

УДК 004.738.5

Декларирование фреймовых структур средствами языка Prolog

О.Н. Половикова
АлтГУ, г. Барнаул

На сегодняшний день является актуальными исследования в области структуризации и формализации данных. Такие исследования имеют широкое практическое применение для различных прикладных областей, в том числе и в области систем основанных на знаниях.

Итогом процесса структуризации какой-либо предметной области может быть создание новых структур данных или заполнение данными структур-шаблонов. Сами по себе структуры данных могут быть классическими (множества, деревья, графы, фреймы и т.д.) или иметь специфические особенности, отвечающие потребностям своей предметной области. Некоторыми структурами можно описывать только данные (массивы, стеки, множества, карты и т.д.). Но, на сегодняшний день, наиболее востребованными являются структуры знаний, в которые можно закладывать не только свойства или значения каких-то сущностей, предметов, но и отношения, правила, связи, функциональный аппарат, логику формирования нового знания и т.д.

Несмотря на разнообразие структур представления знаний, любая структура является производным продуктом от наложения выбранной модели представления знаний на предметную область исследования. Разработка структуры представления знаний, выбор способа её хранения, заполнение структуры знаниями должны подчиняться единой концепции в рамках реализации задачи представления и поиска знаний. Другими словами, решение подзадачи построения структуры знаний является лишь элементом системы к задаче поиска и интерпретации знаний. Поэтому выбор структуры знаний, конечно, зависит от специфики самих знаний, но специфика определяется и выстраивается для процессов обработки, поиска и вывода знаний.

В дальнейшем, после выполнения этапов моделирования и определения структуры знаний (модели знаний) можно сформировать базовые требования к аппаратно-программной реализации системы. Анализ этих требований позволит выбрать приемлемую технологию для реализации системы, программные среды и языки программирования для разработки.

В рамках данной работы обозначим преимущества и недостатки фреймов в качестве модели представления знаний (структура знаний), а также выделим возможности декларативного подхода для программирования систем, основанных на знаниях на примере среды программирования VisualProlog.

Фреймовые структуры получили широкую применимость благодаря заложенным в них возможностям работать как декларативными, так и процедурными знаниями. Особенность совместного использования задекларированных знаний (абстракций для структур, самих знаний оформленных в шаблонах) и задекларированных процедур их вывода и обработки (логику манипулирования и вывода знаний) закладывает дополнительный потенциал для разработки систем, и для их дальнейшего развития. Используя возможность добавления в систему нового знания можно расширять и углублять её функции, повышать релевантность поиска хранимых и выводимых (получаемых системой) знаний.

Универсальность фреймовых структур в качестве модели представления знаний, а также их схожесть с концептуальными структурами, на которые базируется и организуется человеческая память, делают их востребованными для систем, основанных на знаниях. Анализ теоретических работ и практических примеров применения фреймовой модели представления знаний для различных прикладных областей позволяет сократить время и повысить качество процессов проектирования и разработки систем.

При поиске знаний в системе, основанной на фреймовых структурах, в каждой конкретной ситуации можно задействовать актуальную часть всей сети или подсети фреймов, а можно активизирован только конкретный фрейм или его часть (слот). Можно гибко менять фокус, управляя конструкциями от абстрактного к конкретному знанию, от конкретного к абстрактному знанию. Это позволяет динамически настраивать глубину поиска знаний, управлять различного уровня абстракциями, гибко настраивать детализацию выводимых знаний.

Многопрофильность фреймовых структур обеспечивает описание широкого спектра свойств, ролей, понятий, отношений, процедур вывода и операций, также сценариев поведения и режимов состояний и деятельности сущностей какой-либо предметной области. Это подтверждается исследованиями [1, 3], в которых представлены примеры создания фреймов-структур (для обозначения объектов и понятий), фреймов-ролей (для обозначения ролевых обязанностей), фреймов-сценариев (для обозначения поведения), фреймов-ситуаций (для обозначения режимов деятельности, состояний). Возможности углублен-

ного описания самих хранимых знаний, а также конструкций поиска и построения нового знания предоставляют проектировщикам и разработчикам настраиваемую под потребности предметной среды базу для моделирования и создания систем, основанных на знаниях.

Также следует отметить, что фреймовое представление данных имеет схожую специфику с объектно-ориентированной методологией. Но, объектно-ориентированные модели представления знаний и языки программирования (ООЯ) не поддерживают различного рода процедур-демонов (автоматически запускаемые скрипты), которые реализуют определенные действия в системе при наступлении каких-то событий. Например, для фреймовой модели представления, в том числе и операционного знания, существенным является возможность автоматического выполнения процедур, связанных с заполнением (изменением) значения слота (или слотов) отдельного фрейма или с созданием нового фрейма. Поэтому совместно с объектно-ориентированными языками для реализаций процедур-демонов (присоединенных процедур) можно использовать низкоуровневые скриптовые языки.

Конструкции знаний на основе фреймов и фреймовых структур могут быть задекларированы средствами языка Prolog. В рамках данной работы выделим только некоторые возможности и особенности данного языка для представления и обработки фреймовых структур. Описать фреймы-шаблоны (определенного уровня абстракции) можно на основе доменных структур, которые поддерживает любые базовые и специально-определенные типы данных, в том числе списки и процедуры (набор одноименных фактов и правил). Ниже на рисунках представлена структура понятия *Группа студентов* и задекларирован пример кода на языке Visual Prolog для описания фрейма-понятия (шаблона): *Группа_Студенов*.

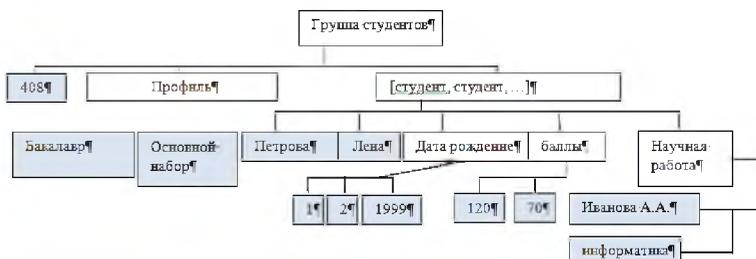


Рисунок 1 – Структура *Группа студентов*

```

domains
birthday=birthday (integer, integer, integer).
ball=ball(integer, integer).
nauch=nauch(string, string).
student=student(string, string, birthday, ball, nauch).
cp=cp(string, string).
group_stud(integer, cp(string, string), student*).

class facts
gr: (integer, cp| student*).

```

Рисунок 2 – Доменная структура *Группа студентов*

Создавать и заполнять на основе фреймов-шаблонов фреймы с конкретными данными предметной области можно с использованием визуальных форм. Иерархические связи от общего к частному можно определить с использованием механизмов наследования, от абстрактного к конкретному – на основе принадлежности фрейма-экземпляра к классу фреймов. При этом следует отметить, что операционные знания фреймовых моделей можно также описать средствами данного языка, используя в качестве элементов доменных структур наборы фактов и правил, которые также декларируются в Пролог-системе. Процедура фактов и правил может храниться в доменной структуре, пока не возникнет необходимость её обработать, вычислить.

Если опираться на общеизвестную классификацию систем в области искусственного интеллекта (предложенную С.С. Лавровым) по форме выдаваемого (формируемого) ответа, то среда Visual Prolog позволяет создавать программы, как с фактографическим, так и процедурными ответами. В конструкции языка заложена возможность формировать ответ в качестве цепочки фактов и правил, также можно предоставлять в качестве ответа и процедуру решений или рассуждений (в общем случае, требуемое решение может быть не найдено, но этапы рассуждений по поиску данного решения предоставлена пользователю). Поэтому системы фреймовых знаний, созданные в среде Visual Prolog, могут конкурировать с системами знаний, построенных специализированными языками разработки фреймов (FRL, KRL, фреймовая оболочка Карра, PILOT/2).

Библиографический список

1. Карпов В.Э. Фреймы. Пролог [Электронный ресурс] //электронная книга (версия от 19.05 2017). – Режим доступа: <http://rema44.ru/resurs/study/ai/present/L05-02-frame.pdf>, свободный.
2. Половикова О.Н. Формализация процесса построения решения с использованием списков для класса логических задач в программах на

языке Пролог // Известия Алтайского государственного университета. – Барнаул, 2011. – №1/1 (69). – С. 117–120.

3. Фреймовая модель знаний [Электронный ресурс] // (версия от 19.05 2017). – Режим доступа: <http://helpiks.org/7-30113.html>, свободный.

УДК 519.65

Исследование статистических данных на выбросы

И.В. Пономарев
АлтГУ, г. Барнаул

При обработке статистических данных появляется необходимость оценить их на предмет выбросов. Выброс – это результат измерения, выделяющийся из общей выборки. Выбросы способны исказить статистики и результаты статистического моделирования.

В данной статье исследуются методы определения выбросов как для классических статистических регрессионных моделей, так и для моделей нечеткой регрессии.

Рассмотрим линейную регрессионную модель

$$y = a_0 + a_1x_1 + \dots + a_kx_k + \varepsilon, \quad (1)$$

где y – зависимая переменная; x_i ($i = \overline{1, k}$) – независимые переменные; ε – ошибка; a_i ($i = \overline{0, k}$) – параметры модели.

Наиболее часто применяемым методом оценки параметров модели (1) является метод наименьших квадратов (МНК), который является очень чувствительным к выбросам, т.е. наличие в исходной выборке выбросов может привести к неверным выводам о наблюдении процесса и ставит под угрозу результаты статистического анализа модели.

В работе [1] для исследования выборки на выбросы предлагается использовать статистику Стьюдента

$$t_i = \frac{\tilde{y}_i - \tilde{y}_i^*}{s_i^2 \sqrt{1 + X_i(X_i^T X_i)^{-1} X_i^T}}, \quad (2)$$

где $\tilde{y}_i, \tilde{y}_i^*$ – прогнозные значения зависимой переменной с учетом и без учета i -го наблюдения; X_i – матрица наблюдений без i -го наблюдения; s_i^2 – оценка дисперсии ошибки регрессии без учета i -го наблюдения. Данная статистика имеет распределение Стьюдента с $N - k - 2$ степенями свободы (N – объем выборки). Если гипотеза подтвержда-