



УДК 004.822:514

СЕМАНТИЧЕСКИЕ МОДЕЛИ И СРЕДСТВА ИНТЕГРАЦИИ И ОТЛАДКИ БАЗ ЗНАНИЙ

Ивашенко В.П.

*Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь*

ivashenko@bsuir.by

Рассматриваются составляющие и применение средств технологии компонентного проектирования баз знаний в виде однородных семантических сети с теоретико-множественной семантикой для решения задач отладки и интеграции баз знаний.

Ключевые слова: база знаний, онтология, семантическая сеть, интеграция.

ВВЕДЕНИЕ

Рассматриваются задачи отладки и интеграции баз знаний в рамках задачи разработки базы знаний для интеллектуальной системы.

База знаний рассматривается как связанная информационная конструкция, с заданной структурой [Ивашенко, 2009а], состоящая из знаков и связей отношений инцидентности между ними.

Задача отладки базы знаний заключается в построении базы знаний, удовлетворяющей некоторому конечному набору требований, на каждый которых может быть получен ответ «да», либо «нет». База знаний строится из исходного конечного множества фрагментов баз знаний. Требования формулируются на формальном логическом языке описания ошибок и описывают целостность, противоречия, неточности и полноту отлаживаемой базы знаний.

Задача интеграции баз знаний рассматривается как подзадача отладки и заключается в построении базы знаний из конечного множества исходных баз знаний и дополнительной метаинформационной конструкции.

Разработка баз знаний наталкивается на следующие трудности:

- отсутствие развитых технологий разработки или их ограничения:
 - ограниченные возможности верификации и отладки баз знаний (отсутствие средств верификации количества синонимичных знаков, неполное выявление противоречий);
 - плохая отчуждаемость и переносимость базы знаний в силу привязки к инструментарию или

конкретной оболочке (CLIPS) [CLIPS, 1991], ограниченность языков и моделей представления знаний

- немногочисленность инженеров баз знаний (из-за высоких стартовых требований к разработчику) – от разработчика требуется владение специальными знаниями по моделям и языкам представления знаний):
 - необходимость выбора среди нескольких моделей знаний, между которыми нет однозначного предпочтения [Martin Ph., 2002];
 - неоднородность моделей представления знаний, приводящая к тому, что в базе появляются синонимичные структуры разных типов, требующие от пользователя их согласования;
 - сложность языков представления знаний
- не полностью решён вопрос интеграции баз знаний:
 - ограничения на расширение базы знаний;
 - ограниченные иерархической таксономической структурой онтологии возможности интеграции баз знаний или отсутствие таких возможностей, необходимость выбора средств интеграции пользователем [Gangemi et al., 1996];
 - отсутствие общих стандартов совместимости разработанных фрагментов баз знаний;
 - ограниченность средств поиска и каталогизации разработанных фрагментов баз знаний;
 - ограниченность или отсутствие простых средств взаимодействия с внешней средой.

Для разработки баз знаний используются:

- языки представления и обработки баз знаний;
- средства создания и отладки баз знаний;
- средства интеграции баз знаний.

К языкам представления знаний в разных системах на сегодняшний день относятся: Conceptual Graph [Sowa et al., 2008], Frame-logic [Michael Kifer et al., 1995], Knowledge Interchange Format [Genesereth et al., 1992], Integrated DEFinition for Ontology Description Capture Method (IDEF5) [IDEF5, 1994], Common Algebraic Specification Language [CoFI:CASL-Summary, 2004], Concept maps/UML [Novak et al., 2008] [ISO24707], RDF/RDFS [W3C:RDFS, 2004], DARPA Agent Markup Language [DAML, 2006], CycL [CycL, 2002], Ontology Inference Layer [W3C:DAML+OIL, 2001], RDF/OWL-Lite, RDF/OWL-DL, RDF/OWL Full [W3C:OWL, 2004], Topic Maps [ISO13250], XTM/LTM/CML/GML [XTM, 2001], Common Logic [ISO24707], Developing Ontology-Grounded Methods and Applications [Mustafa Jarrar et al., 2008], Formal English [Martin Ph., 2002], Gellish [Van Renssen, 2005], RDF/Rule Interchange Format [W3C:RIF, 2010], Open Biomedical Ontologies [Smith et al., 2007], RDF/OWL2 [W3C:OWL2, 2009], RDF/OWL2 EL, RDF/OWL2 RL и др.

Эти языки могут быть классифицированы по различным признакам:

- синтаксические признаки
 - URI-ориентированные
 - линейные языки
 - мультиплетные языки
 - триплетные языки
 - квинтиплетные языки
 - иерархически структурированные языки
 - ЛИСП-подобные языки
 - нелинейные языки
 - графовые языки
 - гиперграфовые языки
- семантические признаки
 - фреймовые языки
 - правила
 - семантическая сеть
 - семантический гиперграф
 - логические языки
 - дескрипционной логики
 - логики предикатов первого порядка
 - логики предикатов высших порядков
 - модальной логики
 - темпоральной логики (линейной, ...)
 - взаимодействующих процессов Хоара
 - объектно-ориентированные языки
 - процедурные языки
 - естественный язык

Кроме очевидных, дополнительно к недостаткам [Martin Ph., 2002] вышеперечисленных языков представления знаний можно отнести:

- отсутствие разделения понятий и терминов (исключения – DOGMA и Gellish),
- отсутствие поддержки монотонного расширения базы знаний (исключения – OWL, OWL2 и языки, построенные на основе классических логических моделей),
- отсутствие у некоторых из вышеперечисленных языков возможности семантического расширения языка (замкнутость языка).

Для поиска в базах знаний и онтологиях [Хорошевский, 2008] используются такие языки, как: RDQL, squish, SPARQL [W3C:SPARQL, 2008], KQML, DMX, Datalog, TSQL, ERROL, RuleML, RQL, OQL, TQL, VERSA, DQL и др.

Среди средств, которые могут рассматриваться в качестве основы для разработки баз знаний, можно выделить: оболочки экспертных систем (CLIPS (FuzzyCLIPS, DYNACLIPS, WxCLIPS) [CLIPS, 1991], SOAR, OPS83, RT-EXPERT, MIKE, BABYLON, WindExS, ES; ACQUARE, Easy Reasoner, ECLIPSE, EXSYS Professional, SIMER+MIR, AT ТЕХНОЛОГИЯ, CAKE v2.0) [Гаврилова и др., 2000]; инструментальные пакеты для разработки экспертных систем (G2, ART, KEE, Knowledge KRAFT); системы, ориентированные на обработку онтологий [Sowa et al., 2008] – Protégé, WebOnto, OntoEdit, WebODE, OilEd, OntoLingua.

Достоинствами приведённых 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.

Для преодоления трудностей семантической интеграции [Doan and Halevy, 2005], [Кудрявцев, 2008] (отображения онтологий (ontology mapping) и интеграции знаний (knowledge integration) в базах знаний используются следующие подходы:

- сравнение и выравнивание онтологий (ontology matching & alignment),
- интеграция онтологий (ontology merging),
- семантическое сравнение (semantic matching),
- семантическая унификация (semantic unification).

Для интеграции онтологий можно выделить методы:

- структурно-синтаксические:
 - анализ внутренней структуры
 - экстенционально-статистические
 - анализ внешней структуры (метаструктурный анализ)
 - терминологические (лексические)
- логико-семантические

К существующим на настоящий момент средствам интеграции онтологий можно отнести: Optima, Prompt, Ontolingua, Chimaera [McGuinness et al., 2000], ONION [Gangemi et al., 1996], COMA++ [Aumuell et al., 2005] и др.

Однако все эти средства и методы имеют ограничения и не преодолевают в полной мере вышеперечисленные трудности.

1. Семантическая модель интеграции баз знаний и фрагментов баз знаний

Семантическая модель интеграции баз знаний и фрагментов баз знаний использует унифицированное представление баз знаний и обеспечивает интеграцию sc-моделей баз знаний и их фрагментов. Унифицированное представление баз знаний обеспечивается унифицированной моделью баз знаний и унифицированной моделью представления знаний, которая является частным случаем такой модели представления знаний, как семантические сети и задаётся семейством совместимых sc-языков, использующих унифицированный способ семантического кодирования Semantic Computer code (SC-код) [Голенков и др, 2001] и поддерживающих представление знаний различного вида [Ивашенко, 2003], [Ивашенко, 2004], [Ивашенко, 2011a]. Особенности SC-кода являются: простой алфавит, содержащий узлы и дуги, простой синтаксис, базовая теоретико-множественная интерпретация. Для описания статических предметных областей sc-языки используют математический аппарат теории множеств, для строгого описания динамических предметных областей, используются понятия нестационарной принадлежности и непринадлежности, которые задают понятия нестационарного множества. В SC-коде различаются понятия обозначения и понятия множества. Для обозначений и множеств рассмотрим функции T_{sign} и T_{sets} , которые сопоставляют каждому обозначению и множеству множество моментов времени существования этого обозначения или множества. Таким образом, знак связи отношения принадлежности (непринадлежности), дуга arc , является обозначением связи нестационарной принадлежности (непринадлежности) тогда и только тогда, когда:

$$(T_{set}(begin(arc)) \cup T_{sign}(end(arc))) / T_{set}(arc) \neq \emptyset \quad (1)$$

Здесь $begin(arc)$ обозначает начало дуги arc , а $end(arc)$ – конец дуги arc . В ином случае дуга arc является обозначением стационарной принадлежности (непринадлежности).

В отличие от вышеперечисленных языков Semantic Code ориентирован на универсальность, т.е. на представление любых видов знаний. Semantic Code является средством унификации представляемых знаний. На множестве sc-языков (языков, представленных в SC-коде) определены

отношения sc-подязыка и трансляции (кодировки). Подязык, являющийся пересечением выделенного семейства совместимых специализированных sc-языков, рассматривается как интегрированный sc-язык представления знаний (SCK). 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: каждому sc-языку однозначно сопоставляется конечное множество ключевых узлов (элементов) этого языка. Каждый ключевой узел задаёт ограничения на собственную семантическую окрестность в информационных конструкциях (текстах) этого языка. Множество ключевых элементов интегрированного sc-языка представления знаний является объединением множеств ключевых элементов остальных sc-языков выделенного семейства. Характеристиками sc-языка являются: мощность множества ключевых узлов sc-языка; семейство множеств собственных семантических окрестностей ключевых элементов языка; наличие функциональных зависимостей между собственными окрестностями множеств ключевых элементов sc-языка; соотношение алгоритмических сложностей поиска или вычисления элементов собственных семантических окрестностей на основании существующих зависимостей. Построена семантическая онтология существующих языков и моделей представления знаний, в которой указано место унифицированной модели представления знаний и её соотношение с существующими моделями представления знаний.

Введенные в sc-языки ключевые элементы поддерживают соответствующее семантически эквивалентное представление следующих теоретико-множественном концептов OWL 2 QL и OWL 2 EL [W3C:OWL2, 2009]: SubClassOf, EquivalentClasses, DisjointClasses, ClassAssertion, DisjointUnion, SameIndividual, SubDataPropertyOf, EquivalentDataProperties, DataPropertyAssertion, DataComplementOf, DataIntersectionOf, DataUnionOf, DataPropertyDomain, DataPropertyRange, DataSomeValuesFrom, DataMinCardinality, DataMaxCardinality, DataExactCardinality, FunctionalDataProperty, SubObjectPropertyOf, EquivalentObjectProperties, DisjointObjectProperties, ObjectPropertyAssertion, ObjectComplementOf, ObjectIntersectionOf, ObjectUnionOf, ObjectPropertyDomain, ObjectPropertyRange, ObjectMinCardinality, ObjectMaxCardinality, ObjectExactCardinality, FunctionalObjectProperty, InverseFunctionalObjectProperty, ReflexiveObjectProperty. Остальные концепты могут быть определены с помощью логического sc-языка.

Унифицированная модель баз знаний описывает базы знаний специального вида, в которых знания представлены с использованием унифицированной модели представления знаний [Ивашенко, 2009a]. В унифицированной модели баз знаний база знаний – связанная структурированная информационная

конструкция, структура которой состоит не менее чем из одного 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. Между базами знаний определены отношения содержательного и структурного включения одной базы знаний в другую. На множестве баз знаний заданы операции содержательной проекции и темпорализации [Ивашенко, 2009a]. Соответствие интеграции каждой паре баз знаний сопоставляет однозначное отображение множества всех знаков, формирующих эти базы знаний, на множество всех знаков результирующей базы знаний. Путём интеграции осуществляется переход от некоторой исходной базы знаний к требуемой оптимизированной базе знаний, которая имеет более высокое качество [Ивашенко, 2011a].

Решение задачи интеграции требуется в основных трёх случаях:

- интеграция разработанных и отлаженных баз знаний;
- интеграция отлаженных компонентов в базу знаний;
- добавление знаний при редактировании базы знаний.

В задаче интеграции двух баз знаний или их фрагментов в качестве исходных данных используются две БЗ (или фрагмента) и дополнительная метаинформация о свойствах знаков, принадлежащих этим базам знаний. Результатом решения этой задачи является нахождение интегрированной базы знаний, такой, что каждый знак из исходных интегрируемых баз знаний, имеет единственное представление в виде соответствующего знака в интегрированной базе [Ивашенко, 2009a]. Будем говорить, что осуществляется слияние двух знаков исходных БЗ тогда и только тогда, когда каждому из обоих соответствует в интегрированной базе знаний единственной знак. При интеграции в качестве дополнительной метаинформации может использоваться информация о внешних обозначениях (идентификаторах) понятий или информация, заданная базовой или ключевой рефлексивной семантикой знаков из этих БЗ. В условиях неполноты информации можно выделить два типа стратегий слияния знаков интегрируемых баз знаний: безопасные (выполняемые однозначно и непротиворечиво) и небезопасные. Безопасная стратегия включается в любую стратегию слияния.

Задача интеграции баз знаний может быть разделена на две задачи – интеграция содержания баз знаний и интеграция структуры баз знаний (обеспечение целостности базы знаний).

Без потери общности для решения задачи интеграции содержания баз знаний рассмотрим задачу интеграции двух фрагментов баз знаний, каждый из которых представляет связную информационную конструкцию sc-языка.

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.

Имея информацию о потенциально синонимичных элементах, можно оценить множество и количество всевозможных структурно различных вариантов слияния 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

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

Для того, чтобы определить множество $I(G)$, используем специальную операцию

$$A \dot{\cup} B = \bigcup_{(P,Q) \in A \times B} \{P \cup Q\}, \quad (3)$$

которая является ассоциативной и коммутативной.

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 [Ивашенко, 2009b], тогда

$$I(G) = \left(\bigcup_{e_{ij} \in E(G)} \{\emptyset, \{e_{ij}\}\} \right) \cap C(G). \quad (4)$$

Число Q элементов множества $I(G)$ можно рассматривать в виде критерия качества: чем меньше это число, тем выше качество и наоборот. Однако, уже для небольших фрагментов баз знаний, это число может оказаться большим и трудным для расчёта, поэтому на практике более целесообразно использовать или логарифм этого числа или логарифм его оценок, или связанные с ними величины (число пар потенциальных синонимов).

$$Q = |I(G)|. \quad (5)$$

В работе [Ивашенко, 2009b] для числа Q установлены соотношения для верхней и нижней границы.

Интеграция онтологий и баз знаний осуществляется через слияние знаков. Рассмотрим m объектов области значений семантической интерпретации некоторого знака. Тогда, на множестве этих объектов, число неоднозначных и однозначных постоянных семантических интерпретаций (семантик) знака равно $2^{2^m}-1$.

Чтобы осуществить слияние знаков (воплощений знака) необходимо, чтобы пересечение областей значения их семантик было непустым. В частности – теоретико-множественных семантик.

Если пересечение областей значения семантик знаков является пустым, то слияние таких знаков невозможно и такие знаки различны.

Тогда как для выявления различных знаков достаточно одного контр-примера, для выявления подлежащих слиянию знаков необходимо не только установить, что пересечение областей значения их семантик не является пустым, но и доказать, что эти знаки совпадают. Для этой цели используются различные утверждения о совпадении в виде аксиом и теорем.

Опишем свойства этого отношения совпадения. Если два знака совпадают, то не существует множества, которому одновременно принадлежит один из них и не принадлежит другой. Если два знака множеств совпадают, то множества равны. Выявить неравенство множеств просто: достаточно найти элемент, который принадлежит одному множеству и не принадлежит другому. Процедура выявления равенства множеств зависит от их свойств. Если множества имеют конечное, не очень большое число принадлежностей или не принадлежностей элементов, что характерно для понятий, то перечислив все принадлежности или все не принадлежности, число которых совпадает соответственно с его мощностью или количеством не принадлежностей, и убедившись, что среди них нет не принадлежностей элементов, принадлежащих другому множеству, можно заключить, что множества равны. Если же множества бесконечны или число принадлежностей, равно как и не принадлежностей у них, очень большое, то равенство множеств можно доказать только через утверждения об их свойствах.

Прежде, чем проводить слияние знаков двух баз знаний, следует провести отображение баз знаний (по аналогии с отображением онтологий).

Для отображения в некоторых работах [Maltese et al., 2010] используются следующие отношения между онтологическими понятиями $\{\equiv, \supseteq, \subseteq, \perp, \sqcap\}$, соответственно – совпадения, обобщения, частности, строгого исключения и строгого пересечения. Например, эти отношения в проекции на теоретико-множественные соответствуют отношениям равенства множеств ($=$), надмножества (\supseteq), подмножества (\subseteq), пустого пересечения множеств ($S1 \cap S2 = \emptyset$) и непустого пересечения множеств ($S1 \cap S2 \neq \emptyset$). Однако, как можно показать,

эти отношения трудно или невозможно установить, когда множества имеют большую или бесконечную мощность, или если исходить из того, что существуют равные, но несовпадающие множества.



Рисунок 1– Пример, приводящий к противоречивому отображению

Для приведённого примера (рис. 1), при условии известности, что понятия ромба и квадрата не пересекаются, будет получено противоречивое «сильное» отображение (многоугольник (S1), многоугольник \wedge четырёхугольник (S2), многоугольник \wedge четырёхугольник \wedge ромб (S3), многоугольник \wedge четырёхугольник \wedge квадрат (S4), замкнутая ломаная (S5), замкнутая ломаная \wedge четырёхугольник (S6), замкнутая ломаная \wedge четырёхугольник \wedge прямоугольник (S7), замкнутая ломаная \wedge четырёхугольник \wedge параллелограмм (S8), замкнутая ломаная \wedge четырёхугольник \wedge параллелограмм \wedge ромб (S9)).

Таблица 1 – Противоречивое отображение

| | S1 | S2 | S3 | S4 |
|----|-------------|-------------|-------------|-------------|
| S5 | \equiv | \supseteq | \supseteq | \supseteq |
| S6 | \subseteq | \equiv | \supseteq | \supseteq |
| S7 | \subseteq | \subseteq | \equiv | \equiv |
| S8 | \subseteq | \subseteq | \equiv | \supseteq |
| S9 | \subseteq | \subseteq | \equiv | \perp |

Кроме этого предложенные [Maltese et al., 2010] методы отображения применимы к так называемым «легко-взвешенным» онтологиям [Giunchiglia et al., 2006] и не применимы к другим, аналогичные трудности встречаются и в других подходах [Aumuellер et al., 2005], [Jean-Mary et al., 2007], [Nagy et al., 2010].

Исходя из этого, автором предложен следующий набор отношений {не уточнены*, связность*, различие*, исключение* (не обобщает, не конкретизирует), возможное включение* (потенциально конкретизирует, потенциально обобщает), симметричное исключение*, строгое пересечение*, совпадение*} отношение потенциальной синонимии можно выразить, как возможная синонимия* \equiv (не уточнены* \cup связность*)/(различие* \cup исключение* \cup возможное включение*)).

Следующее правило (рис. 2) позволяет в рамках унифицированной модели представления знаний единообразно, непротиворечиво и строго, что было одной из трудностей в [Maltese et al., 2010], задать правила отображения и слияния знаков баз знаний

при использовании методов лексико-терминологического анализа (анализа идентификаторов).

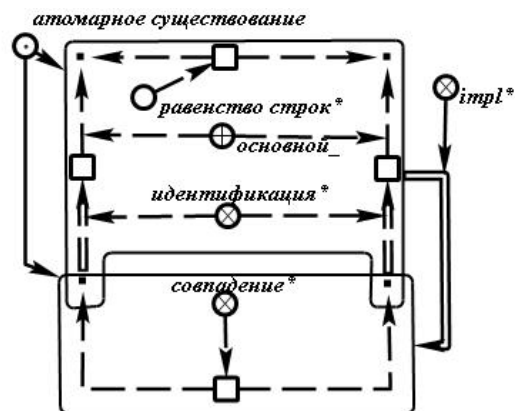


Рисунок 2 – Пример, приводящий к противоречивому отображению

Стратегии слияния при использовании методов лексико-терминологического анализа подробно рассмотрены в [Ивашенко, 2011b].

Следующие правила (рис. 3-8) описывают установление отображающих отношений на множестве знаков интегрируемых фрагментов баз знаний, после проведённых слияний в результате лексико-терминологического анализа.

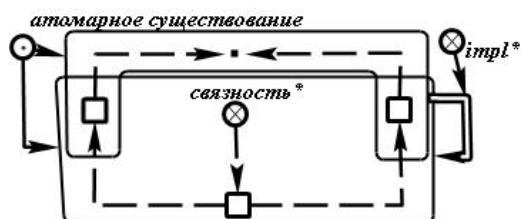


Рисунок 3 – Свойство отношения связности («два множества связны, если имеют общий элемент»)

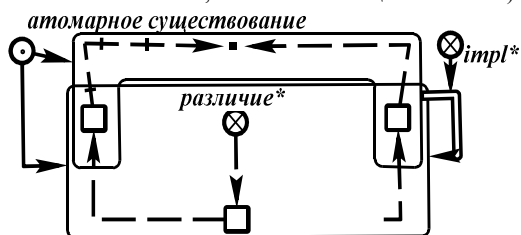


Рисунок 4 – Свойство отношения различия («два элемента различны, если один принадлежит множеству, а другой – нет»)

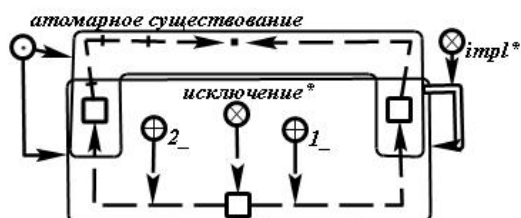


Рисунок 5 – Свойство отношения исключения (не обобщает, не конкретизирует)

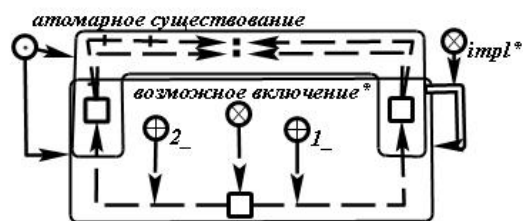


Рисунок 6 – Свойство отношения возможного (потенциального) включения

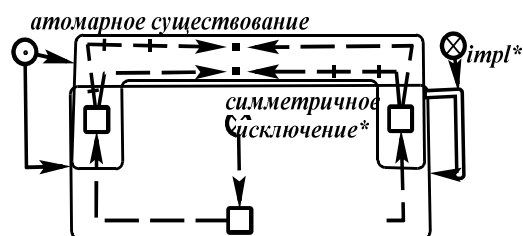


Рисунок 7 – Свойство отношения симметричного исключения

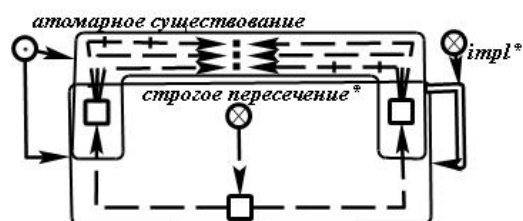


Рисунок 8 – Свойство отношения строгого пересечения

Все эти отношения можно однозначно и непротиворечиво установить в результате соответствующего базовой теоретико-множественной семантике структурного анализа, включая экстенциональный, за полиномиальное время. Более глубокий логико-семантический анализ может потребовать больших временных затрат.

Для анализа на логико-семантическом уровне важно выделять утверждения, которые приводят к совпадению знаков интегрируемых фрагментов, в результате чего можно произвести их слияние. Таковыми высказываниями являются высказывания о свойствах логических высказываний о единственности, высказывания о свойствах множеств без кратных вхождений элементов (канторовских множеств), высказывания об отношениях без кратных связок. Ниже приведён пример (рис. 9), описывающий свойство логических утверждений о единственности и сводящий задачу анализа таких утверждений к задаче выявления связок отношений совпадения. Таким образом, при наличии средств логического вывода для решения задачи интеграции логико-семантическими методами достаточно описать свойства соответствующих ключевых узлов, выражающих

количественные ограничения (например, единственность) или описывающих отсутствие кратных связей, через понятия совпадения sc-элементов (знаков текстов sc-языка).

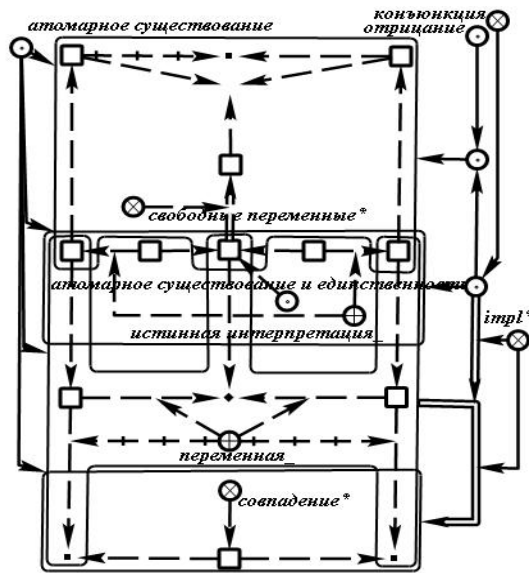


Рисунок 9 – Высказывание о слиянии значений переменных в утверждения единственности

Для интеграции фрагмента базы знаний требуются следующие sc-операции, решающие следующие задачи:

- поиск и выборка элементов заданного интегрируемого фрагмента;
- проверка и выборка элементов, имеющих строковое содержимое и хранящих внешние обозначения (идентификаторы);
- проверка и выборка элементов, имеющих основной идентификатор;
- проверка и выборка элементов, имеющих неосновной идентификатор;
- добавление в словарь элемента с идентификатором;
- поиск и выборка элементов из словаря по идентификатору;
- формирование для заданных идентификаторов факта одинаковых идентификаторов;
- слияние совпадающих элементов;
- выявление и формирование фактов потенциально синонимичных элементов;
- выявление и формирование отображения знаков онтологий;
- формирование исходного варианта слияния элементов заданных разделов;
- применение к заданным элементам утверждений о совпадении элементов, в частности элементов, имеющих одинаковый основной идентификатор, утверждений о различии (несовпадении) и о возможном совпадении элементов;
- выбор и формирование варианта слияния заданных элементов для установленного факта

возможного совпадения и заданного варианта слияния;

- выбор варианта слияния элементов базы знаний, исключение невыбранных вариантов слияния.

Обобщённый алгоритм интеграции фрагментов двух баз знаний приведён в [Ивашенко, 2011b]. Ниже (рис. 10) приведён алгоритм выявления пар потенциальных синонимов, использующий вышеописанные правила экстенционального метода.

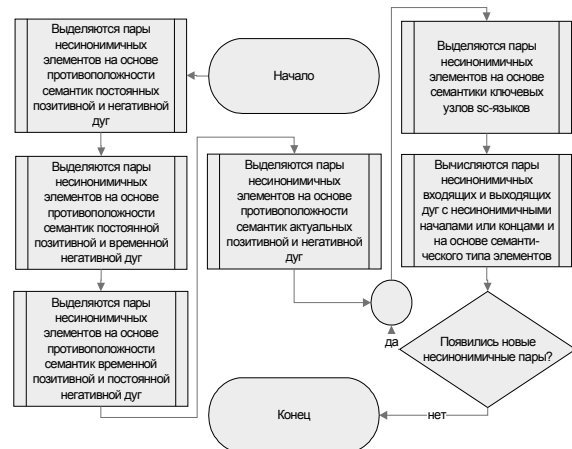


Рисунок 10 – Алгоритм вычисления множества пар различающихся (несинонимичных) элементов

Интеграция структуры баз знаний может быть проведена на исходном уровне или на метауровне. В первом случае, если в базе знаний после содержательной интеграции появились разделы, слияние которых не произошло, то формируется новый раздел базы знаний, который декомпозируется на основные (недекомпозируемые) разделы интегрируемых баз знаний. Этот вариант допустим, если декомпозиция разделов сохраняет древовидную структуру. Второй вариант предполагает формирование мета-разделов, которые описывают основные разделы исходных баз знаний и их структуру и на которые декомпозируется основной раздел интегрированной базы знаний. Второй вариант интеграции структуры баз знаний может иметь смысл, когда содержание исходных баз знаний является слабо связным.

Рассмотрим пример интеграции двух фрагментов баз знаний (см. рис. 11).

Для решения приведённой в примере задачи интеграции требуется следующая дополнительная информация:

- информация о равенстве основных идентификаторов;
- информация о равенстве неосновных идентификаторов;
- информация о том, какие множества являются множествами без кратных вхождений элементов (канторовские множества);
- информация об отсутствии кратных связей в геометрических отношениях;

- утверждение об однозначном задании биссектрисы заданного треугольника, проходящей через заданную вершину;
- утверждение об однозначном задании треугольника заданными тремя вершинами
- утверждение об однозначном задании вписанной окружности заданным треугольником;
- утверждение о совпадении центра вписанной в треугольник окружности и точки пересечения биссектрис треугольника.

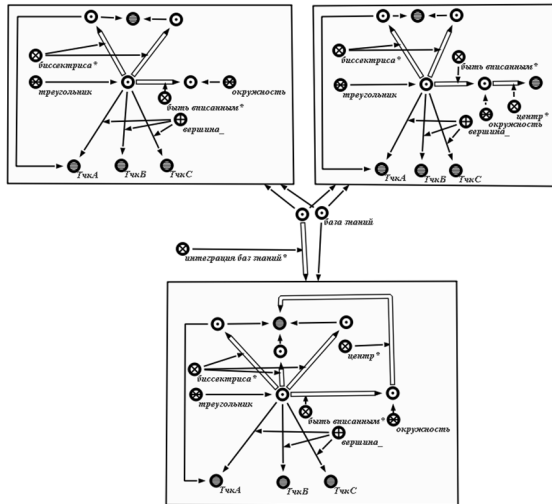


Рисунок 11 – Пример интеграции двух фрагментов баз знаний

2. Компонентное проектирование баз знаний

2.1. Модель компонентов баз знаний

Модель компонентов баз знаний описывает в соответствии с классификацией классы специфицируемых баз знаний и их фрагментов и отношения между ними и их частями. Спецификация базы знаний включает указание класса (типа) базы знаний и описание её количественных и качественных характеристик: состав, сертификат, задачно-ориентированный сборник тестовых вопросов, требования полноты и безошибочности базы знаний, информацию о разработчиках, условиях распространения и информацию для сопровождения.

2.2. Библиотека компонентов баз знаний

Библиотека компонентов баз знаний основывается на модели компонентов баз знаний, которая в свою очередь основана на унифицированной модели баз знаний, и также состоит из базы знаний, машины обработки знаний и пользовательского интерфейса. В зависимости от соотношения значений собственных характеристик [Ивашенко, 2011] и других признаков база знаний может быть классифицирована как:

- терминологический словарь
 - одно-языковой

- многоязыковой
- таксономия
 - конечного числа объектов
 - бесконечного числа объектов
- реляционная модель (онтология)
 - графовая реляционная модель
 - неграфовая реляционная модель
- тезаурус
 - с рекурсивными определениями
 - без рекурсивных определений
- теория
 - теория стационарной предметной области
 - теория нестационарной предметной области
- прикладная база знаний
 - фактографическая
 - нефактографическая
 - задачник
 - библиотека алгоритмов

Эта классификация, как и классификация фрагментов баз знаний, является основой модели компонентов баз знаний.

База знаний библиотеки компонентов баз знаний содержит каталогизированный набор компонентов, операциями машины обработки знаний библиотеки компонентов являются операции поиска и редактирования компонентов в этом каталоге, пользовательский интерфейс поддерживает соответствующие команды запуска этих операций. В структурную декомпозицию базы знаний библиотеки компонентов входят следующие разделы:

- раздел задачно-ориентированной спецификации базы знаний библиотеки компонентов,
- раздел типологии и обобщённых спецификаций компонентов различных типов,
- разделы типов компонентов, в каждом из которых представлены спецификации и описания на исходных текстах компонентов соответствующего типа,
- разделы, описывающие структуру каталога компонентов по различным признакам,
- разделы help-подсистемы.

3. Гипертекстовое представление семантических моделей баз знаний

Для представления исходных текстов баз знаний используются SCg [Голенков и др., 2001] и язык гипертекстового представления в псевдоестественном виде – SCn (Semantic Code natural). Язык SCn является некоторым аналогом языка Gellish и предназначен для упрощения записи и чтения человеком исходных текстов баз знаний.

Тексты языка SCn представляют собой набор sc.n-статей. Каждая sc.n-статья описывает некоторый sc-элемент и состоит из идентификатора этого элемента и набора sc.n-полей, каждое из которых описывает одну или несколько связей

отношения (рис. 12).

Если sc.n-статья имеет единственное sc.n-поле, то такая статья называется sc.n-предложением.

В SCn различаются следующие типы полей:

- sc.n-поле общего вида,
- sc.n-поле специального вида.

По признаку ориентированности отношения:

- sc.n-поле, описывающее связи ориентированного отношения,
- sc.n-поле, описывающее связи неориентированного отношения,

По признаку арности отношения:

- sc.n-поле, описывающее связи бинарного отношения,
- sc.n-поле, описывающее связи ролевого отношения,
- sc.n-поле, описывающее связи неролевого бинарного отношения,
- sc.n-поле, описывающее связи небинарного отношения.

По признаку количества описываемых связей:

- sc.n-поле, описывающее одну связь отношения,
- sc.n-поле, описывающее несколько связей отношения.

К sc.n-полям специального вида относятся:

- sc.n-поле отношения синонимии,
- sc.n-поле отношения пояснения,
- sc.n-поле отношения подмножества,
- sc.n-поле отношения надмножества,
- sc.n-поле отношения принадлежности,
- sc.n-поле отношения семантической эквивалентности,
- sc.n-поле отношения семантической близости,
- sc.n-поле отношения разбиения,
- sc.n-поле отношения определения,
- sc.n-поле отношения примера,
- и другие.

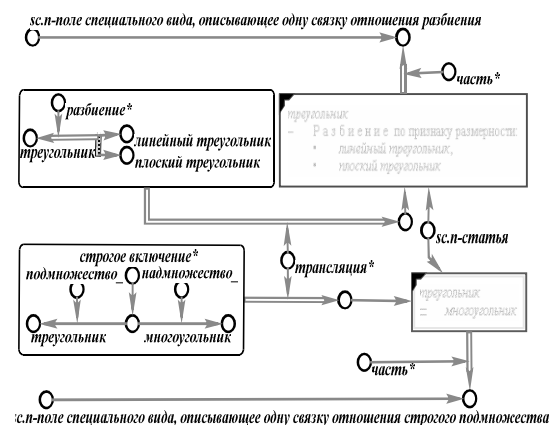


Рисунок 12 – Пример представления на SCn и эквивалентного представления на SCg

Для разметки текста в виде sc.n-статей, используется специальный язык разметки SCnML [Колб, 2012]. Некоторые sc.n-тексты могут быть транслированы в RDF.

4. Семантическая модель отладки баз знаний

4.1. Анализ качества баз знаний

Задача отладки БЗ состоит в тестировании, верификации [Ивашенко, 2011], проверке синтаксической корректности структуры баз знаний и автоматическом и ручном исправлении найденных неточностей и ошибок, основные этапы её решения представлены на рисунке 13.

В отладку базы знаний входят такие задачи как:

- верификация базы знаний,
- тестирование базы знаний,
- редактирование базы знаний.

Семантическая модель анализа качества баз знаний задаётся:

- множеством баз знаний, для которых определены наборы требований верификации в виде логических утверждений;
- множеством авторов-разработчиков фрагментов базы знаний, включая группы авторов;
- отношением авторства фрагментов базы знаний;
- конъюнкциями логических утверждений (описаний), описывающих непосредственно декларативно, либо в виде спецификации процедурных программ sc-операций:
 - нецелостность или неполноту базы знаний;
 - противоречивость или неточность базы знаний и варианты их автоматического исправления в виде:
 - изменения типа элементов;
 - удаления элементов базы знаний;
 - слияния элементов базы знаний;
 - генерации новых элементов базы знаний;
- текстами программ sc-операций редактирования и реализации проверки вышеописанных утверждений с помощью поиска и вычислений и sc-операций, обеспечивающих взаимодействие с разработчиком.

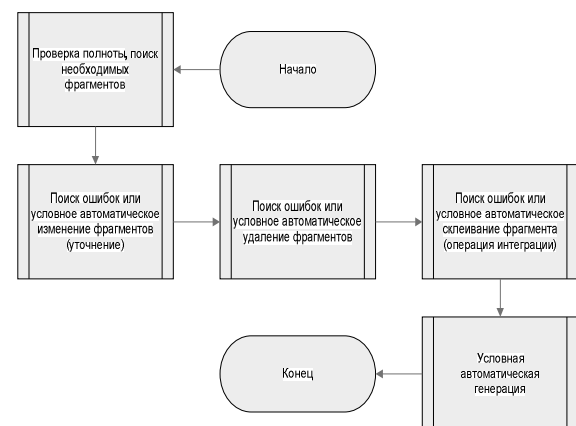


Рисунок 13 – Алгоритм верификации, основные этапы

Классификация описаний верифицируемых

ошибок представлена в [Ивашенко, 2011a].

При выявлении ошибок локализуется раздел, в котором выявлена ошибка, и, при необходимости и наличии соответствующей информации, – sc.n-поле в исходном тексте, которое описывает ошибочную связь. Сообщение о выявленной ошибке отправляется разработчику (рис. 14).

Для анализа качества используются следующие sc-операции:

- sc-операция проверки, поиска и выявления ошибок нелокальной целостности и полноты
- sc-операция проверки, поиска и выявления ошибок, связанных с целостностью, полнотой, противоречивостью и неточностью, для заданного фрагмента базы знаний
- sc-операция поиска и указания раздела или поля исходного текста с ошибкой
- sc-операция формирования сообщений указанному разработчику об ошибках в указанном разделе
- sc-операции выявления и подсчёта потенциально синонимичных элементов базы знаний

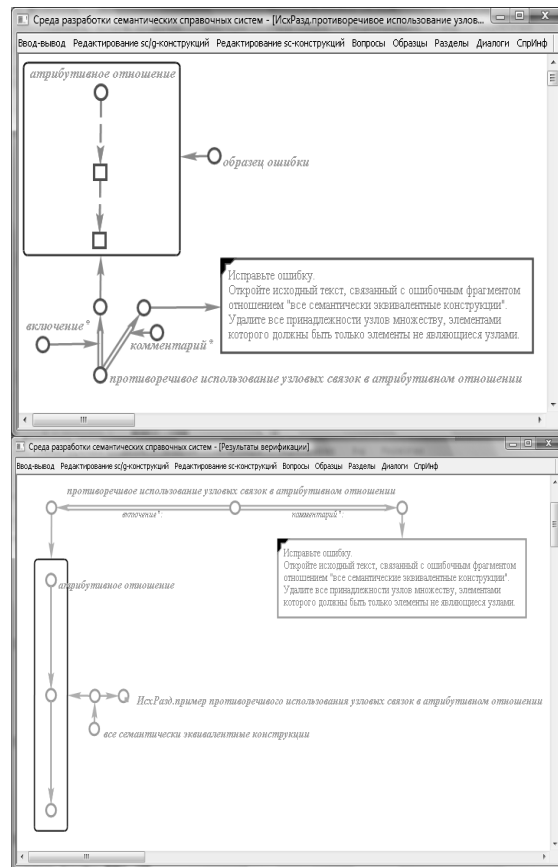


Рисунок 14 – Пример описания ошибочной ситуации противоречия и пример сообщения об обнаруженном противоречии

4.2. Тестирование баз знаний

Задача тестирования баз знаний требует условий и средств поддержки безотказной и надёжной

работы системы и возможности восстановления от сбоев. В отсутствие таких средств тестирование осуществляется большей частью вручную разработчиком на собственный риск с ручным восстановлением работоспособности в случае сбоев системы.

Семантическая модель тестирования баз знаний задаётся:

- множеством баз знаний, для которых определены наборы требований тестирования в виде логических утверждений;
- множеством авторов-разработчиков фрагментов базы знаний, включая группы авторов;
- наборами прав доступа разработчиков к фрагментам базы знаний;
- отношением авторства фрагментов базы знаний;
- конъюнкциями логических утверждений, описывающих непосредственно декларативно, либо в виде спецификации процедурных программ sc-операций тестовые команды и запросы к разрабатываемой базе знаний и ожидаемые правильные и ошибочные ответы;
- текстами программ sc-операций реализации управления правами доступа и состоянием базы знаний;
- текстами программ sc-операций редактирования и реализации проверки вышеописанных утверждений с помощью поиска и вычислений и sc-операций, обеспечивающих взаимодействие с разработчиком.

Классификация описаний верифицируемых ошибок представлена в [Ивашенко, 2011a].

Для тестирования баз знаний используются следующие sc-операции:

- sc-операция ввода произвольно-конструируемого запроса или команды к базе знаний
- sc-операция выявления несанкционированного доступа (нарушения целостности) базы знаний
- sc-операция восстановления состояния базы знаний
- sc-операция сохранения текущего состояния базы знаний
- sc-операции управления правами доступа к фрагменту базы знаний
- sc-операция формирования сообщений указанному разработчику об ошибках в указанном разделе
- sc-операции редактирования спецификации базы знаний (требований верификации)

4.3. Редактирование баз знаний

Редактирование базы знаний может быть осуществлено разработчиком с помощью классов пользовательских команд, которые задаются с помощью действий. Кроме этого разработчик может редактировать исходные тексты базы знаний. Исправление ошибок в исходных текстах и

синхронизация исходных текстов и базы знаний в памяти системы осуществляется разработчиком вручную.

Семантическая модель редактирования баз знаний задаётся:

- множеством баз знаний, для которых определены команды и операции редактирования;
- множеством авторов-разработчиков фрагментов базы знаний, включая группы авторов;
- наборами прав доступа разработчиков к фрагментам базы знаний;
- отношениями между редактируемыми разделами и фрагментами (полями) базы знаний;
- отношением авторства фрагментов базы знаний;
- текстами программ sc-операций реализации управления правами доступа и состоянием базы знаний;
- текстами программ sc-операций редактирования и поиска и формирования выборок редактируемых элементов;
- текстами программ sc-операций, обеспечивающих взаимодействие с разработчиком.

Для редактирования баз знаний используются следующие sc-операции:

- sc-операции поиска элементов базы знаний:
 - sc-операция поиска подразделов раздела базы знаний
 - sc-операция поиска надраздела базы знаний
 - sc-операция поиска элементов множества
 - sc-операция поиска по образцу
 - sc-операция экстенционального поиска
- sc-операции формирования выборки (выделения) найденных элементов базы знаний:
 - теоретико-множественные sc-операции над выборками
 - sc-операция объединения выборок
 - sc-операция пересечения выборок
 - sc-операция вычитания выборок
- sc-операция удаления элементов;
- sc-операция удаления выборки элементов;
- sc-операция изменения типа указанного элемента;
- sc-операции генерации элементов заданного типа, заданных инцидентными элементами;
- sc-операции генерации элементов заданных образом.

ЗАКЛЮЧЕНИЕ

Средства проектирования баз знаний обеспечивают поддержку решения задачи отладки баз знаний, включая интеграцию баз знаний. В состав средств проектирования баз знаний входят:

- внешний интерфейс средств проектирования БЗ
 - пользовательский интерфейс
 - средства программного ввода\вывода и трансляторы
- отладчик баз знаний
 - редактор баз знаний

- верификатор баз знаний
- средства тестирования баз знаний

- интегратор баз знаний.

Каждый из компонентов имеет свою базу знаний и систему операций. Базы знаний перечисленных компонентов образуют базу знаний средств проектирования баз знаний.

База знаний средств проектирования баз знаний содержит определяемую унифицированной моделью баз знаний метаинформацию о проектируемой базе знаний и имеет свою структурную декомпозицию, в которую входят следующие разделы:

- разделы описания процесса проектирования базы знаний и истории взаимодействия с разработчиком базы знаний;
- разделы описания методики проектирования и классов ошибок, присутствие которых выявляется при верификации базы знаний;
- разделы программ операций отладки и интеграции соответствующей машины обработки знаний и их спецификаций;
- разделы проектируемой базы знаний.

Пользовательский интерфейс средств проектирования баз знаний поддерживает необходимый набор соответствующих классов пользовательских команд и использует для общения с пользователем специальные языки (SCg [Голенков и др., 2001], SCn), которые обеспечивают наглядность, совместимость с современными web-технологиями и упрощают процесс наполнения, анализа и редактирования базы знаний.

Рассмотренные модели и средства в рамках семантической технологии проектирования баз знаний [OSTIS, 2010] обеспечивают разработку баз знаний путём добавления и интеграции [Гулякина и др., 2004] в разрабатываемую базу знаний многократно используемых компонентов баз знаний, что позволяет сократить сроки её разработки.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Гаврилова и др., 2000] Гаврилова, Т.А. Базы знаний интеллектуальных систем / Т. А. Гаврилова, В.Ф. Хорошевский. – СПб.: Питер, 2000.
- [Голенков и др., 2001] Голенков, В.В. Представление и обработка знаний в графодинамических ассоциативных машинах / В. В. Голенков [и др.] – Мн. : БГУИР, 2001.
- [Ивашенко, 2003] Представление нейронных сетей и систем продукции в однородных семантических сетях. / В.П. Ивашенко // Известия Белорусской инженерной академии. – 2003. – №1(15)/1. – с.184-188.
- [Гулякина и др., 2004] Гулякина Н.А. Ивашенко В.П. Интеграция знаний в информационных системах. / Н.А. Гулякина, В.П. Ивашенко // Доклады БГУИР. – 2004. – №6. – С. 113-119.
- [Ивашенко, 2004] Применение однородных семантических сетей для представления знаний о нестационарных предметных областях. / В.П. Ивашенко // Известия Белорусской инженерной академии. – 2004. – №1(17)/3. – с.77-80.
- [Ивашенко, 2009а] Ивашенко В.П. Семантические модели баз знаний / В.П. Ивашенко Информационные системы и технологии (IST'2009): материалы V Междунар. конф.-форума в 2-х ч. Ч.2 – Минск: А.Н.Вараксин, 2009.- с.125-128.

- [Ивашенко, 2009b] Ивашенко В.П. Алгоритмы верификации и интеграции баз знаний. Вестник Брестского государственного технического университета, БрГТУ, 2009, №5.
- [Ивашенко, 2011a] Ивашенко В.П. Семантическая технология компонентного проектирования баз знаний. Материалы Международной научн.-техн. Конференции OSTIS, 2011: Минск, Республика Беларусь, БГУИР 10-12 февраля 2011.
- [Ивашенко, 2011b] Ивашенко В.П. Алгоритмы операций отладки и интеграции баз знаний. Дистанционное обучение – образовательная среда XXI века: материалы VII Международной научн.-метод. конференции (Минск, 1-2 декабря 2011г.). – Минск: БГУИР, 2011, сс.227-229.
- [Кудрявцев, 2008] Кудрявцев Д. В. Практические методы отображения и интеграции онтологий. Семинар Знания и онтологии *Elsewhere*. КИИ-2008, Дубна, 2008.
- [Колб, 2012] Колб, Д. Г. Web-ориентированная реализация семантических моделей интеллектуальных систем / Д. Г. Колб // Материалы международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем» – Минск, 2012
- [Хорошевский, 2008] Хорошевский, В.Ф. Пространства знаний в сети Интернет и Semantic Web (Часть 1) / В. Ф. Хорошевский // Искусственный интеллект и принятие решений. - 2008. - № 1. - С.80-97.
- [Aumüller et al., 2005] D. Aumüller, H. Do, S. Massmann, E. Rahm. 2005. Schema and ontology matching with COMA++. Proc. of the 2005 International Conference on Management of Data, pp. 906-908.
- [DAML, 2006] DAML.org: The DARPA Agent Markup Language Homepage, Электронный ресурс. Режим доступа: <http://www.daml.org/>.
- [Doan and Halevy, 2005] A. H. Doan, A. Y. Halevy. Semantic integration research in the database community: A brief survey. AI magazine, 26(1), 2005
- [CLIPS, 1991] CLIPS Version 5.1 User's Guide, NASA Lyndon B. Johnson Space Center, Software Technology Branch, Houston, TX, 1991.
- [CoFI: CASL-Summary, 2004] CoFI Language Design Group. Casl summary. In Casl Reference Manual, LNCS 2960 (IFIP Series), part I. Springer, 2004.
- [CycL, 2002] The Syntax of CycL. Электронный ресурс. Режим доступа: <http://www.cyc.com/cycdoc/ref/cycl-syntax.html>.
- [Gangemi et al., 1996] A. Gangemi, G. Steve and F. Giacomelli, 1996. "ONIONS: An Ontological Methodology for Taxonomic Knowledge Integration", ECAI-96 Workshop on Ontological Engineering, Budapest, August 13th.
- [Genesereth and Fikes, 1992] Michael R. Genesereth and Richard E. Fikes. Knowledge interchange format version 3.0 reference manual. Logic Group First Draft January 1992 Report Logic-92-1 Current Version June 1992.
- [Giunchiglia et al., 2006] F. Giunchiglia, M. Marchese and I. Zaihrayeu (2006). Encoding classifications into lightweight ontologies. University of Trento Technical Report # DIT-06-016, March 2006.
- [Jean-Mary et al., 2007] Jean-Mary Y., Kabuka, M. ASMOV: Ontology Alignment with Semantic Validation. Joint SWDB-ODBS Workshop, September 2007, Vienna, Austria, 15-20.
- [IDEF5, 1994] Information Integration for Concurrent Engineering (IICE). IDEF5 Method Report. - Knowledge Based Systems, Inc. , 1408 University Drive East College Station, Texas, USA, September 21. 1994.
- [ISO13250] ISO/IEC 13250:2003 Topic maps – representation and interchange of knowledge, with an emphasis on the findability of information.
- [ISO24707] ISO/IEC 24707:2007 – Information technology – Common Logic (CL): a framework for a family of logic-based languages.
- [Maltese et al., 2010] V. Maltese, F. Giunchiglia, A. Autayeu: Save up to 99% of your time in mapping validation In Proceedings of ODBASE, 2010.
- [Martin Ph., 2002] Martin Ph. (2002). Knowledge representation in CGLF, CGIF, KIF, Frame-CG and Formalized-English. Proceedings of ICCS 2002, 10th International Conference on Conceptual Structures (Springer, LNAI 2393, pp. 77-91), Borovets, Bulgaria, July 15-19, 2002.
- [McGuinness et al., 2000] D. L. McGuinness, R. Fikes, J. Rice, and S. Wilder. The chimaera ontology environment. In Proceedings of AAAI, pages 1123-1124, 2000.
- [Michael Kifer et al., 1995] Michael Kifer, Georg Lausen, James Wu: Logical Foundations of Object-Oriented and Frame-Based Languages. Journal of ACM 42(4): 741-843 May (1995).
- [Mustafa Jarrar et al., 2008] Mustafa Jarrar and Robert Meersman: "Ontology Engineering -The DOGMA Approach". Book Chapter (Chapter 3). In Advances in Web Semantics I. Volume LNCS 4891, Springer, 2008.
- [Nagy et al., 2010] Miklos Nagy and Maria Vargas-Vera. Towards an Automatic Semantic Data Integration: Multi-Agent Framework Approach. Sematic Web. Gang Wu (ed), Chapter 7, pp. 107-134; In-Tech Education and Publishing KG; 2010, ISBN 978-953-7619-54-1.
- [Novak et al., 2008] Joseph D. Novak & Alberto J. Canas, The Theory Underlying Concept Maps and How to Construct and Use Them. Florida Institute for Human and Machine Cognition. Pensacola FL, 32502. Technical Report IHMC CmapTools 2006-01 Rev 2008-01.
- [OSTIS, 2010] Открытая семантическая технология проектирования интеллектуальных систем [Электронный ресурс]. – 2010. - Режим доступа: <http://ostis.net>. – Дата доступа: 2.11.2010
- [Smith et al., 2007] Smith, B.; Ashburner, M.; Rosse, C.; Bard, J.; Bug, W.; Ceusters, W.; Goldberg, L. J.; Eilbeck, K. et al. (2007). "The OBO Foundry: Coordinated evolution of ontologies to support biomedical data integration". Nature Biotechnology 25 (11): 1251-1255. doi:10.1038/nbt1346.
- [Sowa et al., 2008] Sowa, J. Conceptual Graphs/ John F. Sowa, F. van Harmelen, V. Lifschitz, B. Porter// eds., Handbook of Knowledge Representation, Elsevier, 2008, pp. 213-237.
- [W3C:DAML+OIL, 2001] DAML+OIL (March 2001) Reference Description: W3C Note 18 December 2001. Dan Connolly, Frank van Harmelen, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider, Lynn Andrea Stein and Lucent Technologies, Inc, Электронный ресурс. Режим доступа: <http://www.w3.org/TR/daml-oil-reference>.
- [W3C:RDFS, 2004] RDF Vocabulary Description Language 1.0: RDF Schema: W3C Recommendation 10 February 2004 Электронный ресурс. / eds. Dan Brickley, R.V. Guha. Режим доступа: <http://www.w3.org/TR/rdf-schema/>, свободный.
- [W3C:RIF, 2010] RIF Overview: W3C Working Group Note 22 June 2010. Электронный ресурс. / eds. Michael Kifer, Harold Boley. Режим доступа: <http://www.w3.org/TR/2010/NOTE-rif-overview-20100622/>.
- [W3C:OWL, 2004] OWL Web Ontology Language. Overview: W3C Recommendation 10 February 2004 Электронный ресурс. /eds. Deborah L. McGuinness, Frank van Harmelen. Режим доступа: <http://www.w3.org/TR/owl-features/>.
- [W3C:OWL2, 2009] OWL 2 Web Ontology Language Document Overview: W3C Recommendation 27 October 2009. Электронный ресурс. / eds. W3C OWL Working Group. Режим доступа: <http://www.w3.org/TR/owl2-overview/>.
- [W3C:SPARQL, 2008] SPARQL Query Language for RDF: W3C Recommendation 15 January 2008. Электронный ресурс. / eds. Eric Prud'hommeaux, Andy Seaborne. Режим доступа: <http://www.w3.org/TR/rdf-sparql-query/>.
- [XTM, 2001] XML Topic Maps (XTM) 1.0: TopicMaps.Org Specification. Members of the TopicMaps.Org Authoring Group. Электронный ресурс. / eds. Steve Pepper, Graham Moore. Режим доступа: <http://www.topicmaps.org/xtm/>.
- [Van Renssen, 2005] Van Renssen, Andries (2005). Gellish: A Generic Extensible Ontological Language. Delft University Press. ISBN 90-407-2597-4. <http://repository.tudelft.nl/view/ir/uuid%3Ade26132b-6f03-41b9-b882-c74b7e34a07d/>.

SEMANTIC MODELS AND TOOLS FOR KNOWLEDGE BASE INTEGRATION AND DEBUGGING

Ivashenko V.P.

*Belarusian State University of Informatics and
Radioelectronics, Minsk, Republic of Belarus*

ivashenko@bsuir.by

The work concerns structure and application of tools component knowledge base design technology (OSTIS) for knowledge base integration, verification and correction. This technology is based on homogeneous semantic network having set-theoretic semantic.