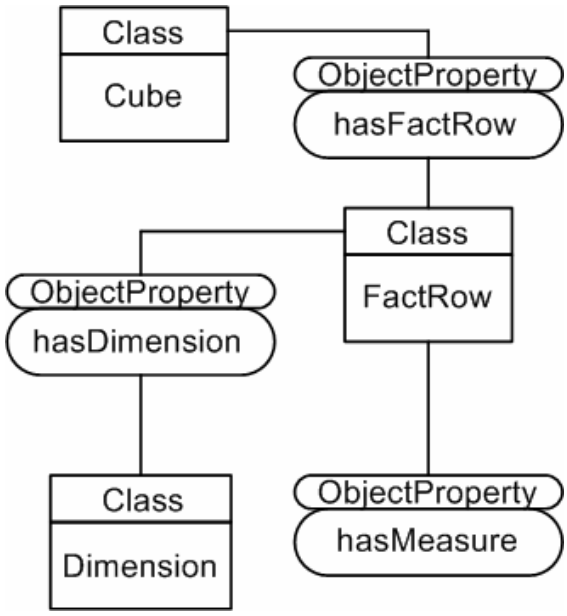
OLAP куб. Описание отмеченных сущностей на языке OWL соответственно имеет вид:

```
<owl: FunctionalProperty rdf:
ID="hasDimension">
<rdf: Type rdf:
resource="http://www.w3.org/2002/07/owl#
ObjectProperty"/>
<rdfs: Domain rdf: resource="#Cube"/>
</owl: FunctionalProperty >,
<owl: FunctionalProperty rdf:
ID="hasFactRow">
<rdf: Type rdf:
resource="http://www.w3.org/2002/07/owl#
ObjectProperty"/>
<rdfs: Domain rdf: resource="#Cube"/>
</owl: FunctionalProperty >,
<owl: FunctionalProperty rdf:
ID="hasMeasure">
<rdf: Type rdf:
resource="http://www.w3.org/2002/07/owl#
ObjectProperty"/>
<rdfs: Domain rdf: resource="#Cube"/>
</owl: FunctionalProperty >
```

С учетом отмеченных сущностей (свойств OLAP куба), определение сущности Cube на языке OWL имеет вид:

```
<owl:Class rdf:ID="Cube">
<rdfs:subClassOf
rdf:resource="http://www.w3.org/2002/07/owl
#Class"/>
<rdfs:subClassOf>
<owl:Restriction>
<owl:minCardinality
rdf:datatype="http://www.w3.org/2001/XMLSch
ema#int">1</owl:minCardinality>
<owl:onProperty>
<owl:ObjectProperty
rdf:about="#hasFactRow"/>
</owl:onProperty>
</owl:Restriction>
<owl:Restriction>
<owl:minCardinality
rdf:datatype="http://www.w3.org/2001/XMLSch
ema#int">1</owl:minCardinality>
<owl:onProperty>
<owl:ObjectProperty
rdf:about="#hasDimension"/>
</owl:onProperty>
</owl:Restriction>
<owl:Restriction>
<owl:minCardinality
rdf:datatype="http://www.w3.org/2001/XMLSch
ema#int">1</owl:minCardinality>
<owl:onProperty>
<owl:ObjectProperty
rdf:about="#hasMeasure"/>
</owl:onProperty>
</owl:Restriction>
</rdfs:subClassOf>
</owl:Class>
```

Графически онтологию верхнего уровня для описания OLAP кубов можно представить согласно рисунку.



Онтология верхнего уровня для описания OLAP кубов

Данная онтология не рассматривает иерархии, уровни иерархии измерений и члены уровней. Поясним необязательность рассмотрения данных сущностей в рассматриваемой онтологии.

Наиболее популярной моделью данных для хранилища данных является многомерная модель [4]. Такая модель может быть представлена в форме «звезды» (star schema), «снежинки» (snowflake schema), «созвездия» (fact constellation schema).

Схема «созвездие» является комбинацией схем «звезды» и/или «снежинки».

Схема «звезды» состоит из одной таблицы фактов и необходимого числа таблиц измерений. Каждое измерение в данной схеме представлено единственной таблицей с набором атрибутов.

Расширением схемы «звезды» является схема «снежинки». Отличием является возможная нормализация таблиц измерений. Таким образом, таблица измерений может быть расщеплена на несколько таблиц измерений.

В связи со сказанным онтология верхнего уровня для описания OLAP куба представлена одной сущностью Dimension, что соответствует ненормализованной таблице измерения многомерной модели. При необходимости сущность Dimension может быть разбита на несколько подобных сущностей, тем самым образуя иерархию измерений.

Результаты

1. Рассмотрен процесс проектирования онтологии верхнего уровня для описания OLAP кубов. Данное описание является унифицированным, не зависящим от конкретной пред-

метной области и может быть использовано для проектирования предметных онтологий.

2. Предложено описание OLAP кубов на языке OWL согласно спроектированной онтологии верхнего уровня.

Библиографический список

1. *T. Gruber*. "A translation approach to portable ontology specifications". Knowledge Acquisition volume 5, 1993. PP.199-220.

2. Data warehouses and OLAP: concepts, architectures, and solutions / *RobertWrembel* and *Christian Koncilia*, Idea Group Inc, 2007. PP.1-26.

3. S*ébastien Nedjar*, *Alain Casali*, *Rosine Cicchetti*, and *Lotfi Lakhal*. Emerging cubes for trends analysis in olapdatabases. In Il Yeal Song, Johann Eder, and Tho Manh Nguyen, editors, DaWaK, volume 4654 of Lecture Notes in Computer Science, Springer, 2007. PP.135-144.

4. *Jim Gray*, *Surajit Chaudhuri*, *Adam Bosworth*, *Andrew Layman*, *Don Reichart*, *Murali Venkatrao*, *Frank Pellow*, *Hamid Pirahesh*. Data Cube: A Relational aggregation operator generalizing group-by, cross-tab, and sub-totals export. Data mining and knowledge discovery, vol. 1, no. 1, 1997. PP.29-53.

5. *Клещев А.С.*, *Шалфеева Е.А.* Онтологии и их классификация. – Владивосток. ИАПУ ДВО РАН, 2005. – 19 с.

6. *Deborah L. McGuinness*, *Frank van Harmelen*. OWL Web Ontology Language Overview. W3C Recommendation, 10 February 2004. http://www.w3.org/TR/owl-features. Дата просмотра 15.11.2009 г.

7. *Teo Lachev*. Applied Microsoft Analysis Services 2005 and Microsoft Business Intelligence Platform. Prologika Press, 2005. -713 p.