

Рис. 12. Спектрограмма обработанного сигнала, график оценки ERLE и результат

NOVIKOV A.E., PETROVSKIY A.A. VHDL Realization of the adaptive FIR-filter on distributed arithmetics

The paper describes a synthesis approach of the FIR-filters based on the distributed arithmetics and the least-squares coefficients adaptation. A device is implemented on the Xilinx Spartan 3 FPGA and can be used in such applications as electrical echo-cancellation in telecommunications, etc. A discussion of the achieved results is present.

УДК 004.822

Ивашенко В.П.

АЛГОРИТМЫ ВЕРИФИКАЦИИ И ИНТЕГРАЦИИ БАЗ ЗНАНИЙ

Введение. Для широкого применения интеллектуальных систем, способных повысить качество решения прикладных задач, необходимо большое количество баз знаний. Быстрой разработке достаточного количества баз знаний могло бы способствовать наличие средств разработки интеллектуальных систем, обеспечивающих разработку и проектирование различных компонентов интеллектуальной системы, включая базу знаний. Среди средств, которые могут рассматриваться в качестве основы для разработки баз знаний, можно выделить: 1) оболочки экспертных систем (CLIPS (FuzzyCLIPS, DYNACLIPS, WxCLIPS), SOAR, OPS83, RT-EXPERT, MIKE, BABYLON, WindExS, ES; ACQUARE, Easy Reasoner, ECLIPSE, EXSYS Professional, SIMER+MIR, AT ТЕХНОЛОГИЯ, CAKE v2.0) [1]; 2) инструментальные пакеты для разработки экспертных систем (G2, ART, KEE, Knowledge KRAFT); 3) системы, ориентированные на обработку онтологий – Protégé, WebOnto, OntoEdit, WebODE, OilEd, OntoLingua.

Достоинствами приведенных инструментальных средств являются: поддержка представления знаний различного вида различными моделями представления знаний в рамках одной системы; наличие средств визуального проектирования баз знаний; наличие средств верификации базы знаний, включая проверку на непротиворечивость; возможность монотонного расширения базы знаний; наличие средств интеграции баз знаний; наличие средств поддержки обмена данными с внешней средой, включая средства обмена данными в реальном времени. Однако для всех указанных средств характерны следующие недостатки: в силу различных ограничений велики сроки разработки баз знаний (отсутствие развитых технологий разработки); узок круг инженеров баз знаний (из-за высоких стартовых требований к разработчику) – от разработчика требуется вла-

дение специальными знаниями по моделям и языкам представления знаний; не полностью решен вопрос интеграции баз знаний.

В работе предлагается подход к созданию технологии проектирования баз знаний, в основе которой лежат следующие положения: модульное проектирование баз знаний, интеллектуализация средств поддержки проектирования баз знаний и семантическое представление знаний. Эта технология включает: унифицированную модель баз знаний; библиотеку ip-компонентов баз знаний и инструментальные средства проектирования баз знаний; методику проектирования и интеграции баз знаний.

Унифицированная модель баз знаний. Одной из основных проблем разработки унифицированной модели базы знаний является проблема разработки единых мер, критериев оценки качества представленных знаний. Проблема заключается в том, что существующие критерии оценки, разработанные для отдельных моделей представления знаний, не определены для других моделей; критерии непротиворечивости, разработанные для теорий в логических моделях не всегда просто применить к моделям, в которых теория в явном виде отсутствует или знания в которых носят императивный характер, а не декларативный; критерии нечеткости представленной информации, разработанные для отдельных нечетких множеств и предикатов, не всегда просто применить в случаях, когда имеются сложноструктурированные знания нечеткого характера; оценки количества информации, разработанные для традиционных способов кодирования, могут иметь различные значения в зависимости от того, как представлять знания такими способами. Поэтому требуется иметь систему мер, которые работают для любых видов знаний в

Ивашенко В.П., аспирант Белорусского государственного университета информатики и радиоэлектроники. Беларусь, БГУиР, 220013, г. Минск, ул. П. Бровки, 6.

рамках разрабатываемой модели и не зависят от особенности ее программной реализации. Тогда уже, исходя из такой системы, можно решать проблему унифицированного представления знаний, которые бы отвечали некоторым общим критериям качества. Унифицированная модель представления знаний использует: унифицированное кодирование знаний, унифицированное представление знаний, описание структуры базы знаний, методы оценки качества и сравнения баз знаний.

Основой предлагаемых средств представления знаний является Semantic Code (SC-код) – способ семантического кодирования, обеспечивающий унифицированное кодирование знаний [2]. Особенности SC-кода являются: простой алфавит, содержащий узлы и дуги, простой синтаксис, базовая теоретико-множественная интерпретация. Базовая теоретико-множественная интерпретация (семантика) заключается в том, что все узлы и дуги SC-кода трактуются как знаки множеств [2], кроме того дуги являются знаками пар отношения принадлежности (или непринадлежности) элементов множеству. Принадлежность может быть как временной, так и постоянной.

Унификация представления знаний обеспечивается семейством совместимых sc-языков, использующих унифицированный способ семантического кодирования (SC-код), которые обеспечивают представление базовых математических абстракций – множеств, чисел, отношений; знаний, которые могут быть представлены различными моделями представления знаний – фреймов, продукций, логических утверждений, нейросетевых моделей; знаний об информации, хранимой на различных медиа-носителях; информации о временных и причинно-следственных взаимосвязях; знаний, описывающих структуру и специфицирующих базы знаний, включая описание целей, вопросов и задач, на решение которых ориентированы интеллектуальные системы. Основным принципом построения sc-языков является представление понятий, соответствующих основным классам объектов, описываемых sc-языком, и отношений между этими объектами – ключевыми узлами такого sc-языка.

В частности, принципами построения гипермедийного sc-языка являются: рассмотрение понятий различных информационных конструкций, рассмотрение отношений между конструкциями одного типа, рассмотрение отношений между конструкциями разных типов (синонимия, трансляция, идентификация). Под синонимичными элементами или синонимами далее в статье будем понимать знаки sc-языка, значения которых совпадают.

Принципами логического sc-языка описания нестационарных структур являются: использование временных (ситуативных, нестационарных) дуг принадлежности для представления ситуативных множеств; использование понятия состояния для представления статичных фрагментов предметной области; использование временных отношений между ситуативными множествами и состояниями; использование причинно-следственных отношений между ситуативными множествами и состояниями.

Принципами sc-языка целей, вопросов, задач и обобщенных задач являются: трактовка описаний целей и вопросов как открытых формул со свободными переменными, значения которых надо найти; трактовка вопросов, как целей частного вида, допускающих как минимум два различных целевых состояния – ответа; трактовка понятия задачи, как отношения между описанием одного или нескольких допустимых исходных состояний и описанием одного или нескольких целевых состояний.

Множество всевозможных объединений текстов этих языков рассматривается как интегрированный sc-язык представления знаний.

База знаний – связанная структурированная информационная конструкция, структура которой состоит не менее чем из одного атомарного раздела, каждому из которых принадлежит свой, описываемый в этом разделе, ключевой элемент этой конструкции, причем в описании хотя бы одного ключевого элемента присутствует его внешнее обозначение (терм). Структура базы знаний включает:

- декомпозицию разделов базы знаний;
- непустое множество знаков разделов базы знаний;
- непустое множество знаков внешних обозначений;

- множество знаков информационных конструкций, не являющихся текстами sc-кода;
- непустое множество ключевых элементов базы знаний, включая множества знаков предметов, классов, отношений, атрибутивных отношений, утверждений, определений, теорий, задач, доказательств, программ.

Элементом базы знаний является узел или дуга SC-кода, поэтому все элементы базы знаний являются знаками. С целью поддержки модульного проектирования баз знаний и обеспечения их совместимости в модель вводятся необходимые отношения, соответствия и операции.

Для множества баз знаний определяется двуместное соответствие интеграции, которое каждой паре баз знаний сопоставляет однозначное (гомоморфное) отображение множества всех элементов (знаков), формирующих эти базы знаний, на множество всех элементов результирующей базы знаний.

Рассмотрим задачу интеграции двух баз знаний, задаваемую некоторым подмножеством упомянутого выше соответствия интеграции двух баз знаний. Пусть все знаки атомарных и неатомарных разделов исходных баз знаний отображаются им соответственно на знаки атомарных и неатомарных разделов результирующей базы знаний, а все знаки элементов атомарных разделов и ключевых элементов исходных баз отображаются на соответствующие (синонимичные) знаки элементов атомарных разделов и ключевых элементов результирующей базы знаний. Кроме того, результирующая база знаний имеет неатомарный раздел, который не соответствует ни одному из разделов исходных баз знаний, в состав декомпозиции которого входят все разделы исходных баз знаний, являющиеся наиболее и одинаково удаленными по отношению декомпозиции от атомарных разделов каждой из исходных баз знаний. Такое подмножество описывает один из простейших результатов интеграции двух баз знаний.

Имея такой результат интеграции или произвольно взятую базу знаний, далее можно переходить от неё путём интеграции требуемой информации к базе знаний, которая имеет более высокое качество, или к оптимизированной базе знаний. Следовательно, основная проблема, которая возникает при интеграции баз знаний, заключается в получении как можно более качественной результирующей базы знаний. С этой целью для каждой базы знаний задаётся множество функций критериев качества, каждая из которых определена для этой базы знаний и принимает значения из некоторого множества величин, на котором определено некоторое отношение предпочтения (отношение порядка). Эти функции используются методами оценки качества и сравнения базы знаний. К таким методам относятся следующие: методы оценки объема базы знаний по различным типам элементов, методы проверки связности различных онтологических уровней базы знаний, методы оценки числа пар потенциальных синонимов в базе знаний, методы верификации стилистики исходных текстов базы знаний, методы выявления ошибок в базах знаний.

Каждому типу раздела соответствует множество вопросов, на которые требуется получать утвердительный ответ в виде конструкции из элементов только этого раздела, а также получить множество вопросов, на описывающие все возможные ошибочные конструкции из элементов только этого раздела. Таким образом, вопросы этих двух множеств позволяют построить разрешающую процедуру для выявления принадлежности произвольного раздела указанному типу. Прагматически вопросы первого и второго множества описывают требования полноты (база знаний может не являться замкнутой) и безошибочности раздела. Покрывающим подмножеством исходного множества вопросов будем называть такое множество вопросов, что из наличия (отсутствия) положительных ответов на каждый вопрос из него следует наличие (отсутствие) ответа на вопрос из исходного, покрываемого множества. В спецификацию базы знаний входят по возможности минимальные покрывающие подмножества из множеств таких вопросов для каждого типа разделов этой базы знаний. Спецификация базы знаний может включать или включает: указание класса (типа) базы знаний, описание её количественных и качественных характеристик, сертификат, состав, задачно-ориентированный сборник тестовых вопросов, информацию о разработчиках, условиях распространения и информацию для сопровождения.

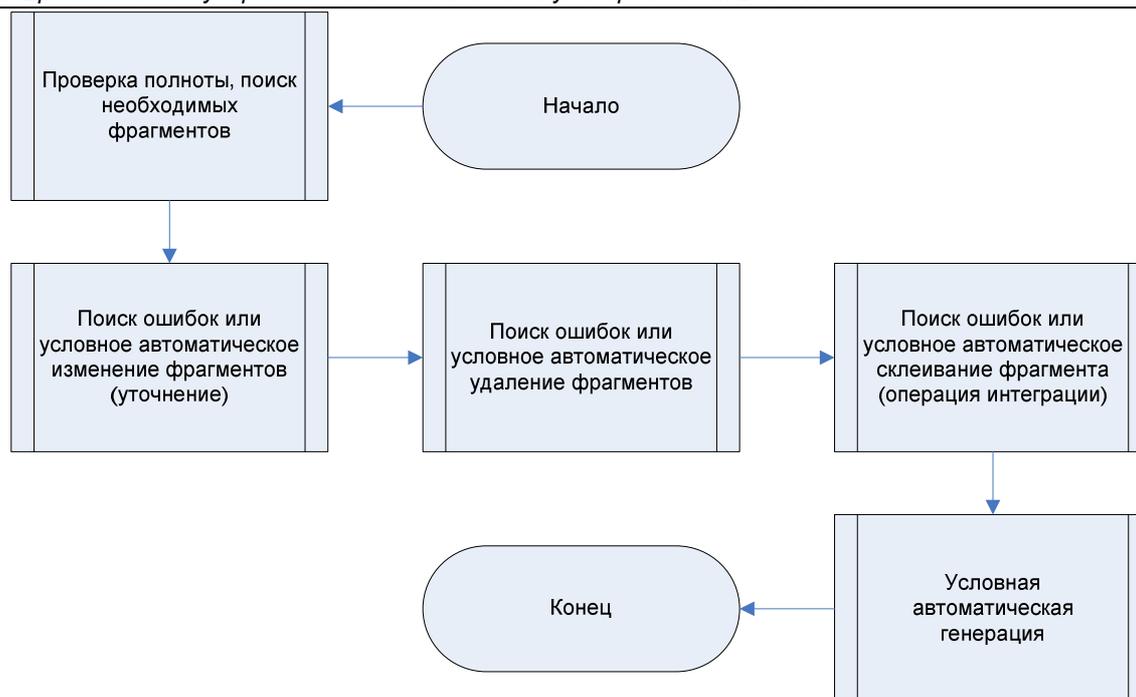


Рис. 1. Схема алгоритма верификации. Основные этапы

Средства автоматизации проектирования баз знаний. Основными задачами, на решение которых ориентированы (инструментальные) средства автоматизации проектирования баз знаний [3], являются задачи редактирования текстов баз знаний, верификации и отладки фрагментов баз знаний и интеграции фрагментов баз знаний. Рассмотрим задачи верификации и интеграции.

Верификация и отладка базы знаний является итерационным процессом. Верификации подлежат как отдельные вновь разработанные фрагменты баз знаний, так и базы знаний и их фрагменты, полученные в результате интеграции. Актуальность верификации и отладки обосновывается наличием человеческого фактора и необходимостью получения корректных, полных, надёжных и качественных результатов.

Проблема верификации и отладки заключается в двух взаимосвязанных аспектах: сокращение времени верификации и минимизация затрат разработчика на чтение сообщений об ошибках и исправление базы знаний. Из существующих подходов можно заключить, что необходимой частью решения этой проблемы является наличие языка правил верификации, а также наличие операций и методики верификации и отладки. Аспект оптимизации процедур верификации с целью сокращения времени их работы рассмотрен в работах [4,5]. Рассмотрим аспект минимизации затрат разработчика. Основная идея сокращения затрат разработчика на исправление базы знаний – использование автоматических механизмов редактирования и исправления базы знаний. Так как база знаний предполагается открытой, то в технологии различаются требования достаточной полноты и требования корректности (непротиворечивости) баз знаний. Особенностью является отсутствие соответствующих разработанных механизмов обработки требований для SC-кода.

Методика верификации и отладки предполагает использование разработчиком встроенных команд верификации и вопросов из сборника тестовых вопросов. При необходимости добавления нового правила верификации или правила автоматического исправления, разработчик может сформулировать это правило на sc-языке и указать соответствующую встроенную команду, при вызове которой это правило будет применяться. Фактически правила верификации определяются такими встроенными командами и вопросами. С целью сокращения возможной повторной верификации правила упорядочиваются по различным типам. Очевидно, что проверка фрагмента базы знаний на полноту может оказаться бессмысленной после его проверки на некорректность (противоречивость) и последующего

автоматического исправления (удаления) некорректных фрагментов, в этом случае разработчик может получить неадекватное сообщение о неполноте базы знаний вместо сообщения об ошибке, связанной с наличием некорректного фрагмента. В связи с этим необходимо выработать некоторую приемлемую последовательность выполнения задач верификации для различных типов правил. Поэтому в соответствии с выделенными типами правил задача верификации разбивается на этапы. Последовательность основных этапов процесса верификации и отладки отражена на рис. 1.

Как следует из рисунка – выделено пять этапов, которые можно распределить по трём фазам: 1) проверка на полноту; 2) поиск и исправление ошибок; 3) автоматическая генерация. На первом этапе с помощью множества вопросов, на которые требуется получить утвердительный ответ, проверяется полнота разделов базы знаний. На втором этапе, как и на двух следующих, с помощью вопросов, описывающих все возможные ошибочные конструкции, выявляются ошибки (в данном случае связанные с неполным или неверным указанием семантического типа элементов). Некоторые ошибки могут быть скорректированы путем автоматического изменения семантического типа элементов. На третьем этапе выявляются ошибки, связанные с некорректными фрагментами информационных конструкций. Некоторые из фрагментов могут быть автоматически удалены. На четвертом этапе выявляются ошибки, связанные с наличием избыточных элементов или фрагментов конструкций. Некоторые фрагменты могут быть автоматически исправлены – интегрированы. На пятом этапе автоматически могут быть добавлены конструкции, повышающие качество или полноту базы знаний. На рис. 2 приведён пример результата верификации в виде сообщения разработчику об обнаруженной ошибке.

Основной проблемой при сборке баз знаний из компонентов является проблема интеграции [6] фрагментов баз знаний (рис. 3), она включает решение таких задач, как выявление избыточности и синонимии, оптимизация сигнатуры (набора понятий) базы знаний, другие задачи повышения качества базы знаний. Принципами и подходами для решения этих проблем являются: выявление потенциальных синонимов в базе знаний, анализ логической схемы (иерархии) понятий базы знаний, использование экспертных знаний для повышения качества баз знаний. Нерешённость проблемы качественной интеграции снижает качество базы знаний, приводит к получению неоднозначных ответов либо не приводит вообще. Следовательно её решение представляется актуальным.

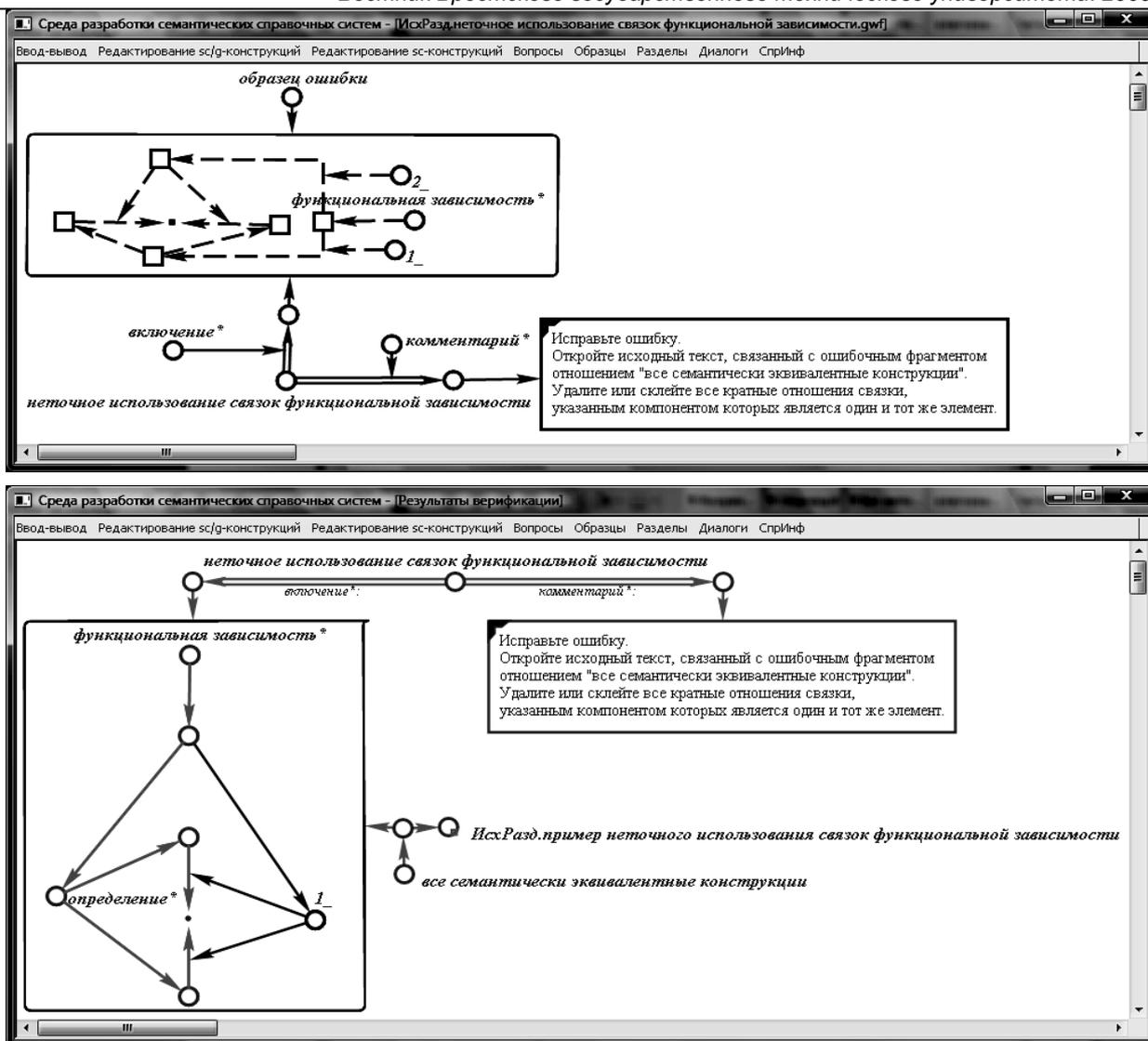


Рис. 2. Примеры описания ошибочной ситуации и диалога системы с разработчиком базы знаний на этапе отладки базы знаний

При интеграции двух баз знаний между ними выявляется некоторое множество пар потенциально синонимичных элементов. Методика точного решения этой задачи включает как решение этой задачи вручную, путём получения соответствующего ответа от разработчика, так и автоматизированное решение на основе известной базовой теоретико-множественной семантики элементов и ключевых узлов sc-языка. Исходя только из базовой теоретико-множественной интерпретации, уже в некоторых фрагментах базы знаний можно легко установить факт наличия или отсутствия синонимии. Зная множество таких пар и множество всех возможных пар всех элементов в заданном фрагменте базы знаний, легко вычислить множество и число пар потенциально синонимичных элементов такой базы знаний. Приведённый выше обобщённый алгоритм поиска множества пар элементов, не являющихся синонимами, представленный на рис. 4, позволяет автоматически установить факт отсутствия синонимии, что снижает нагрузку на разработчика. В ином случае следует адресовать инженеру знаний, являющемуся разработчиком базы знаний, задание на подтверждение или отрицание синонимии для указанных пар потенциально синонимичных элементов. С целью того, чтобы снизить нагрузку на разработчика, необходимо минимизировать число таких заданий. Это может быть достигнуто за счёт того, что ответ, даваемый разработчиком, в силу семантических связей между элементами в базе знаний приводит к тому, что неактуальными становятся не одна пара потенциально синонимичных элементов, а большее количество, то есть число пар потенциально синонимичных элементов сокращается максимальным

образом. За счёт этого можно решать задачу о минимизации. В зависимости от ответа разработчика (реакции внешней среды) это может приводить к тому или иному сокращению числа пар потенциально синонимичных элементов.

Имея информацию о потенциально синонимичных элементах, можно оценить множество и количество всевозможных структурно различных вариантов склеивания потенциально синонимичных элементов в элементы результирующей базы знаний, являющееся подмножеством результатов соответствия интеграции. Это множество обозначим величиной $I(G)$, где G – симметричный ориентированный граф, множество рёбер $E(G)$ которого является множеством всех пар потенциально синонимичных элементов исходных баз знаний, а множество вершин $V(G)$ – множеством всех элементов исходных баз знаний.

$$E(G) \subseteq (V(G))^2.$$

Для того чтобы определить множество $I(G)$, введена специальная операция

$$A \overset{\text{def}}{\cup} B = \bigcup_{(P,Q) \in A \times B} \{P \cup Q\},$$

которая является ассоциативной и коммутативной, т.е. имеет следующие свойства:

$$A \overset{\text{def}}{\cup} B = B \overset{\text{def}}{\cup} A,$$

$$(A \overset{\text{def}}{\cup} B) \overset{\text{def}}{\cup} C = A \overset{\text{def}}{\cup} (B \overset{\text{def}}{\cup} C).$$

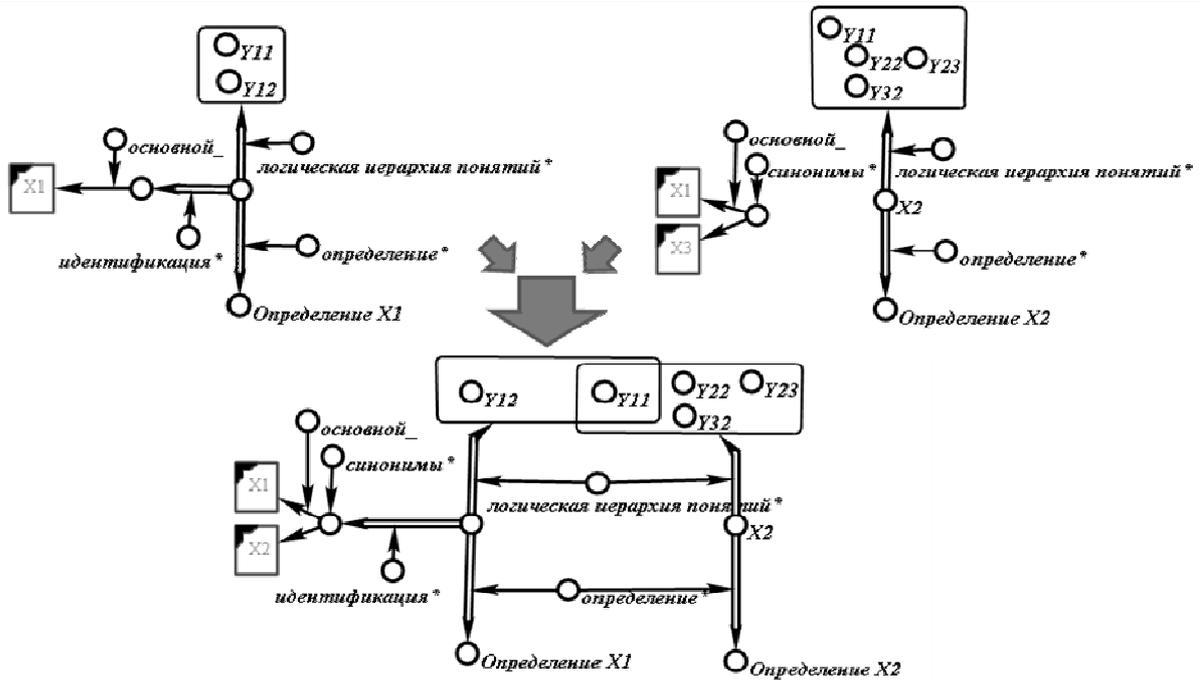


Рис. 3. Иллюстрация интеграции фрагментов баз знаний



Рис. 4. Схема алгоритма выявления пар элементов, не являющихся синонимами

Чтобы задать множество $I(G)$, введём семейство всевозможных множеств рёбер ориентированных графов $C(G)$ на множестве вершин графа G , каждая компонента связности которых является полным подграфом.

$$C(G) = \left\{ \bigcup_{z \in X} Z^2 / \{(x, x) | x \in V(G)\} \mid X = 2^{V(G)} \cap \left\{ S \mid S \cap \bigcup_{y \in X \setminus \{S\}} Y = \emptyset \right\} \right\},$$

тогда

$$I(G) = \left(\bigcup_{e_{ij} \in E(G)} \{ \emptyset, \{e_{ij}\} \} \right) \cap C(G).$$

Число Q элементов множества $I(G)$ можно рассматривать в виде критерия качества: чем меньше это число, тем выше качество и наоборот.

$$Q = |I(G)|.$$

Для числа Q справедливы следующие соотношения

$$1 + |E(G)|/2 \leq Q \leq Bell(\{|V(G)\} \cup \{|E(G)|/2\}),$$

где $Bell$ – функция, возвращающая n -ое число из последовательности чисел Белла.

Минимальное возможное число элементов результирующей базы знаний, на которые отображены элементы исходных баз знаний, можно выразить:

$$V_{\min} = |V(G)| - \max \bigcup_{R \in I(G)} \left\{ \sum_{S \in Z} (|S| - 1) \mid Z = \{ \{i\} \cup Y \mid Y = \{x \mid (x, i) \in R\} \} \right\}.$$

Чтобы получить наиболее качественную базу знаний, необходимо выяснить, являются ли потенциально синонимичные элементы синонимами или не являются. Если вероятность получить отрицательный ответ о синонимии элементов высока, то тогда с целью повышения качества результирующей базы знаний необходимо выбирать для задания на подтверждение или отрицание синонимии такую пару элементов $e_{ij} = (i, j)$, для которой верно следующее:

$$\begin{aligned} \dot{e}_{ij} &\in \{x \mid \{R \mid in(x) \cap R \neq \emptyset\} \cap I(G)\} = \\ &= \max_y \left\{ \{R \mid in(y) \cap R \neq \emptyset\} \cap I(G) \right\}, \end{aligned}$$

где in – функция, возвращающая для указанного элемента, множество элементов семантической окрестности по входящим дугам.

Если же высока вероятность получить положительный ответ о синонимии элементов, то тогда с целью повышения качества результирующей базы знаний рекомендуется выбирать такие элементы для задания, для которых верно:

$$\dot{e}_{ij} \in \left\{ x \mid \sum_{R \in \{S \mid in(x) \cap S \neq \emptyset\} \cap I(G)} |R| = \max_y \left\{ \sum_{R \in \{S \mid in(y) \cap S \neq \emptyset\} \cap I(G)} |R| \right\} \right\}.$$

В последнем случае нет гарантии, что выбранная таким способом пара элементов при положительном ответе приведёт к минимальному числу формулируемых заданий. Однако такое требование предлагается в качестве некоторого приближения к оптимальному решению.

Следует отметить что, несмотря на возможность выполнения этих требований, не для всех пар элементов можно понятно сформулировать требуемое задание на подтверждение или отрицание синонимии. Соответственно, тогда должны выбираться пары таких

элементов, для которых имеются термины (внешние обозначения), относящиеся к языку разработчика, с которым ведётся диалог.

Заключение. Особенности и достоинства предложенной технологии проектирования баз знаний являются: унифицированное представление баз знаний и их фрагментов; перенос акцента от традиционной инженерии баз знаний к интеграции баз знаний, т.е. к сборке баз знаний из крупных модулей, основывающейся на унифицированной модели баз знаний; обеспечение совместимости разрабатываемых баз знаний; использование визуальных методов проектирования; инструментальные средства проектирования баз знаний рассматриваются как система, имеющая пользовательский интерфейс, машину обработки знаний и базу знаний, разработанную в соответствии с данной технологией. База знаний инструментальных средств проектирования баз знаний хранит программы операций редактирования, верификации и отладки, интеграции, а также образцы навигационных вопросов и конструкции гипермедийного языка для хранения исходных текстов баз знаний.

СПИСОК ЦИТИРОВАННЫХ ИСТОЧНИКОВ

1. Базы знаний интеллектуальных систем / Т.А. Гаврилова, В.Ф. Хорошевский – СПб: Питер, 2000.
2. Представление и обработка знаний в графодинамических ассоциативных машинах / В.В. Голенков [и др.]; под ред. В.В. Голенкова. – Минск: БГУИР, 2001.
3. Инструментальные средства разработки баз знаний. / В.П. Ивашенко // Дистанционное обучение образовательная среда XXI века: Материалы VI Междунар. науч.-метод. конференции, 22–23 нояб. – Минск. – 2007. – С. 182–185.
4. Ivashenko, V. Applications of optimization methods for search of semantic network constructions. / Ivashenko V. // Abstracts of the Conference of the European Chapter on Combinatorial Optimization ECCO XVIII, Minsk, Belarus, May 26–28, 2005. – Minsk. – 2005. – P. 23–24.
5. Язык описания синтаксических правил для однородных семантических сетей. / В.П. Ивашенко // Дистанционное обучение образовательная среда XXI века: Материалы VI Междунар. науч.-метод. конференции, 22–23 нояб. – Минск. – 2007. – С. 185–188.
6. Интеграция знаний в информационных системах. / Н.А. Гулякина, В.П. Ивашенко // Доклады БГУИР. – 2004. – №6. – С. 113–119.

Материал поступил в редакцию 22.10.09

IVASHENKO V.P. Algorithms of verification and integration of bases of knowledge

The description making semantic technology of designing of bases of knowledge is given: models of bases of knowledge constructed on the basis of semantic networks, means of automation of designing of bases of knowledge, including algorithms of verification and integration of fragments of bases of knowledge. For the decision of a task of integration the methods of the theory of sets are used. The application of results of algorithm for reduction of pairs potential synonyms in base of knowledge is considered.

УДК 530.1

Волков Е.Г., Дереченник С.С.

ОСОБЕННОСТИ РАСЧЕТА ФРАКТАЛЬНОЙ РАЗМЕРНОСТИ КЛАСТЕР-КЛАСТЕРНЫХ АГРЕГАТОВ

Введение. Многие реальные физические объекты представляют собой кластерные структуры. Например, коллоидные кластеры образуются в растворах в результате химических реакций, твердотельные кластеры формируются в результате различных превращений в твердой фазе: в ходе твердотельных (фото)химических реакций, в результате спекания, в процессе перехода аморфной фазы в кристаллическую и др. [1]. Формирование таких структур происходит по механизму кластер-кластерной агрегации (ККА), который может моделироваться описанным ниже алгоритмом. В ограниченном модельном простран-

стве случайным образом размещается некоторое число частиц, которые начинают блуждание в пространстве в соответствии с броуновским движением. При столкновении частицы слипаются и двигаются вместе, образуя первичный кластер. Процесс заканчивается, когда все первичные кластеры объединяются в конечный кластер.

Полученный кластер-кластерный агрегат, как правило, является фракталом. У всех фрактальных кластеров, образующихся в природе, наблюдается непосредственная связь между их структурой и физико-химическими свойствами. Например, в [2] был проведен

Волков Евгений Геннадьевич, аспирант, ассистент кафедры электронных вычислительных машин и систем Брестского государственного технического университета.

Дереченник Станислав Станиславович, к.т.н., доцент, зав. кафедрой электронных вычислительных машин и систем Брестского государственного технического университета.

Беларусь, БрГТУ, 224017, г. Брест, ул. Московская, 267.

Физика, математика, информатика