УДК 004

АКТИВНАЯ РАССЛОЕННАЯ ТЕРМИНОЛОГИЧЕСКАЯ СИСТЕМА

Сергей Геннадьевич Маслов, кандидат технических наук, доцент Удмуртского государственного университета

Анатолий Петрович Бельтюков, доктор физико-математических наук, профессор, заведующий кафедрой Удмуртского государственного университета

Аннотация

В статье рассмотрены некоторые аспекты проявления активности расслоенной терминологической системы, которая реализуется через разнообразие постановок задач и организацию компьютеринга. В этом случае, исследователь может строить индивидуальную информационно-технологическую сферу для решения возникающих у него проблем, а также получить более совершенную среду для обмена профессиональным опытом.

КЛЮЧЕВЫЕ СЛОВА: терминологическая система, индивидуальная информационно-технологическая сфера, естественный и искусственный компьютеринг, системное моделирование.

THE ACTIVE STRATIFIED TERMINOLOGICAL SYSTEM

Sergey Gennadyevich Maslov, PhD in Technical Sciences, associate professor of the Udmurt State University

Anatoly Petrovich Beltukov, Doctor of Physical and Mathematical Sciences, professor, head of the Udmurt State University Department

Abstract

The article covers some aspects of display of the active stratified terminological system, which is realized through a variety of problems' formulation and the organization of computing. In this case, the researcher can build an individual information-technological sphere to solve different problems and to receive a better environment for the professional experience exchange.

KEYWORDS: terminological system, individual information-technological sphere, natural and artificial computing, system modeling.

Введение

Жизненная среда человека в своем непрерывном развитии требует выделить индивидуальную информационно-технологическую сферу (ИИТ-сфера) как особый объект, коренным образом меняющий жизненные процессы [4]. Если искусственная и естественная сфера связаны с материальными и энергетическими потоками, то ИИТ-сфера связана прежде всего с системным моделированием и компьютерингом, т.е. с организацией мышления, созданием объектов и процессов материального и идеального миров. Именно эти процессы позволяют человеку разумно использовать ресурсы и высвободить время для своего развития и создания разнообразия жизненной среды. ИИТ-сфера человека позволяет ему лучше и более своевременно осознавать и более эффективно решать возникающие перед ним проблемы.

Решение любой сложной проблемы, в свою очередь, начинается с ее обсуждения, с выявления системы понятий, описывающей контекст возникновения и суть проблемы. Система понятий осознается через терминологию и определения, утверждения, гипотезы (*терминологическую систему*), представляющие собой проекцию видения проблемы конкретным человеком (или другими людьми). Основные знания для решения проблем накапливаются в виде текстов (или корпусов текстов) на естественных и искусственных языках. Концептуальная модель терминологической системы рассмотрена нами в работе [6], однако проблемы, связанные с построением, развитием и использованием терминологической системы, требуют более детального рассмотрения. Схематически необходимо уточнить смысл и реализацию переходов в терминологической системе в виде следующей простой цепочки понятий:

термины (атомы со свойствами) – ассоциации – активизации.

Основная цель этой статьи – рассмотреть проявление активности терминологической системы.

Структура терминологической системы

Предлагается способ построения терминологической системы с помощью ответов на заранее подготовленную систему вопросов. Для формализации этого используется функция g , описывающая эти вопросы и возможные ответы на них [6]:

$$g: M \times S \times K \times V \times Q \rightarrow W. \quad (1)$$

где

M – модификаторы вида формализации (метафорическая, концептуальная, математическая, ...);

S – модификаторы вида системных представлений (морфологический, функциональный, либернетический, ...);

K – модификаторы слоя знаний (дескриптивный, конструктивный, ...);

V – модификаторы форм представления (моносенсорные, полисенсорные, ...);

Q – система вопросов;

W – множество возможных ответов на задаваемые вопросы.

При формировании ответов на вопросы могут возникать *противоречия*, которые необходимо разрешать.

Особенности построения и активности терминологической системы

При построении терминологической системы можно выделить ряд важных моментов:

- построение форм представления (лингвистические, полисенсорные);
- виды форм вычисления (символьные, численные, с индексными структурами);
- *двухэтапность* вычислений;
- внутренний и внешний *компьютинг*, проявление симбиоза;
- формирование разнообразия постановок задач;
- постановка задачи как импульс активизации процессов в терминологической системе, а, в частном случае, как замена традиционной формы технического задания в проектировании систем.

Поскольку ключевым фактором в данном перечислении является понятие разнообразия постановок задач как средство активизации терминологической системы, перейдём к этому вопросу.

Разнообразие постановок задач в рамках расслоенной терминологической системы можно построить на следующих принципах:

- *Принцип дивергенции.* Оптимальное по эффективности и продуктивности использования распределение по взаимодействующим слоям описаний с фиксацией уровней абстракций и форм представления. Применение *принципа дивергенции* к таким задачам вынуждает нас разделить описание обстановки, в которой решается задача, на дескриптивное и конструктивное, экспериментальное и теоретическое, прикладное и фундаментальное знание. При этом принимаются во внимание разные формы представления знаний (например, вербальные и невербальные, зрительные и звуковые, полисенсорные) и уровни абстракции. Нахождение баланса между этими расслоениями, уровнями и формами и является основанием для оптимизации постановки задачи.
- *Принцип варьирования (либернетический принцип).* Варьирование объектов системы в трёх контекстах: *известность – неизвестность, инвариантность – вариативность, степень свободы – ограничение.* Варьирование (ин)вариантности, например, входа, приводит к разделению задач ещё на два класса: задачи с фиксированным входом (например, конкретные входные данные заранее известны) и задачи с меняющимся входом (построение программ, обрабатывающих различные данные). Варьирование между степенью свободы и ограничением также приводит к расщеплению класса задач, но несколько иным образом. Например, ограничение может быть

функциональной зависимостью между входными и выходными данными. Построение *двойственных* объектов. Пример *принципа двойственности* – взаимная смена ролей между объектом и инструментом.

- *Принцип целостности.* Целостность в рамках *системной ситуации*, отражающая синтез *субъективного* и *объективного*, *естественного* и *искусственного*, *живого* и *косного*. *Принцип целостности* хорошо иллюстрируется созданием человеко-машинных систем.
- *Принцип оценки (меры и измерения).* Использование меры и оценка сложности: *цена – ценность*, *(бес)полезность – (без)вредность – нейтральность*. Например, при программировании, *принцип оценки* может заключаться как в оценке ресурсов, необходимых при работе программы, при решении задачи, так и в оценку самого качества полученного решения.
- *Принцип моделирования.* Мыслительный и компьютерный эксперимент – *компьютерный эксперимент* (оптимизация материализации решения). *Моделирование* нужно, с одной стороны – для понимания обстоятельств задачи, с другой стороны – для сокращения материальных затрат и времени.
- *Принцип изменения.* Генерация дополнительного контекста, переход в другой слой или переформулирование задачи при преодолении препятствий и устранении ошибок (неудач). *Принцип изменения* диктуется тем, что наши знания несовершенны и любое их использование может приводить к ошибкам и неудачам. Чтобы перевести обстановку решения задачи из *косного* состояния в «живое», необходимо реализовать идею *инвариантного цикла деятельности*, который организует информационные потоки путём структуризации информационной среды на основе вопросов и модификаторов системы знаний.

Трудность в осознании постановки задачи состоит в том, что при их решении в рамках выделенных ресурсов необходимо как можно раньше понять, что нас ожидает в процессе решения поставленной задачи: *успех* или *неудача*.

Наиболее распространённый подход к классификации постановок задач – выделение *входа*, *выхода* и *процесса* [3, 7]. Варьирование известности приводит, в частности, к таким типам задач, как *проверка* (всё известно) или *решённая задача* по классификации [3], *синтез процесса* (процесс неизвестен, вход и выход известны) или *задача проектирования* по классификации [3]. *Вход* при этом может получаться как ответ на вопрос «что» с ответом на

вопрос «когда» - «раньше». *Процесс* – ответ на вопрос «как». В работе [3] дана классификация задач, представленная в таблице 1.

Таблица 1. Классы задач

№	Вход	Процесс	Выход	Класс задач
1	+	+	+	решенные задачи
2	+	+	-	задачи прогнозирования
3	+	-	+	задачи проектирования
4	+	-	-	не поставленные задачи (не известны цель и метод)
5	-	+	+	задачи распознавания образов
6	-	+	-	не поставленные задачи (не известны цель и причина)
7	-	-	+	не поставленные задачи (не известны причина и метод)
8	-	-	-	нерешенные задачи

В работе [3] считается, что «задача поставлена», если заданы 2 элемента системы; и «задача не решена», если неизвестны три элемента.

Также выделяются следующие группы методов управления знаниями:

- методы сбора и хранения знаний (вход);
- методы обработки и интегральной оценки знаний (процесс);
- методы управления реализацией знаний (выход).

Разнообразие постановок задач определяется большим числом комбинаций вопросов (Q) с различными модификаторами (M , S , K , V), которые являются необходимыми средствами организации знаний.

Решение задач часто состоит в комбинации построения недостающего элемента и поиска его в окружающей среде. При этом применяются такие способы построения, как комбинация, трансформация, отсечение лишнего. Иногда нужно выйти на метауровень – изменить саму окружающую среду.

Рассмотрим разнообразие задач с точки зрения реализации компьютеринга. Здесь надо выделить качественные и количественные подходы к формулировке и решению задач, а также – символьные и численные подходы. Качественные и численные подходы связываются понятием меры (здесь мера понимается как количественная граница качественного изменения). Символьный компьютеринг можно считать частным случаем качественного компьютеринга, количественный компьютеринг – частным случаем численного. Здесь мы считаем, что число – не обязательно количество предметов или результат измерения. Например, в задачах шифрования могут использоваться числа, не имеющие никакого количественного или измерительного смысла. Кроме того, приходится

использовать такие абстрактные «неколичественные» числа как комплексные, кватернионы, трансфинитные и т.д.

В продолжение этого следует отметить проявление разнообразия в математических средствах: теория графов и алгебра – теория категорий, логика, геометрия, топология. Между этими средствами имеется сложная система взаимных переходов. Эти средства также характеризуются различными оценками ресурсоёмкости компьютеринга. Это даёт нам основания выбора адекватных и эффективных средств решения задачи.

Далее используется так называемая *крупноблочная спецификационная нотация*, которая соотносится с традиционной логической нотацией следующим образом. Спецификационной формуле:

$$(x_1: X_1, \dots, x_n: X_n, A_1(x_1, \dots, x_n), \dots, A_m(x_1, \dots, x_n) \Rightarrow y_1: Y_1, \dots, y_k: Y_k B_1(x_1, \dots, x_n, y_1, \dots, y_k), \dots, \\ B_l(x_1, \dots, x_n, y_1, \dots, y_k) | \dots | \\ z_1: Z_1, \dots, z_k: Z_k C_1(x_1, \dots, x_n, z_1, \dots, z_k), \dots, C_l(x_1, \dots, x_n, z_1, \dots, z_k)) \quad (2)$$

соответствует логическая формула:

$$\forall x_1: X_1 \dots \forall x_n: X_n, (A_1(x_1, \dots, x_n) \& \dots \& A_m(x_1, \dots, x_n) \Rightarrow \exists y_1: Y_1 \dots \exists y_k: Y_k (B_1(x_1, \dots, x_n, y_1, \dots, y_k) \& \dots \& \\ B_l(x_1, \dots, x_n, y_1, \dots, y_k)) \vee \dots \vee \\ \exists z_1: Z_1 \dots \exists z_k: Z_k (C_1(x_1, \dots, x_n, z_1, \dots, z_k) \& \dots \& C_l(x_1, \dots, x_n, z_1, \dots, z_k))). \quad (3)$$

Спецификационная нотация будет использоваться нами как более адекватная решаемой задаче.

Контекст постановки задачи синтеза в наиболее простом виде (см., например, [1]) можно первоначально представить так:

$$f: (a: A \Rightarrow B(a)), \quad (4)$$

где A – тип входного объекта, a – имя экземпляра входного объекта, $B(a)$ – тип выходного объекта (вообще говоря, зависящий от a), f – преобразователь входного объекта в выходной (правило, процесс, трансформация и т. п.). Надо заметить, что в этой записи учтены не все детали постановки задачи. Во-первых, нас интересует структура этого преобразователя. Предполагается, что для функционирования этого преобразователя требуется некоторый *исполнитель* (например, универсальная машина). Будем обозначать его через u . Исполнитель в своей работе использует *программу*, обозначаемую далее через p . Эта программа может ссылаться на некоторые инструменты, обозначаемые далее через i , ресурсы, обозначаемые далее через r , и *собственные данные* (внутренние данные [см. работы 8 и 9]), обозначаемые далее через d .

В наиболее общем виде описанную задачу будем представлять следующим образом (с использованием нотации [1]):

$$u(p,i,r,d): (a: A, R(a) \Rightarrow b: B(a), Pr(a,b), R''(a,b) | N(a), R'(a)), \quad (5)$$

где $R(a)$ – описание ресурсов, выделенных, и реально потребляемых при решении задачи, $Pr(a,b)$ – свойства выхода (ограничения на b), $R''(a,b)$ – описание возвращаемых ресурсов, $N(a)$ – описание возможных неудач при решении задачи, $R'(a)$ – описание ресурсов, возвращаемых в случаях неудач. Таким образом, преобразователь f можно представить в виде кортежа: $\langle u,p,i,r,d \rangle$, или при фиксированном u – в виде кортежа $\langle p,i,r,d \rangle$. Кроме того, задача может решаться под управлением некоторых мер μ , предназначенных для оптимизации решения и процесса решения. Примерами таких мер могут служить количественные оценки надёжности, точности и эффективности решения, оценки времени работы, степени достижения решения. В случае, если a – поток, мера может отражать интенсивность процесса преобразования. Эта мера, вообще говоря – функция от аргумента преобразования, это можно изобразить так:

$$(a: A, R(a) \Rightarrow (\mu(a)) b: B(a)), \quad (6)$$

как было введено ещё в [2].

Строящееся терминологическое пространство (или среда) позволяет перейти к эффективной реализации операций поиска, навигации и синтеза не просто для удовлетворения когнитивных потребностей, а прежде всего креативных и конструктивных потребностей, т.е. к построению новых объектов с заданными свойствами. Фактически любая возникающая проблема в рамках терминологической среды либо актуализирует и связывает некоторую группу терминов, либо выявляет противоречия и новые проблемы. Разрешение этих ситуаций осуществляется через переход в другой терминологический слой и путем переформулирования проблемы, либо осуществляется через расширение (введение новых сущностей) или изменение связей терминов в существующем слое.

Рассматриваемое расслоённое терминологическое пространство допускает несколько типов переходов. Основными типами переходов являются:

- переход по аналогии;
- переход к конкретизации/обобщению;
- переход форма-содержание.

Выпуск подготовлен по итогам Международной конференции по фундаментальным проблемам устойчивого развития в системе природа – общество – человек (24 и 25 октября 2011 г., проект РФФИ №11-06-06128-г).

При этих переходах могут использоваться определённые операторы, например:

- преобразования симметрий;
- переход к двойственным структурам;
- переход к противоположности;
- разбиение и агрегация;
- функциональное и либернетическое оснащение.

Все переходы должны сопровождаться анализом степеней свобод и их управлением для достижения поставленных и возникающих целей.

Кроме того, решение поставленной задачи будем рассматривать в конструктивном стиле, используя явно и неявно группу вопросов и ответов.

Возможные конструктивные переходы при преобразовании некоторых исходных объектов в результирующие опишем в виде следующей конструкции:

$$p:(x:X \Rightarrow y:Y(x)/n:N(x)):c, \quad (7)$$

где

X – исходная область объектов;

x – исходная точка отправления;

$Y(x)$ – область результата (иногда зависящая от точки отправления);

p – метод или правила способствующее переходу $x:X$ к $Y(x)$ и $p \in P$;

c – метод или факторы препятствующие переходу $x:X$ к $Y(x)$ и $c \in C$;

N – возможные типы неудач перехода;

n – обозначение конкретной неудачи.

Можно выделить следующие типы неудач перехода $x:X$ к $Y(x)$, частично придерживаясь также классификации в работе [1, 5]:

N_I – объект вообще не создан, например, если все ресурсы израсходованы;

N_{II} – объект создан, но не надлежащего качества;

N_{III} – создан объект требуемого качества, но неэффективным образом;

N_{IV} – создан объект требуемого качества, но система ценностей и потребностей субъекта изменилась;

N_V – объект и не мог быть создан в данных условиях, например, в случае некорректного использования синтезированного решения;

N_{VI} – в процессе выполнения процедуры выяснилась её принципиальная незавершимость в данных условиях;

N_{VII} – в процессе функционирования процедуры выявился дефект в логике работы системы, построившей саму процедуру.

Это означает, что должна быть построена формальная процедура q для классификации неудач:

$$q:(x:X(x),N(x) \Rightarrow N_I(x)|N_{II}(x)|N_{III}(x)|N_{IV}(x)|N_V(x)|N_{VI}(x)|N_{VII}(x)). \quad (8)$$

Могут существовать и в принципе не обнаруживаемые ошибки, например, несоответствие формальной постановки задачи конкретной области, при которой разрешающая процедура создаёт лишь иллюзию решения. К сожалению, такие ошибки в постановке задачи учесть невозможно.

Если под задачей понимать определение неизвестных компонент конструктивного перехода, то фактической средой реализации будет, выделенная часть общей среды реализации (E), которая является организованной совокупностью информации и знаний о X , Y , P , C . Каждый конструктивный переход или его компоненту на метафорическом уровне можно представить в виде ствола дерева (или нейрона), у которого есть корни (из чего он построен или откуда принимает информацию) и крона (где он используется или куда выдает свою информацию).

Если выделены степени свобод в преобразованиях и в преобразуемых объектах, то можно выделить операторы управления степенями свобод. Каждый такой оператор h осуществляет замену некоторого объекта f в нашей системе a на некоторый объект g , переходя от подцели G_i к подцели G_k :

$$h:(G_i(a(f)) \Rightarrow G_k(a(f))):c, \quad (9)$$

где c – противодействие этому переходу.

Необходимо также рассмотреть уровни управления степенями свобод для построения системы. Каждый уровень включает редукторы и генераторы степеней свобод.

Параметрический уровень включает непосредственное изменение значений параметров, возможно используя скрытую структуру их взаимосвязи, например, включает фиксации и регуляции параметров системы:

$$x_\alpha \rightarrow const, \quad const \rightarrow x_\alpha. \quad (10)$$

На этом же уровне реализуется управление посредством отрицательной и положительной обратной связи (реакции и рефлексy).

Структурный уровень непосредственно оперирует элементами и связями между ними, связывая группы переменных принадлежащие элементам (ξ_{im} , i – номер элемента (e_i), m – номер переменной в элементе) или связям (ζ_{kn} , k – номер связи (r_k), n – номер переменной в связи). Можно «рубить» $e_n \rightarrow e_i$, e_j или «склеивать» элементы e_i , $e_j \rightarrow e_n$,

изменяя состав переменных. Можно добавлять или удалять связи (r_v), вводя синергии и инварианты или освобождая ранее взаимосвязанные переменные. Интересно, что для достижения цели требуется динамический способ изменения активных и пассивных переменных.

Базисный уровень, сохраняя форму, включает изменение характеристик среды реализации (субстанции), например, изменение механической среды вычислений на электронную или химическую. Это осуществляется путем изменения базиса переменных:

$$x_i \rightarrow y_k, \quad (11)$$

или изменения области допустимых значений переменных, например:

$$x_i \in [a, b] \rightarrow x_k \in [c, d]. \quad (12)$$

Динамически может меняться точка зрения (или наблюдения):

$$a = T(b), \quad a \in A, \quad b \in B, \quad (13)$$

где a, b – кортежи параметров наблюдаемых объектов;

B, A – первое и второе пространства параметров соответственно;

T – оператор преобразования точек зрения.

Динамически можно также изменять функциональное насыщение (зависимости между частями) системы, т.е. при сохранении входных и выходных параметров (u, v), изменять функции их преобразования (например, f на g), изменяя тем самым надежность или эффективность вычислений:

$$(f(u) = v) \rightarrow (g(u) = v). \quad (14)$$

Рефлексивный уровень позволяет перевести объекты среды из пассивного состояния в активное, т.е. использовать знание о среде и самих себе, что позволяет организовывать «цепные реакции» для эффективного достижения целей: собственных локальных для объекта и глобальных достигаемых совместно с субъектом.

Примером рефлексии могут служить конструктивные электронные издания, которые активно перестраивают свой материал под решаемую проблему и уровень знаний субъекта.

Важно, что в рамках всех четырех уровней организации степеней свобод отдается приоритет программному управлению, а не простому механизму «стимул – реакция».

Кроме того, степени свободы могут появляться или исчезать при непосредственном добавлении или удалении параметров x , элементов e и связей r .

Таким образом, формирование разнообразия постановок задач и реализация их решений с помощью процессов поиска, навигации, синтеза (вывода, построения) и

управления позволяют использовать активную расслоенную терминологическую систему как основу реализации ИИТ-сферы.

Заключение

В широком понимании активность терминологической системы – это её компьютеринг, реализующий процессы решения задач предметной области и выполняемый в разных средах. В решении задач важно согласование с особенностями субъекта: человек должен находить своё место в процессе и результатах решения задачи. Терминологическая система связывает воедино образовательную, научную и инженерную деятельность за счёт целостности системы знаний.

Литература

1. Бельтюков, А.П. Малые сложностные классы и автоматический дедуктивный синтез алгоритмов// Известия института математики и информатики УдГУ: вып. №2. – Ижевск: УдГУ, 1995. – с. 3 – 89.
2. Бельтюков, А.П. Формальная теория для порождения правильных программ заданной вычислительной сложности//Всес.конф. «Методы математической логики в проблемах искусств интеллекта и систематическое программирование»: часть 1 – Вильнюс, 1980. – с. 64 – 66.
3. Большаков, Б.Е., Шамаева, Е.Ф. Системный анализ методов управления знаниями в области устойчивого развития// Электронное научное издание «Устойчивое инновационное развитие: проектирование и управление»: том 4 (2009). – с. 39 – 55. URL: <http://www.rypravlenie.ru/?cat=12>.
4. Маслов, С.Г. Индивидуальная ИТ-сфера//Знания-Онтологии-Теории (ЗОНТ-09): матер. Всероссийской конф. с междун. уч., том 2. – Новосибирск: Институт математики им.С.Л.Соболева СО РАН, 2009. – с. 110 – 118.
5. Маслов, С.Г. Системный подход к формированию среды программирования//Интеллектуальные системы управления. – М.: Машиностроение, 2010. – с.41 – 47.
6. Маслов, С.Г., Бельтюков, А.П. Расслоенная терминологическая среда для научных исследований и образования//Электронное научное издание «Устойчивое инновационное развитие: проектирование и управление»: том 7 вып. № 1 (2011). – с. 1 – 17. URL: <http://www.rypravlenie.ru>.
7. Непейвода, Н.Н., Свириденко, Д.И. К теории синтеза программ//Математическая логика и теория алгоритмов. – Новосибирск: Наука. 1982. – с. 159 – 175.
8. Beltiukov, A.P. A strong induction scheme that leads to polynomially computable realizations// Theoretical Computer Science: 322 (2004). – p. 17 – 39.
9. Beltiukov, A.P. Intuitionistic formal theories with realizability in subrecursive classes//Annals of Pure and Applied Logic: vol. 89, 1997. – p. 3 – 15.

Работа выполнена при финансовой поддержке РФФИ (проект № 11-07-00640-а).