



OSTIS-2011

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822

СЕМАНТИЧЕСКАЯ ТЕХНОЛОГИЯ КОМПОНЕНТНОГО ПРОЕКТИРОВАНИЯ БАЗ ЗНАНИЙ

В.А. Ивашенко (ivashenko@bsuir.by)

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

Рассматриваются составляющие технологии компонентного проектирования баз знаний, использующей для представления знаний однородные семантические сети с теоретико-множественной семантикой и ориентированной на расширение контингента разработчиков и сокращение сроков проектирования

Ключевые слова: база знаний, технология, семантическая сеть, интеллектуальная система.

Введение

Для широкого применения интеллектуальных систем, способных повысить качество решения прикладных задач, необходимо большое количество баз знаний. Для быстрой разработки достаточного количества баз знаний, кроме наличия средств разработки интеллектуальных систем, обеспечивающих разработку и проектирование различных компонентов интеллектуальной системы, включая базу знаний, требуется наличие соответствующей отлаженной технологии проектирования баз знаний. Таким образом, целью является поиск или создание технологии, которая обеспечивает быструю и качественную разработку баз знаний.

Проблемы

К проблемам разработки баз знаний можно отнести следующие:

- в силу различных ограничений велики сроки разработки баз знаний (отсутствие развитых технологий разработки):
 - 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 конкретной оболочке (CLIPS), 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;
 - 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;
 - 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.

Текущее состояние преодоления проблем

Среди средств, которые могут рассматриваться в качестве основы для разработки баз знаний, можно выделить: 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) [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 представления знаний в разных системах на сегодняшний день используются различные языки (см. таблицу 1).

Таблица 1 – Языки представления знаний и описания онтологий

Год	Язык	Синтаксис						Модель представления знаний с соответствующей семантикой				
		линейный			URI	графовый	гиперграфовый	фреймовая	правила	семантическая сеть	логическая	другое
		мультимедиа	иерархически структурированный	линейный								
1976	Conceptual Graph			+		+				+	+	
1989	Frame-logic			+				+				ООП
1990	Knowledge Interchange Format		+								FOL	
1994	Integrated DEFinition for Ontology Description Capture MEthod (IDEF5)			+		+				+		

Окончание табл.1

1995	Common Algebraic Specification Language			+							FOL мат. индукция функции HOL Modal LTL CSP	
1996	Concept maps/UML			+		+						
1998	RDF/RDFS	3		+	+	+				+		
1999	DARPA Agent Markup Language	3		+	+	+			+	+		
2000	CycL		+								HOL Modal микро-теории	
2001	Ontology Inference Layer	3		+	+	+		+			DL	
2001	RDF/OWL-Lite	3		+	+	+					DL	
2001	RDF/OWL-DL	3		+	+	+					DL	
2001	RDF/OWL-FULL	3		+	+	+		+			HOL	
2003	Topic Maps XTM/LTM/CML/GML			+	+		+			+		семанти- ческий гиперграф
2003	Common Logic		+	+						+	FOL	
2003	Developing Ontology-Grounded Methods and Applications	5								+		
2005	Gellish	3										ЕЯ
2005	RDF/Rule Interchange Format	3		+	+	+			+		DL (Default Logic)	
2006	Open Biomedical Ontologies			+						+		
2007	Neno/FHAT	3		+	+	+				+		процедуры
2008	RDF/OWL2	3		+	+	+					DL	
2008	RDF/OWL2 EL	3		+	+	+					DL	
2008	RDF/OWL2 RL	3		+	+	+			+			

DL – дескриптивные логики, FOL – логика предикатов первого порядка, HOL – логика предикатов второго порядка и выше, LTL – линейная темпоральная логика, Default Logic – логики умолчаний, Modal – модальная логика, URI – унифицированный (единообразный) идентификатор ресурса, CSP – взаимодействующие последовательные процессы (последовательные процессы Хоара), LISP – синтаксис языка Лисп, 3 – триплеты, 5 – пятёрки (квинтплеты).

Кроме очевидных, дополнительно к недостаткам 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 используются такие языки, как: RDQL, squish, SPARQL, KQML, DMX, Datalog, TMQL, ERROL, RuleML, RQL, OQL, TQL, VERSA, DQL и др.

К существующим на настоящий момент средствам интеграции онтологий можно отнести: Optima, Prompt, Ontolingua, Chimaera, ONION и др. При интеграции знаний используются различные подходы и методы такие, как например: выравнивание онтологий (ontology alignment), сравнение и отображение онтологий (ontology matching & ontology mapping), S-Match и другие.

Однако все эти средства и методы имеют ограничения и не решают в полной мере вышеперечисленных проблем.

Предлагаемый подход

В основе предлагаемой технологии проектирования баз знаний лежат следующие положения: представление знаний семантическими сетями, модульное проектирование баз знаний, интеллектуализация средств поддержки проектирования баз знаний и семантическое представление знаний. Эта технология включает: унифицированную модель баз знаний; библиотеку ip-компонентов (повторно используемых фрагментов, ip – intellectual property) баз знаний и инструментальные средства проектирования баз знаний, составляющие среду проектирования баз знаний; методику проектирования и интеграции баз знаний; семантическую help-систему. В процессе разработки технологии решались следующие задачи:

- разработка унифицированной модели представления знаний;
- разработка унифицированной модели баз знаний;
- разработка модели ip-компонентов баз знаний;
- разработка методов и алгоритмов автоматизации проектирования баз знаний;
- реализация среды проектирования баз знаний.

Среда проектирования баз знаний состоит из двух составляющих: инструментальных средств проектирования баз знаний и библиотеки ip-компонентов баз знаний. Инструментальные средства основываются на унифицированной модели баз знаний и состоят из базы знаний, машины обработки знаний и пользовательского интерфейса. Принципы и порядок использования среды проектирования баз знаний регламентируются методикой проектирования и интеграции баз знаний, входящей в состав технологии.



Рисунок 1 – Фрагмент онтологии модели баз знаний

Унифицированная модель баз знаний

Унифицированная модель баз знаний описывает базы знаний специального вида, в которых знания представлены с использованием унифицированной модели представления знаний [Ивашенко, 2009]. В унифицированной модели баз знаний база знаний – связная конечная 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 [Ивашенко, 2009]. Информационная конструкция задаётся, множеством знаков, связей отношения инцидентности между ними и семантикой.

Стоит отметить, что следует различать понятие базы знаний и понятие формальной теории – первая является конечным множеством знаков и связей инцидентности между ними, а вторая – конечным или бесконечным множеством информационных конструкций, выражающих аксиомы или доказуемые утверждения.

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, 2009]. 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, 2009].

Представление знаний в унифицированной модели баз знаний

Обеспечение унифицированного представления знаний, которые бы отвечали некоторым общим критериям качества в рамках унифицированной модели баз знаний, требует разработки модели представления знаний и выявления в её рамках единой системы мер и критериев оценки качества представленных знаний, которые бы работали для любых видов знаний и не зависели бы от особенностей программной реализации. Унифицированная модель представления знаний 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 [OSTIS, 2010]. Особенности SC-кода являются: простой алфавит, содержащий узлы и дуги, простой синтаксис, базовая теоретико-множественная интерпретация. На множестве sc-языков определены отношения sc-подязыка и трансляции (кодировки). Множество всевозможных объединений текстов этих языков рассматривается как интегрированный sc-язык представления знаний. 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. 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.

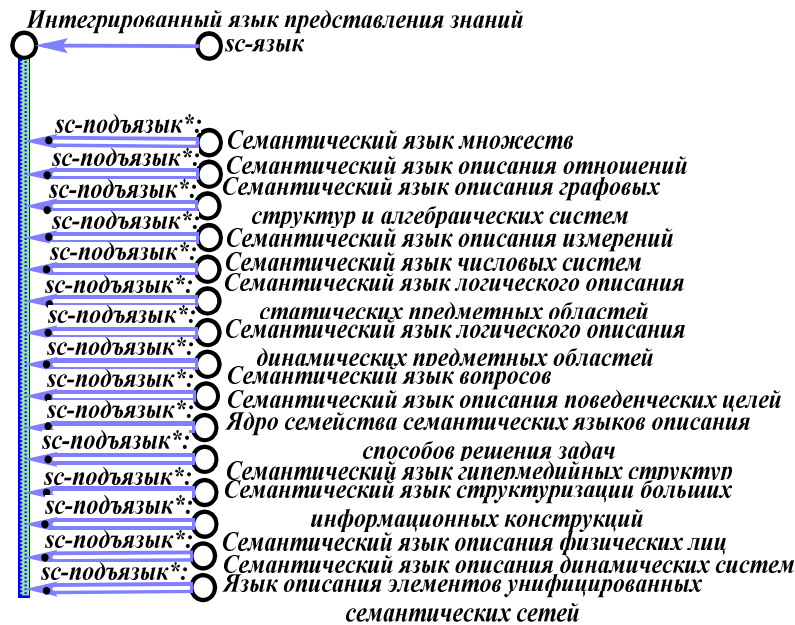


Рисунок 2 – Фрагмент онтологии унифицированной модели знаний



Рисунок 3 – Фрагмент онтологии унифицированной модели знаний



Рисунок 4 – Фрагмент онтологии семантического языка описания отношений

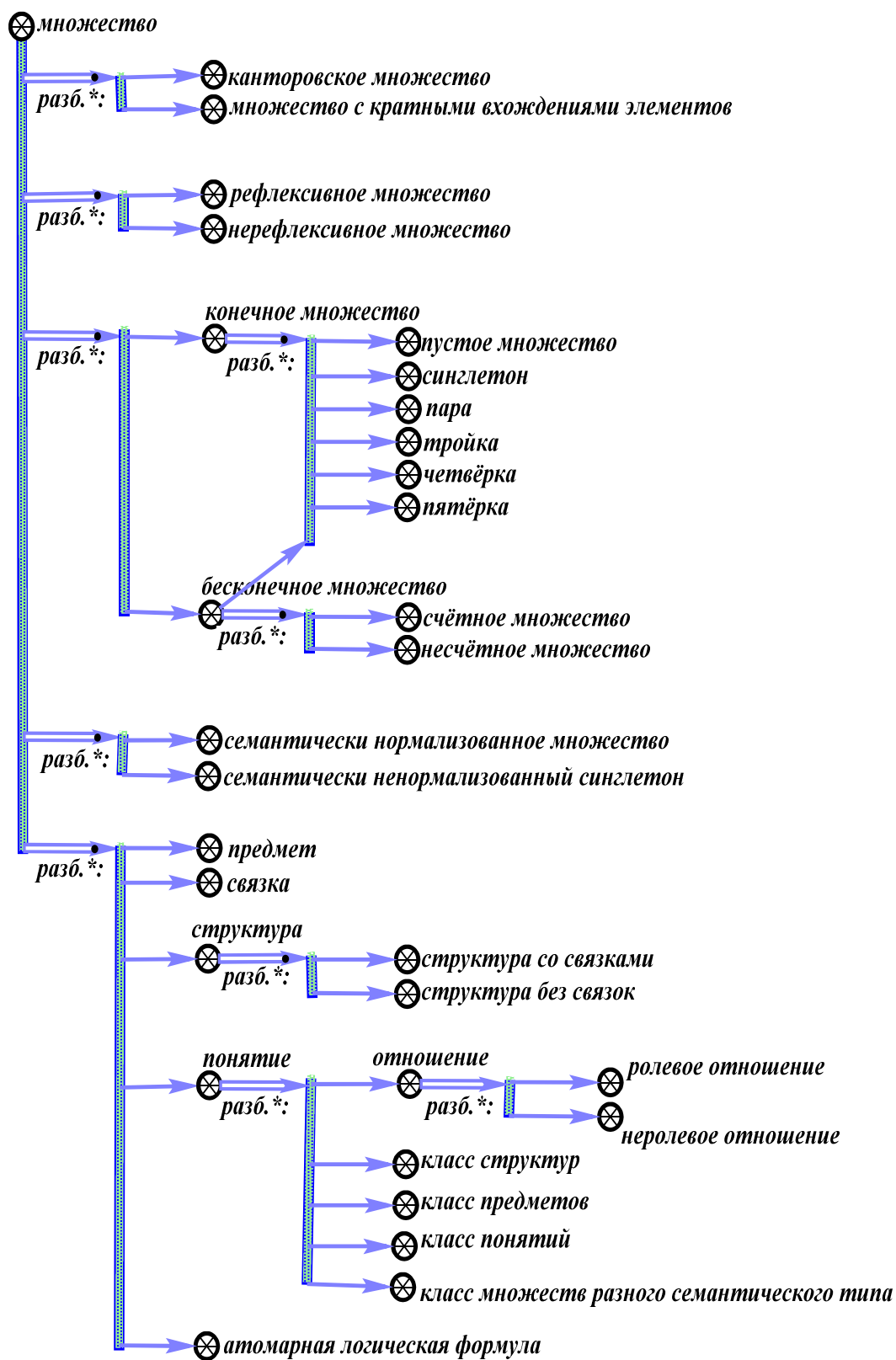


Рисунок 5 – Фрагмент онтологии семантического языка множеств



Рисунок 6 – Фрагмент онтологии семантического языка логического описания стационарных структур

Библиотека ip-компонентов баз знаний

Библиотека ip-компонентов баз знаний основывается на модели ip-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, 2009] и других признаков база знаний может быть классифицирована как терминологический словарь, таксономия, модель, тезаурус (справочник), теория или прикладная база знаний. Эта классификация, как и классификация фрагментов баз знаний, является основой модели ip-компонентов баз знаний.

Модель ip-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.



Рисунок 7 – Типология ir-компонентов баз знаний

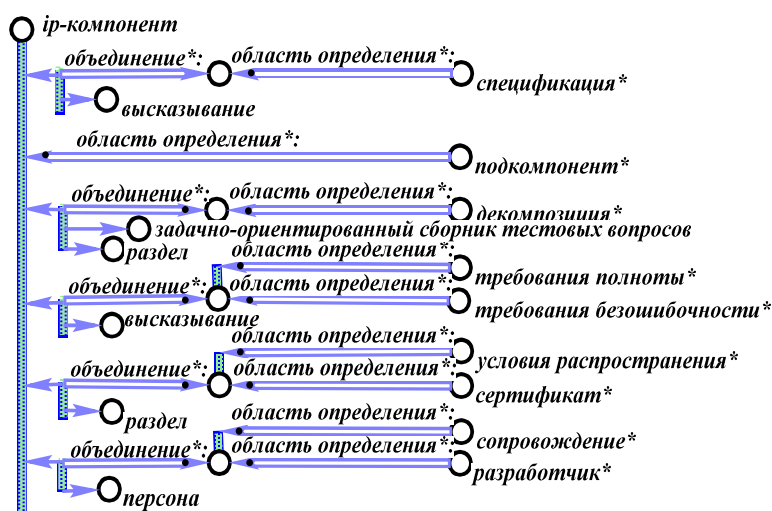


Рисунок 8 – Фрагмент онтологии модели ir-компонентов

БЗ.Семантический язык гипермедиальных структур

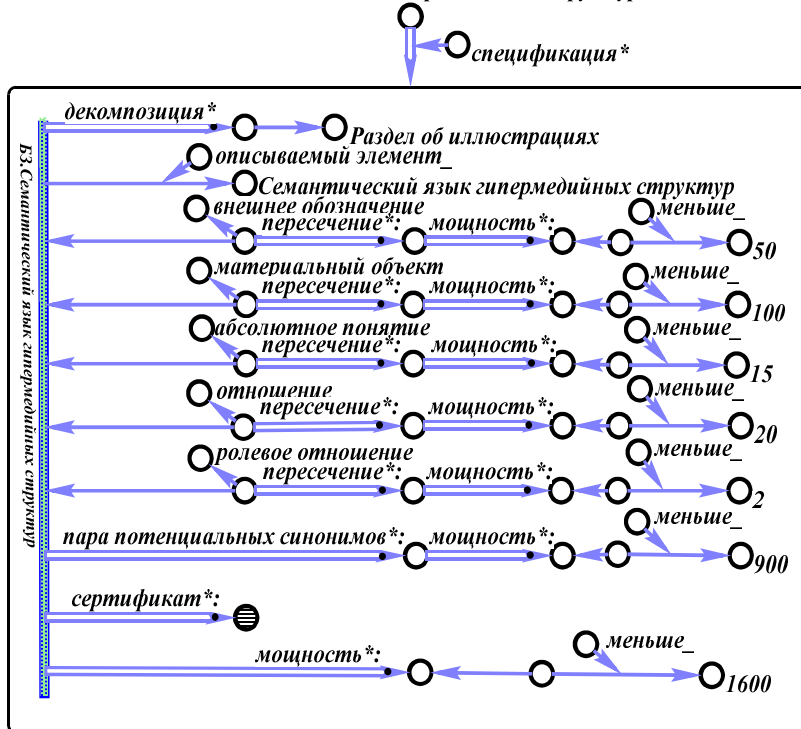


Рисунок 9 – Спецификация ip-компонентов баз знаний

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

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

Средства проектирования баз знаний на основе SC-кода

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 машины обработки знаний и их спецификаций;
- разделы проектируемой базы знаний.

Операциями машины обработки знаний средств проектирования баз знаний являются операции поддержки задач редактирования, интеграции, верификации, тестирования и оценки качества баз знаний. Соответствующие методы и алгоритмы автоматизации проектирования баз знаний предназначены для ускорения процесса проектирования базы знаний, сокращения физических

затрат разработчика и связанных с человеческим фактором ошибок. Методы и алгоритмы автоматизации направлены на решение задач редактирования, верификации и повышения качества баз знаний. Алгоритмы редактирования используют методы планирования и алгоритмы поиска и генерации по образцу. Алгоритмы верификации используют методы планирования, алгоритмы поиска, включая поиск по образцу, и алгоритмы формирования сообщений пользователю. Алгоритмы повышения качества баз знаний используют методы и алгоритмы проверки связности, поиска и анализа фрагментов онтологии, выявления и упорядочения пар потенциально синонимичных элементов, канонической разметки графа.

Алгоритмы верификации можно разделить на алгоритмы, использующие поиск по безусловно заданному набору образцов и по условно заданному набору образцов. Среди алгоритмов поиска по заданному набору образцов можно выделить алгоритм поиска по одному заданному образцу. В зависимости от структуры этого образца этот алгоритм способен с целью верификации выявлять следующие классы ошибок: классы ошибочных противоречий, классы ошибочных ситуаций избыточности.

В зависимости от типа *ip*-компонента формируется структура образца. Учитывается объём и глубина содержания верифицируемых понятий, представленных ключевыми элементами. Так, например, для верификации реляционной модели предметной области достаточно выделить классы верифицируемых отношений, включая и классы верифицируемых атрибутов. Так как отношения могут быть определены не только на объектах предметной области, но и на классах этих объектов, то глубина содержания констант, включённых в образец может достигать пяти или шести. Как правило глубина содержания в образце верификации должна превышать на единицу глубину содержания верифицируемых понятий. Таким образом образцы верификации определяются для различных типов *ip*-компонентов.

При классифицирующей идентификации верификации по одному заданному образцу используются следующие принципы:

- если образец содержит элементы, не входящие в содержание ни одной из констант, то классификация производится не по содержанию, а по суперсодержанию (множество всех элементов, доступных по выходящим дугам принадлежности, непринадлежности и нечёткой принадлежности);
- множество констант образца разбивается на упорядоченные классы по возрастанию количества вторичных элементов поиска на пути к граничным элементам пересечения (супер)содержания константы и общего (супер)содержания остальных констант;
- в рамках выделенных классов константы упорядочиваются по возрастанию быстроты поиска граничных элементов;
- в рамках выделенных подклассов константы упорядочиваются по объёму собственного (супер)содержания;
- при необходимости дальнейшее упорядочение происходит на основании алгоритма канонической разметки графа образца;
- при необходимости окончательное упорядочение происходит по лексикографическому порядку основных терминов констант.

Идентификация и классификация образцов осуществляется на основе лексикографического порядка сформированных классов.

К классу ошибочных противоречий относятся, например, следующие подклассы.

- Противоречивая принадлежность элемента указанного типа указанному множеству.
- Противоречивая принадлежность элемента указанного типа строгому подмножеству указанного множества.
- Противоречивая принадлежность элемента указанного типа множеству указанного семейства.
- Противоречивая принадлежность элемента указанного типа строгому подмножеству множества указанного семейства.
- Противоречивая принадлежность элемента указанного типа множеству строгого подмножества указанного семейства.
- Противоречивая принадлежность элемента указанного типа строгому подмножеству множества строгого подмножества указанного семейства.
- Противоречивая принадлежность связки указанному множеству.
- Противоречивая принадлежность указанному множеству связки двух различных (различимых) элементов.
- Противоречивая принадлежность связки отношению совпадения.

- Противоречивая принадлежность указанному множеству связки двух элементов, принадлежащий одному из которых элемент не принадлежит другому.
 - Противоречивая принадлежность связки отношению равенства.
- Противоречивая принадлежность указанному множеству связки с заданными указанными множествами ролями двух множеств с общим элементом.
 - Противоречивая принадлежность связки отношению вычитания множеств (разность, вычитаемое).
- Противоречивая принадлежность указанному множеству связки с заданными указанными множествами ролями двух элементов, принадлежащий одному из которых элемент не принадлежит другому.
 - Противоречивая принадлежность связки отношению вычитания множеств (разность, уменьшаемое).
 - Противоречивая принадлежность связки отношению включения.
 - Противоречивая принадлежность связки отношению строгого включения.
- Противоречивая принадлежность указанному множеству связки с заданными указанными множествами ролями трёх элементов, принадлежащий одному из которых элемент не принадлежит другим.
 - Противоречивая принадлежность связки отношению вычитания множеств (уменьшаемое, вычитаемое, разность).
- Противоречивая принадлежность указанному множеству связки с заданными указанными множествами ролями двух элементов, принадлежащий элементу одного из которых элемент не принадлежит другому.
 - Противоречивая принадлежность связки отношению объединения.
 - Противоречивая принадлежность связки отношению области определения.
- Противоречивая принадлежность указанному множеству связки с заданными указанными множествами ролями двух элементов, принадлежащий одному из которых элемент не принадлежит элементу другого.
 - Противоречивая принадлежность связки отношению пересечения.
- Противоречивая принадлежность связки строгому подмножеству указанного множества.
- Противоречивая принадлежность связки множеству указанного семейства.
- Противоречивая принадлежность двух связок указанному множеству.
- Противоречивая принадлежность указанному множеству двух связок с заданными указанными множествами ролями двух элементов, роли которых в этих связках противоположны.
- Противоречивая принадлежность двух связок строгому подмножеству указанного множества.
- Противоречивая принадлежность двух связок множеству указанного семейства.
- Противоречивая принадлежность множеству указанного семейства двух связок с заданными указанными множествами ролями двух элементов, роли которых в этих связках противоположны.
 - Противоречивая принадлежность связок асимметричного отношения.
- Противоречивая принадлежность двух связок двум указанным множествам.
- Противоречивая принадлежность двум указанным множествам двух связок с заданными в одной из них указанными множествами ролями двух элементов.
 - Противоречивая принадлежность связок равенства и строгого включения.
- Противоречивая принадлежность двум указанным множествам двух связок с заданными указанными множествами ролями двух элементов, роли которых в этих связках противоположны.
 - Противоречивая принадлежность связок включения и строгого включения.

К классу ошибочных ситуаций избыточности относятся, например, следующие подклассы.

- Неточные кратные не принадлежности элемента указанному множеству.
- Неточные кратные принадлежности элемента указанному множеству.
- Неточные кратные не принадлежности элемента строгому подмножеству указанного множества.
- Неточные кратные принадлежности элемента строгому подмножеству указанного множества.
- Неточные кратные не принадлежности элемента множеству указанного семейства.
- Неточные кратные принадлежности элемента множеству указанного семейства.
- Неточные две кратные принадлежности элемента множеству указанного семейства.
- Неточные три кратные принадлежности элемента множеству указанного семейства.

- Неточные кратные принадлежности элемента множеству указанного семейства.
- Неточные кратные непринадлежности элемента строгому подмножеству множества указанного семейства.
- Неточные кратные принадлежности элемента строгому подмножеству множества указанного семейства.
- Неточные кратные непринадлежности элемента множеству строгого подмножества указанного семейства.
- Неточные кратные принадлежности элемента множеству строгого подмножества указанного семейства.
- Неточные кратные непринадлежности элемента строгому подмножеству множества строгого подмножества указанного семейства.
- Неточные кратные принадлежности элемента строгому подмножеству множества строгого подмножества указанного семейства.
- Неточная принадлежность связки указанному множеству.
- Неточная принадлежность указанному множеству связки двух множеств с общим элементом.
- Неточная принадлежность связки строгому подмножеству указанного множества.
- Неточная принадлежность связки множеству указанного семейства.

Классы ошибочных ситуаций неполноты описываются в основном условно заданным набором образцов.

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

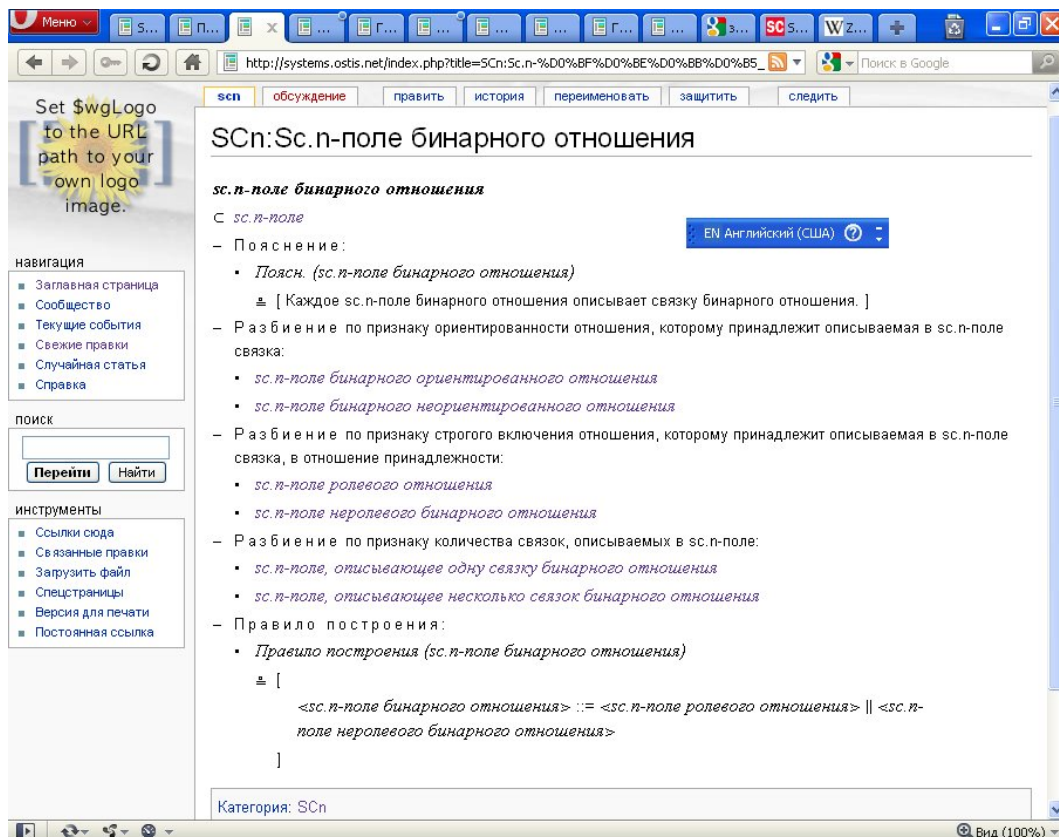


Рисунок 10 – Оформление исходных текстов баз знаний

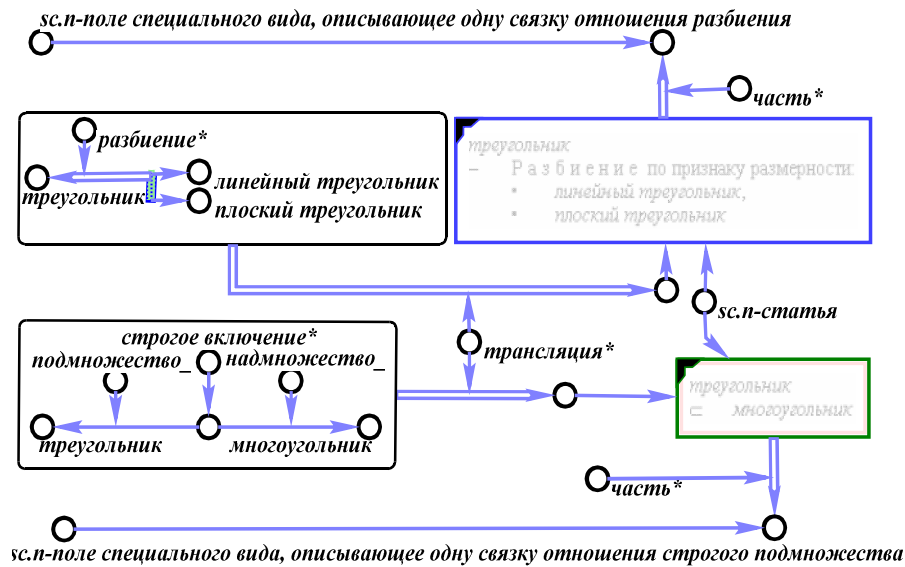


Рисунок 11 – Пример представления на SCn и эквивалентного представления на SCg

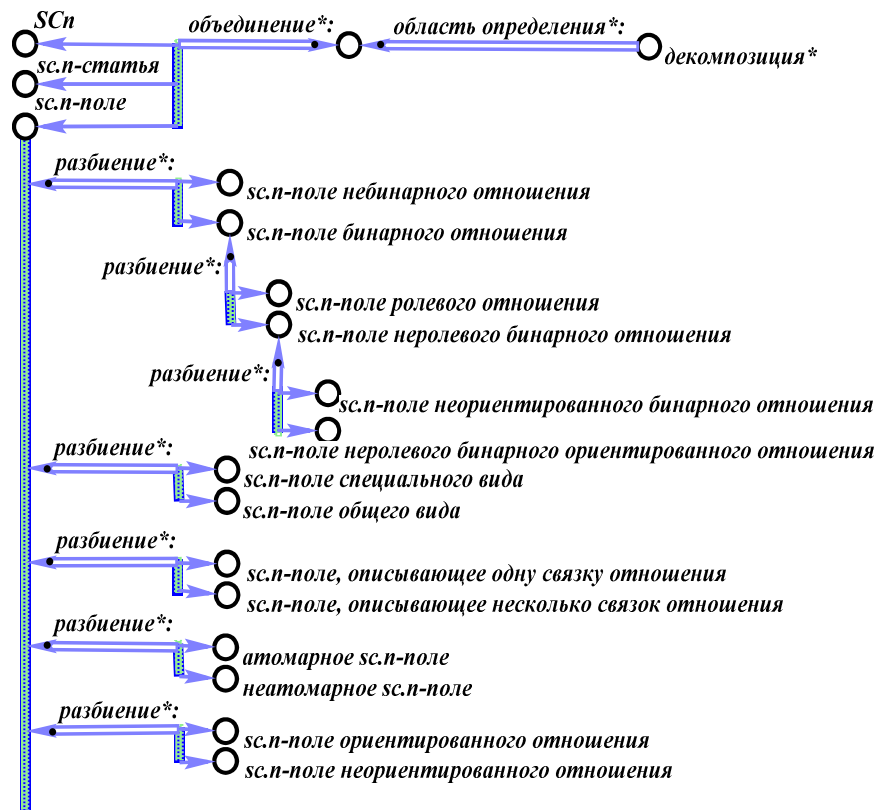


Рисунок 12 – Фрагмент онтологии SCn-кода

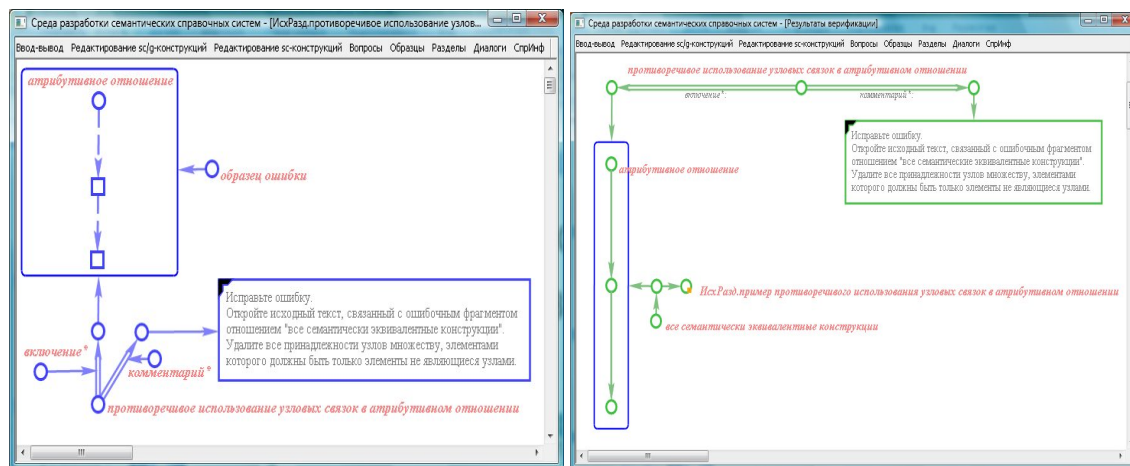


Рисунок 13 – Пример описания ошибочной ситуации противоречия и пример сообщения об обнаруженном противоречии

Методика проектирования баз знаний на основе SC-кода

В основу методики проектирования баз знаний положена концептуальная модель процесса проектирования базы знаний. Методика описывает этапы работы со средой проектирования баз знаний и их порядок. Этапы могут быть декомпозированы на другие этапы. В методике выделяются три основных класса этапов проектирования:

- задачно-ориентированное проектирование базы знаний;
- концептуальное проектирование базы знаний;
- структурное проектирование базы знаний.

Задачно-ориентированное проектирование подразумевает построение задачно-ориентированного сборника вопросов и ответов к базе знаний. При задачно-ориентированном проектировании структура базы знаний соответствует структуре классов и подклассов вопросов сборника. При формировании сборника вначале вопросы и ответы записываются в естественно-языковой форме. Затем вопросы группируются в классы на основании существующей классификации вопросов языка вопросов и знаний разработчика. Затем осуществляется терминологический поиск понятий, по терминам, которые присутствуют в вопросах и ответах. В соответствии с результатами поиска множество терминов сборника разбивается на:

- множество подходящих найденных терминов,
- множество неподходящих найденных терминов (омонимов),
- множество ненайденных терминов.

Далее, для подходящих найденных терминов в соответствии с определениями выраженных ими понятий и грамматическими правилами записываются формальные выражения, элементам которых сопоставляются некоторые из ранее неподходящих или ненайденных терминов. Формальные выражения записываются в соответствующие разделы, содержащие вопрос и ответ на него. Для записи используется один или более способов представления исходных текстов баз знаний. После записи исходных текстов они конвертируются, верифицируются и интегрируются. После каждого этапа верификации формируются два множества:

- множество разделов прошедших верификацию,
- множество ошибочных разделов.

Процесс продолжается до тех пор, пока множество ошибочных разделов не станет пустым. Оставшиеся неподходящие или ненайденные термины используются в качестве исходных терминов для начала концептуального проектирования.

Концептуальное проектирование подразумевает формирование множества объектов онтологии, требующих описания. При концептуальном проектировании структура базы знаний соответствует лексикографическому порядку внешних обозначений описываемых объектов. Для каждого оставшегося неподходящего термина выявляется вид понятия, которое он выражает: абсолютное или относительное. Относительные могут быть отличены от абсолютных с помощью следующего способа: для относительных могут быть одномоментно

истинными два утверждения заданных следующими схемами, тогда как для абсолютных понятий – нет.

«X [<глагол>] Y ([<предлог>] U)»
«X не [<глагол>] Y ([<предлог>] V)»

Далее для отношений, представляющих относительные понятия, выявляется их арность, схема и область определения. С целью обеспечения связности онтологии среди абсолютных и областей определения относительных понятий выявляются или добавляются к ним наиболее общие понятия. Каждое выявленное понятие рассматривается как описываемый элемент некоторого раздела. В соответствии с выявленными характеристиками каждого выявленного понятия устанавливается тип соответствующей статьи и раздела, в котором оно описывается. Все формируемые разделы являются не декомпозируемыми на другие разделы (атомарными) разделами, разделами, на которые непосредственно декомпозируется база знаний. Производится формирование исходных текстов разделов, описывающих объекты онтологии, с целью их наполнения в соответствии со спецификацией разделов установленного типа. Процесс продолжается конвертированием, верификацией и интеграцией до тех пор, пока множество ошибочных фрагментов не станет пустым.

Структурное проектирование подразумевает выделение и формирование структуры разделов и подразделов определённых типов в соответствии с заданными правилами. Для каждого описываемого элемента устанавливаются его роли, тип и семантическая окрестность, включающая связи между ними. В соответствии с установленными ролями и типом формируются разделы описаний этих элементов. На основании связей между описываемыми элементами устанавливаются связи между разделами описаний. На основании сформированных связей между разделами происходит формирование разделов и декомпозиции их к ранее сформированным. База знаний включается в эту систему разделов, как единственный не декомпозируемый на разделы раздел или как самый большой декомпозируемый раздел. Производится формирование исходных текстов разделов с целью их наполнения в соответствии со спецификацией разделов установленного типа. Процесс продолжается конвертированием, верификацией и интеграцией до тех пор, пока множество ошибочных фрагментов не окажется пустым.

Семантическая help-система содержит описания онтологий, которые используются языками из семейства совместимых семантических языков представления знаний.

Заключение

Особенностями и достоинствами предложенной технологии проектирования баз знаний являются: унифицированное представление баз знаний и их фрагментов; перенос акцента от традиционной инженерии баз знаний к интеграции баз знаний [Гулякина и др., 2004], т.е. к сборке баз знаний из крупных модулей, основывающейся на унифицированной модели баз знаний; обеспечение совместимости разрабатываемых баз знаний; использование визуальных методов проектирования.

Библиографический список

- [Вагин и др., 2008] Достоверный и правдоподобный вывод в интеллектуальных системах / Вагин В.Н. [и др.]; – М. : ФИЗМАТЛИТ, 2008.
- [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.
- [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, 2009] 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.
- [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.
- [OSTIS, 2010] Открытая семантическая технология проектирования интеллектуальных систем[Электронный ресурс]. – 2010. - Режим доступа: <http://ostis.net>. – Дата доступа: 2.11.2010
- [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.