

Графодинамические модели параллельной асинхронной обработки знаний (продолжение)

Голенков В.В., доктор технических наук, профессор,

Гулякина Н.А., кандидат физико-математических наук, доцент

В статье рассматриваются принципы построения технологии проектирования интеллектуальных систем, ориентированных на семантическое представление знаний, расширение контингента разработчиков и сокращение сроков проектирования.

• графодинамическая модель обработки информации • графодинамическая память • многоагентная система над графодинамической памятью • графовый язык программирования • семантическая сеть • бинарная семантическая сеть • язык визуализации семантических сетей • унификация семантических сетей • интеграция семантических сетей • семантическая технология проектирования интеллектуальных систем • семантическая технология проектирования баз знаний • семантическая технология проектирования решателей задач • семантическая технология проектирования пользовательских интерфейсов • библиотека семантически совместимых многократно используемых компонентов интеллектуальных систем.

The article presents the principles of construction technology of designing intelligent systems which are oriented on semantic representation of knowledge, expansion of the number of developers and shortening time of design are considered.

• graph-dynamic information processing model • graph-dynamic memory multiagent system over graph-dynamic memory • graph programming language • semantic network • binary semantic network • semantic network visual language, unification of semantic network • semantic network integration • semantic network merging • semantic network fusion • semantic intelligent system design technology • semantic knowledge base design technology • semantic problem solver design technology • semantic user interface design technology • library of reusable semantically compatible components for intelligent systems.

Принцип 5. Унификация семантического представления знаний

Определение структуры унифицированных семантических сетей, обеспечивающих представление различных видов знаний, предполагает разработку соответствующего стандарта, выделяющего из всего многообразия абстрактных языков семантических сетей определённый базовый универсальный язык, который мы назвали SC-кодом (Semantic Computer code).

Для того чтобы перейти от семантических сетей произвольного вида к текстам SC-кода, уточним критерии качества SC-кода, к которым относится, во-первых, переход от семантических сетей, имеющих унарные и многокомпонентные (многоместные) связи, к семантическим сетям, имеющим только бинарные связи. Такие сети будем называть бинарными семантическими сетями; во-вторых, минимизация алфавита, т.е. минимизация числа меток, используемых в семантических сетях; в-третьих, универсальность разрабатываемого языка, т.е. возможность представления любых знаний в виде текстов этого языка.

Семантическую сеть G_b , задаваемую пятёркой $\langle V_b, C_b, I_b, M_b, K_b \rangle$, будем называть **бинарной семантической сетью** только в том случае, если:

— все связи, входящие во множество C_b , являются бинарными (каждая связь имеет только два компонента: это могут быть элементы множеств V_b , C_b и K_b);

— $I_b = \{i_1, i_2\}$, где i_1 — бинарное ориентированное отношение инцидентности, связывающее бинарную связь с её компонентом; i_2 — бинарное ориентированное отношение инцидентности, связывающее бинарную ориентированную связь с её вторым компонентом. Очевидно, что $i_1 \subset i_2$.

Для любой семантической сети G , задаваемой пятёркой $\langle V, C, I, M, K \rangle$, можно построить семантически эквивалентную ей бинарную семантическую сеть G_b , задаваемую пятёркой $\langle V_b, C_b, I_b, M_b, K_b \rangle$, следующим образом:

— во множество V_b включаются все вершины (V) семантической сети G и все небинарные связи (C_n) семантической сети G ($C_n \subset G$);

— $I_b = \{i_1, i_2\}$ — это отношения инцидентности, связывающие бинарные ориентированные связи (из множества C) с их компонентами;

— $K_b = K \cup (I_b)$. Т.е. знаки отношений инцидентности, связывающих (в рамках G) небинарные связи с их компонентами, становятся в рамках G_b ключевыми узлами;

— $M_b = M$. При этом, если в алфавит M не входила метка отношения принадлежности (re), то она дополнительно вводится в алфавит M_b ;

— во множество C_b включаются все бинарные связи (C_b) семантической сети G ($C_b \subset C$); все ориентированные пары всех отношений инцидентности (I), связывающих небинарные ориентированные связи (из множества C) с их компонентами; все пары принадлежности, которые явно связывают ключевые узлы семантической сети G_b , соответствующие различным отношениям инцидентности (I) семантической сети G , с парами принадлежности, которые явно связывают небинарные связи с их компонентами и которые в рамках этих связей выполняют роли, обозначаемые соответствующими отношениями инцидентности.

Нетрудно заметить, что приведение семантической сети к бинарному виду приводит к минимизации числа отношений инцидентности. Все пары инцидентности, связывающие небинарные связи семантической сети G с их компонентами, в бинарной семантической сети C_b «превращаются» в бинарные ориентированные связи, которые принадлежат отношению принадлежности и, следовательно, должны быть помечены знаком отношения принадлежности re ($re \in M_b$).

Таким образом, каждая небинарная связь семантической сети G в семантической сети G_b трактуется как множество связываемых им элементов семантической сети, связь которого с его элементами представляется в виде дополнительно вводимых связей принадлежности. Эти связи принадлежности, в свою очередь, могут быть вторыми компонентами других связей

принадлежности, связывающих указанные связки с ключевыми узлами, обозначающими различные роли компонентов небинарных ориентированных связок исходной семантической сети.

Бинарные связки легко изображать графически (в виде линий, каждая из которых соединяет графические изображения двух связываемых элементов семантической сети). Использование только бинарных связок существенно упрощает машинное кодирование семантических сетей и, в частности, упрощает разработку специальной памяти для их хранения сетей.

В основе **минимизации алфавита** элементов лежит следующее свойство семантических сетей. Метки, входящие в состав алфавита элементов семантической сети, и ключевые узлы этой сети семантически эквивалентны, т.е. являются синтаксически различными способами выделения (задания) различных классов элементов семантической сети. При этом, в отличие от алфавита символов линейного текста, все элементы алфавита (все метки) семантической сети, как и все её ключевые узлы, имеют семантическую интерпретацию на описываемой предметной области — все они являются знаками различных классов элементов этой предметной области. Таким образом, метки элементов семантической сети без какого-либо изменения её семантики можно преобразовывать в ключевые узлы. При этом семантическая интерпретация каждого ключевого узла будет совпадать с семантикой соответствующей преобразованной метки.

Если мы имеем дело с корректно (правильно) построенной семантической сетью и записанной на некотором языке семантических сетей, то её перевод на любой другой язык семантических сетей не требует больших усилий, так как синтаксическое и семантическое устройства всех языков семантических сетей сходны. Фактически эти языки отличаются алфавитами, элементы которых легко преобразуются в ключевые узлы семантических сетей.

Итак, число меток семантической сети можно уменьшать ценой расширения множества её ключевых узлов. Вопрос в том, до какого предела это можно делать и как выглядит минимальный алфавит универсального языка семантических сетей. Универсальный язык семантических сетей — такое множество семантических сетей, элементами которых являются семантические сети, представляющие любую информацию о любой описываемой предметной области.

Из вышесказанного следует, что в состав минимального алфавита универсального языка семантических сетей, по крайней мере, должна входить метка *re*, обозначающая **отношение принадлежности**. Без этой метки невозможно описать связи ключевых узлов семантической сети с элементами обозначаемых ими классов объектов, а также невозможно осуществить переход к бинарным семантическим сетям.

Текст, принадлежащий SC-коду, т.е. **sc-текст** (sc-структура, sc-конструкция), является семантической сетью частного вида, имеющей следующие особенности:

— все связки sc-текстов — бинарные связки — **sc-коннекторы**. Неориентированные sc-коннекторы — **sc-ребра**, ориентированные — **sc-дуги**.

Множество меток элементов sc-текстов (алфавит sc-элементов, алфавит SC-кода) включает в себя метки:

- **sc-узлов** (вершин sc-текстов);
- **sc-ребер**;
- **sc-дуг общего вида**;
- **sc-дуг принадлежности**;
- **sc-дуг основного вида**.

Множество отношений инцидентности элементов sc-текстов состоит из двух следующих отношений:

- **быть компонентом sc-коннектора** (sc-ребра или sc-дуги);

— **быть вторым компонентом sc-дуги;**

— множество ключевых узлов sc-кода (ключевых узлов sc-текстов) вместе с метками sc-элементов задаёт базовую семантическую типологию sc-элементов, т.е. базовую онтологию SC-кода.

К числу ключевых узлов SC-кода, определяющих разбиение множества sc-элементов по признаку константности, относятся:

— ключевой sc-узел, обозначающий множество всевозможных константных sc-элементов (**sc-констант**). Каждая sc-константа — обозначение некоторого конкретного фиксированного объекта;

— ключевой sc-узел, обозначающий множество всевозможных переменных sc-элементов (**sc-переменных**), каждая из которых обозначает некоторый произвольный, нефиксированный объект из некоторого дополнительно уточняемого множества объектов. Используются sc-переменные в логических формулах (в т.ч. в высказываниях), в программах (в обобщённых описаниях способов решения различных классов задач), в формулировках вопросов.

К числу ключевых узлов SC-кода, определяющих разбиение множества sc-элементов по структурному признаку, относятся:

— ключевой sc-узел, обозначающий множество всевозможных **sc-коннекторов**. Более детальное разбиение множества sc-коннекторов по структурному признаку осуществляется с помощью меток sc-элементов (на **sc-ребра**, **sc-дуги общего вида**, **sc-дуги принадлежности**, **sc-дуги основного вида**);

— ключевой sc-узел, обозначающий множество всевозможных sc-узлов, каждый из которых обозначает некоторое связующее множество sc-элементов. Указанные sc-элементы — **связующие sc-узлы**. К числу ключевых узлов SC-кода, определяющих более детальную структурную типологию связующих sc-узлов, относятся:

— ключевой sc-узел, обозначающий множество всевозможных sc-узлов, каждый из которых обозначает некоторую неатомарную связь между sc-элементами, т.е. связь, не являющуюся sc-коннектором. Такие sc-узлы будем называть **неатомарными sc-связками**. Более детальная структурная типология неатомарных sc-связок задаётся такими ключевыми узлами SC-кода, как быть **унарной sc-связкой**, быть **бинарной неатомарной sc-связкой**, быть **многокомпонентной sc-связкой**, быть **ориентированной неатомарной sc-связкой**, быть **неориентированной неатомарной sc-связкой**;

— ключевой sc-узел, обозначающий множество всевозможных sc-узлов, каждый из которых обозначает некоторую структуру из sc-элементов — **sc-структуру**;

— ключевой sc-узел, обозначающий множество всевозможных sc-узлов, каждый из которых обозначает некоторый класс sc-элементов — **sc-понятия**. Более детальная структурная типология sc-понятий задаётся следующими ключевыми узлами SC-кода: быть **отношением** (классом однотипных связей), быть **бинарным отношением**, быть **унарным отношением**, быть **многоместным отношением**, быть **ориентированным отношением**, быть **неориентированным отношением**, быть **ролевым отношением** (т.е. отношением, которое является подмножеством отношения принадлежности), быть **классом структур**, быть **классом терминальных sc-узлов** (первичных sc-узлов, которые не являются обозначениями множеств sc-элементов);

— ключевой sc-узел, обозначающий множество всевозможных sc-узлов, каждый из которых обозначает некоторый объект, который не является множеством sc-элементов. Такие sc-узлы будем называть **терминальными sc-**

узлами (первичными sc-узлами). Более детальная структурная типология терминальных sc-узлов задаётся следующими ключевыми узлами sc-кода:

— быть **предметным sc-узлом**, каждый из которых обозначает некоторый реальный (материальный, физический) или вымышленный внешний объект некоторой предметной области;

— быть **sc-ссылкой**, каждая из которых обозначает либо определённый файл, который можно просматривать или в котором закодирована в определённом формате некоторая внешняя, инородная для sc-кода информационная конструкция, либо некоторую компьютерную систему, с которой можно взаимодействовать;

— быть **терминальным элементом шкалы или шаблона** (sc-элементы, для которых трудно установить обозначаемые ими объекты, поскольку эти sc-узлы просто являются терминальными элементами каких-либо шкал, шаблонов, типовых структур, с которыми устанавливаются соответствия, сравниваются, сопоставляются различные объекты и структуры).

К числу ключевых узлов SC-кода, уточняющих семантику sc-дуг принадлежности, относятся:

— **ключевые узлы SC-кода, определяющие разбиение множества sc-дуг принадлежности по признаку позитивности:**

— быть **sc-дугой позитивной принадлежности;**

— быть **sc-дугой негативной принадлежности;**

— быть **sc-дугой нечёткой принадлежности** (т.е. sc-дугой позитивность или негативность которой в текущий момент не установлена);

— **ключевые узлы SC-кода, определяющие разбиение множества sc-дуг принадлежности по признаку стационарности:**

— быть **sc-дугой стационарной принадлежности**, семантический тип которой является постоянным, не изменяющимся во времени;

— быть **sc-дугой нестационарной принадлежности**, семантический тип которой изменяется во времени.

Заметим, что **sc-дуги основного вида**, которые выделяются с помощью соответствующей метки, семантически трактуются как **sc-дуги позитивной стационарной принадлежности**.

Особенности и достоинства SC-кода

Унифицированные семантические сети (sc-тексты) — абстрактная семантическая модель знаний — **инвариант** различных способов представления и кодирования этих же знаний (в т.ч. и самих семантических сетей). Наличие такого инварианта необходимо для решения проблемы интеграции различных видов знаний. На основе унифицированных семантических сетей можно строить семантические модели различных компьютерных систем и решить проблему их интеграции.

SC-код — **абстрактный язык**: способ изображения (материализации) его текстов не уточняется. Следовательно, можно разрабатывать различные графические уточнения SC-кода (например, SCg-код), различные варианты изображения sc-текстов в виде строк символов (например, SCs-код) и машинного представления sc-конструкций в адресной памяти традиционных компьютеров, а также в специальной структурно перестраиваемой ассоциативной памяти будущих компьютеров, ориентированных на обработку баз знаний.

Все sc-элементы, кроме терминальных sc-узлов, являются обозначениями множеств, состоящих из sc-элементов — **вторичных sc-элементов**. Из этого следует то, что SC-код имеет базовую теоретико-множественную семантическую интерпретацию.

SC-код — достаточно простой компьютерный код семантических сетей, который является не «инородным» представлением семантических сетей, а их представлением в виде се-

мантических сетей, но максимально простого вида — с минимальным алфавитом и бинарными связками.

SC-код ориентирован на представление информации в компьютерной памяти и может рассматриваться как основа модели структурно перестраиваемой ассоциативной памяти будущих компьютеров, ориентированных на обработку семантических сетей. Т.е. SC-код можно рассматривать как универсальную основу машинного кодирования знаний в памяти будущих компьютеров, ориентированных на обработку семантических сетей. В такой памяти биты и байты «уступят место» sc-дугам, sc-ребрам и sc-узлам.

На базе SC-кода можно создавать целое семейство совместимых специализированных языков, ориентированных на представление разных видов знаний (логических формул и высказываний, программ, вопросов, поведенческих целей, различных видов моделей динамических систем и т.п.) таким образом, чтобы тексты всех этих специализированных языков полностью соответствовали SC-коду, т.е. были sc-текстами. Такие специализированные языки называют **sc-языками**. Каждый sc-язык определяется своим расширением множества ключевых узлов SC-кода.

SC-код — ядро **универсального открытого языка семантических сетей**, являющегося результатом интеграции всевозможных языков семантических сетей, построенных на основе SC-кода, и задаваемого:

- фиксированным алфавитом (алфавитом SC-кода);
- постоянно расширяемым (открытым) семейством ключевых узлов, в состав которого входят все ключевые узлы всех интегрируемых языков.

SC-код — **единство языка и метаязыка**. Так, например, в виде sc-конструкций можно описать синтаксис, семантику и онтологию SC-кода. С формальной точки зрения, SC-код можно трактовать как метаязык базовой семантической спецификации sc-элементов с помощью специального набора ключевых узлов SC-кода.

Единство языка и метаязыка в SC-коде проявляется и на самом низком уровне — на уровне sc-дуг принадлежности $\langle si, ei \rangle$, в которых сам sc-элемент ei , а не обозначаемый им объект, является элементом множества, обозначаемого sc-узлом si .

SC-код позволяет описать структуру любой информационной конструкции, не принадлежащей SC-коду, на любом уровне (на любом этапе синтаксического и семантического анализа). В частности, первичную синтаксическую структуру любой информационной конструкции можно представить в виде изоморфной sc-конструкции. Следовательно, SC-код может быть использован в качестве метаязыка для описания любого внешнего языка, т.е. языка, тексты которого не являются sc-конструкциями.

SC-код хорошо приспособлен к использованию в условиях **не-факторов** — нестационарности, неточности, противоречивости, неактуальности знаний, а также неполноты знаний (нечёткости, несформированности множеств, несформированности внешних информационных конструкций) [49].

Информационные конструкции SC-кода (sc-конструкции) легко **визуализируются**.

Принцип 6. Унификация структуризации баз знаний

Трактовка семантической структуры **базы знаний** интеллектуальной системы как отражения иерархической системы взаимосвязанных друг с другом

предметных областей, представляемых в базе знаний, предполагает, во-первых, уточнение понятия предметной области; во-вторых, разработку языковых средств описания структуры предметных областей с помощью унифицированных семантических сетей; в-третьих, разработку языковых средств описания типологии предметных областей и различных видов связей между ними.

Структуризация базы знаний, выделение в ней различных связанных между собой подструктур необходимы по целому ряду причин, в частности, для дидактических целей (человеку, усваивающему некоторые знания, желательно иметь своего рода оглавление или «карту» этих знаний, что позволяет планировать их усвоение и рассматривать их с различной степенью детализации), а также для организации распределения работ по проектированию баз знаний (разным исполнителям поручается разработка разных фрагментов базы знаний, имеющих достаточно чёткие границы).

Таким образом, база знаний рассматривается как система взаимосвязанных между собой интегрируемых структур — **фрагментов базы знаний**. Связи между ними могут быть самыми различными. Каждый фрагмент и вся база знаний в целом могут иметь несколько вариантов декомпозиции на подфрагменты (частные фрагменты).

По структурно-семантическому принципу можно выделить следующие типы фрагментов баз знаний:

- база фактов некоторой предметной области, которая представляет собой результат интеграции всех известных в текущий момент фактографических высказываний, являющихся истинными для указанной предметной области;
- **иерархическая система нескольких предметных областей**, которые нецелесообразно объединять (интегрировать) в одну предметную область;
- **семантическая окрестность заданного объекта**;
- **связная семантическая окрестность конечного множества заданных объектов**, представляющая сравнение (сравнительный анализ) и связи объектов из заданного конечного множества, т.е. описывающая сходства (аналогии), отличия заданных объектов, а также «близкие» связи между ними.

При структуризации базы знаний некоторым её фрагментам приписывается статус **разделов базы знаний**, которые именуются, нумеруются и входят в её состав (оглавления).

Рассмотрим формальное уточнение понятия «**предметная область**» с помощью SC-кода. Если в рамках **sc-модели базы знаний** явно вводится некоторая предметная область, то она трактуется как некоторая sc-структура, для которой в базе знаний явно вводится обозначающий её sc-узел, который, в свою очередь, связывается входящей в него sc-дугой основного вида с sc-узлом, обозначающим класс sc-структур, являющихся предметными областями. После этого в указанной sc-структуре необходимо явно задать роли некоторых **ключевых узлов**, входящих в состав этой sc-структуры. К числу таких ролей относятся:

- быть **максимальным классом исследуемых объектов**, т.е. множеством всех исследуемых объектов и только их. В каждой предметной области существует только один ключевой узел, выполняющий такую роль;
- быть **классом исследуемых объектов**. Каждая предметная область может иметь любое число таких классов;
- быть **классом вторичных объектов, построенных на основе исследуемых**;
- быть **классом вспомогательных объектов**, через связи с которыми описываются некоторые характеристики исследуемых объектов;
- быть **отношением, каждая связка которого связывает только исследуемые объекты** или вторичные объекты, построенные на основе исследуемых;
- быть **отношением, каждая связка которого связывает исследуемые объекты со вспомогательными**.

Такое явное указание ролей ключевых элементов предметных областей есть не что иное, как их семантическая спецификация, уточняющая то, *какие объекты* описываются (иссле-

дуются) в данной предметной области, и о *каких характеристиках*, связях исследуемых объектов в данной предметной области идёт речь.

Можно говорить о достаточно богатой типологии предметных областей. В частности, можно выделить следующие классы предметных областей:

- предметная область, описывающая теоретико-множественные характеристики связи заданного семейства объектов. Такие предметные области, в частности, могут быть онтологиями других предметных областей;
- терминологическая сеть заданного фрагмента базы знаний;
- текст формальной теории, описывающей свойства и закономерности заданной предметной области. Классами объектов исследования такой предметной метаобласти являются класс логических формул и, в частности, высказываний интерпретируемых на заданной предметной области; класс элементов заданной предметной области, используемых в качестве констант в указанных логических формулах; класс переменных, используемых в указанных логических формулах, возможными значениями которых являются соответствующие элементы заданной предметной области;
- логическая система понятий, описываемых в заданной формальной теории. Эта предметная метаобласть выделяет класс понятий, неопределяемых в заданной формальной теории, и связывает каждое определяемое понятие с теми понятиями, на основе которых оно определяется;
- логическая система утверждений заданной формальной теории. Эта предметная метаобласть выделяет класс аксиом для заданной формальной теории, каждой теореме ставит в соответствие одно из её доказательств и связывает каждую теорему со всеми утверждениями и определениями, которые используются в основном доказательстве этой теоремы.

Заметим, что некоторым из перечисленных классов предметных областей может соответствовать унифицированный набор используемых в них ключевых понятий. Унификация (стандартизация) таких наборов понятий — важнейшее средство более глубокой семантической совместимости (интегрируемости) различных фрагментов базы знаний. Результат такой унификации — разработка средств SC-кода, ориентированных на представление предметных областей соответствующего класса. Такие языковые средства будем называть специализированным **sc-языком**.

К числу таких специализированных sc-языков можно отнести:

- **теоретико-множественный sc-язык**, обеспечивающий описание теоретико-множественных характеристик и связей заданного семейства объектов. С помощью такого языка, в частности, могут быть представлены предметные области, являющиеся теоретико-множественными отношениями других предметных областей;
- **терминологический sc-язык**, обеспечивающий построение терминологических сетей;
- **логический sc-язык**, обеспечивающий построение sc-текстов формальных теорий.

Таким образом, **SC-код** — ядро целого семейства различных **sc-языков**, ориентированных на описание различных классов предметных областей, в каждый из которых входят предметные области с разными множествами исследуемых объектов, но с одинаковыми предметами исследования.

Каждому такому специализированному sc-языку ставится в соответствие **множество ключевых узлов**, обозначающих различные классы исследуемых объектов, различные отношения и алгебраические операции, заданные на множестве исследуемых объектов.

SC-язык, являющийся объединением всевозможных специализированных sc-языков, будем называть **языком SCK** (Semantic Code Knowledge). Этот язык рассматривается, как интегрированный язык представления знаний. Язык SCK — открытый (расширяемый), поскольку его всегда можно пополнить новым sc-языком, описывающим структуры нового вида предметных областей.

Построение **семантической структуры базы знаний** интеллектуальной системы требует не только явного представления спецификаций каждой описываемой предметной области в виде sc-текста, но и явного описания всевозможных связей между этими предметными областями.

Переходя к рассмотрению отношений, заданных на множестве предметных областей, мы фактически переходим к некоторой предметной *метаобласти*, объектами исследования которой являются всевозможные предметные области (в т.ч. и сама эта предметная метаобласть).

Обобщая понятия «гомоморфизм» и «изоморфизм» алгебраических систем, можно говорить о гомоморфизме и изоморфизме предметных областей. Это даёт основу для выявления глубоких нетривиальных аналогий между предметными областями.

Элементы одной предметной области также могут быть элементами другой. При этом возможны различные варианты такого пересечения: строгое пересечение, строгое включение. Общие элементы пересекающихся предметных областей в рамках этих областей могут играть как одинаковые, так и разные роли. Так, например, первичные элементы одной предметной области могут входить в состав другой предметной области в качестве вторичных или ключевых элементов. Объекты исследования одной предметной области могут входить в состав другой предметной области и в качестве вспомогательных элементов.

В качестве примеров отношений, заданных на множестве предметных областей, рассмотрим несколько вариантов выделения частных предметных областей:

— на основе выделения подмножества из максимального класса исследуемых объектов. Таким способом из предметной области геометрии, объектами исследования которой являются геометрические точки, фигуры и семейства фигур, можно выделить предметную область планиметрии, изучающую планарные фигуры и планарные семейства фигур (т. е. семейства фигур, лежащих в одной плоскости) и предметную область стереометрии, которая изучает непланарные фигуры и непланарные семейства фигур, которые могут состоять как из непланарных, так и из планарных фигур;

— выделение частной предметной области, на основе выделения подмножества из семейства классов исследуемых объектов. Таким способом из предметной области, изучающей треугольники, можно выделить предметную область, изучающую остроугольные, тупоугольные и прямоугольные треугольники, а также предметную область, изучающую равнобедренные, равнобедренные и равнобедренные треугольники;

— выделение частной предметной области, на основе выделения подмножества из семейства отношений, заданных на исследуемых объектах. Таким способом из геометрической предметной области можно выделить предметную область, объектами исследования которой являются геометрические фигуры, а предметом исследования — их числовые характеристики, и предметную область, объектами исследования которой являются геометрические фигуры, а предметом исследования — различные виды их конгруэнтности (движений).

Предметная область позволяет рассматривать исследуемые объекты на разных уровнях детализации, как в рамках исходной (заданной) предметной области, расширяя эту предметную область в соответствующих направлениях, так и в системе самостоятельных, но связанных между собой предметных областей.

Первый и важнейший этап проектирования базы знаний — уточнение структуры описываемой предметной области или нескольких взаимосвязанных предметных областей. Уточ-

нение такой структуры — это прежде всего уточнение класса исследуемых объектов, предмета исследования, всего семейства ключевых узлов семантической сети, представляющей предметную область. При этом для заданного класса исследуемых объектов и заданного предмета исследования можно построить более и менее качественные предметные области.

Рассмотрим ещё один тип фрагментов баз знаний — **семантические окрестности**. В общем случае семантическая окрестность заданного объекта — описание некоторых примет, числовых характеристик, свойств и связей заданного объекта. Частными видами семантических окрестностей являются:

- описание характеристик, свойств или связей заданного объекта, *однозначно* определяющих (устанавливающих) заданный объект (для понятий — это определение или любое другое высказывание определяющего типа);
- полная (интегрированная) семантическая окрестность заданного объекта, содержащая все известные в *текущий момент* сведения об этом объекте в рамках заданного раздела базы знаний или в рамках всего текущего состояния базы знаний;
- описание числовых характеристик (параметров, признаков) заданного объекта;
- описание свойств заданного объекта (истинные, не фактографические высказывания, в которых знак этого объекта используется в качестве константы);
- описание теоретико-множественных связей заданного объекта с другими объектами;
- описание разноязычных терминов, иероглифов, пиктограмм, используемых для внешней идентификации заданного объекта (в т. ч. и описание происхождения этих идентификаторов);
- перечень отношений, соответствующих заданному объекту (отношений, в области определения которых заданный объект входит в качестве элемента, подмножества, надмножества, отношений, область определения каждого из которых строго пересекается с заданным объектом).

В **sc-модели базы знаний** каждая семантическая окрестность представляется в виде соответствующего sc-текста. Для явного введения (задания) этой семантической окрестности в рамках sc-модели базы знаний необходимо:

- 1) явно ввести sc-узел, обозначающий эту семантическую окрестность;
- 2) явно связать введённый sc-узел sc-дугами основного вида со всеми элементами обозначаемого им sc-текста;
- 3) явно указать центральный элемент семантической окрестности с помощью ролевого отношения **быть центром семантической окрестности**;
- 4) явно связать введённый sc-узел sc-дугой основного вида со специальным sc-узлом, обозначающим **класс sc-структур, являющихся семантическими окрестностями**.

После этого можно описывать различные характеристики (в частности, типологию) явно введённой семантической окрестности, а также её различные связи с другими фрагментами базы знаний.

Принцип 7. ГРАФОВЫЕ ЯЗЫКИ ПРОГРАММИРОВАНИЯ

Графовые языки программирования используются для описания способов решения задач и поведения агентов над общей графодинамической памятью. Они ориентированы на обработку унифицированных семантических сетей. Их программы — унифицированные семантические сети.

Если все используемые в интеллектуальной системе графовые языки программирования привести к общему унифицированному стандарту — SC-коду, то можно достаточно эффективно решать проблему формализации семантической совместимости программ, написанных не только на одном, но и на разных языках программирования.

В традиционных языках программирования синтаксическая структура и семантика хранящихся в памяти обрабатываемых данных отдаются на откуп программисту. Поэтому о семантической совместимости программ не может идти и речи.

Наряду с применением в интеллектуальных системах целого множества sc-языков различного назначения, востребованным является использование целого семейства *совместимых* sc-языков программирования, которые могут иметь разный уровень, быть последовательными, процедурными и декларативными.

Важнейшая особенность всех этих языков — использование ассоциативного доступа к обрабатываемым фрагментам хранимого в графодинамической памяти sc-текста.

Операционная семантика каждого такого графового языка программирования (точнее, sc-языка программирования) задаётся коллективом агентов над общей графодинамической памятью, обеспечивающих интерпретацию любой программы указанного языка программирования, хранящейся вместе с обрабатываемой информацией в указанной графодинамической памяти.

Программы, представленные в виде семантической сети и описывающие обработку семантических сетей, а также соответствующие им языки программирования, фактически, открывают новую страницу в теории программирования, которую можно назвать семантической теорией программ и языков программирования. Основным лейтмотивом такой теории должно быть обеспечение семантической совместимости программ и языков программирования.

Принцип 8. Унификация формального описания агентов, работающих над семантической памятью

Выделение базового sc-языка программирования предназначено для унификации формального описания поведения агентов, работающих над общей графодинамической памятью. Такой базовый sc-язык программирования будем называть **языком SCP** (Semantic Code Programming), а написанные на нём программы — **scp-программы**.

Принцип 9. Унификация семантических моделей обработки знаний

На основе унифицированных семантических сетей (sc-текстов) уточнить понятие «**Унифицированная модель обработки информации**», а также понятие «Унифицированная модель решения задач».

Все указанные абстрактные модели будем называть **sc-моделями обработки знаний** или **sc-машинами**, поскольку в их основе лежит использование SC-кода. Каждая такая модель (sc-машина) представляет собой многоагентную систему, состоящую из **sc-памяти** — графодинамической памяти, в которой хранятся и обрабатываются тексты SC-кода, и **sc-агентов** — коллектива агентов, работающих над общей для них sc-памятью и взаимодействующих между собой только через неё.

Очевидно, что sc-модели обработки знаний являются частным, унифицированным видом графодинамических моделей параллельной асинхронной обработки информации.

Каждый sc-агент реагирует на соответствующий ему класс ситуаций и/или событий, происходящих в sc-памяти, и осуществляет определённое преобразование sc-текста, находящегося в семантической окрестности обрабатываемой ситуации и/или события. Типология sc-агентов достаточно богата:

- sc-агенты, обеспечивающие интерпретацию программ различных sc-языков программирования высокого уровня;
- sc-агенты информационного поиска;
- sc-агенты, обеспечивающие реализацию правил логического вывода, соответствующих различным логическим исчислениям;
- sc-агенты сведения задач к подзадачам;
- sc-агенты анализа качества хранимой базы знаний, в частности, её корректности, полноты;
- sc-агенты обнаружения автоматического склеивания синонимичных sc-элементов;
- sc-агенты автоматического устранения некоторых ошибок в базе знаний;
- sc-агенты удаления информационного мусора (в частности, удаления фрагментов базы знаний, которые редко востребованы и могут быть достаточно легко восстановлены в случае их отсутствия);
- sc-агенты, обеспечивающие трансляцию вводимой информации с различных внешних языков в SC-код;
- sc-агенты, обеспечивающие трансляцию sc-текстов, вводимых пользователю на различные внешние языки;
- рецепторные sc-агенты;
- эффекторные sc-агенты.

В понятии sc-машины набор агентов не задаётся, т.е. могут существовать разные sc-машины с разным набором sc-агентов. Несколько разных sc-машин можно **интегрировать**. С формальной точки зрения это сделать несложно. Во-первых, интегрировать sc-текст, описывающий текущее состояние взаимодействия sc-агентов одной sc-машины, с аналогичным sc-текстом другой sc-машины. Во-вторых, полученный интегрированный sc-текст поместить в sc-память интегрированной sc-машины. В-третьих, включить в интегрированную sc-машину все sc-агенты первой и второй интегрируемых sc-машин.

Более того, одна sc-машина может **интерпретировать** другую. Для этого необходима разработка целого семейства sc-языков программирования различного уровня. Тексты (программы) всех этих языков должны храниться в sc-памяти, т.е. должны быть семантическими сетями, представленными в SC-коде. Операционная семантика (интерпретация) каждого из этих языков задаётся определённым набором sc-агентов, процедура выполнения (поведения) каждой из которых описывается программой, написанной на языке *более низкого уровня*.

В абстрактных sc-машинах можно выделить следующие языки программирования.

- Семейство sc-языков программирования высокого и сверхвысокого уровня (как процедурные, так и непроцедурные). Тексты программ этих языков хранятся в базе знаний интеллектуальной системы и описывают способы решения различных классов задач в соответствующих предметных областях.
- Базовый sc-язык программирования (язык SCP), на котором описываются sc-агенты и интерпретации sc-языков программирования высокого и сверхвысокого уровня, а также sc-операции, обеспечивающие интерпретацию различных логических исчислений, различных моделей интеллектуального решения задач.
- SC-язык программирования, на котором описываются sc-агенты интерпретации базового sc-языка программирования. Фактически, это первый язык микропрограмм для **sc-компьютера**, обеспечивающего аппаратную интерпретацию языка SCP.

— При необходимости можно ввести второй язык микропрограммирования, описывающий интерпретацию первого и т.д.

На основе понятия абстрактной *sc*-машины можно уточнить понятие «Унифицированная логико-семантическая модель интеллектуальной системы». Такую унифицированную модель интеллектуальной системы будем называть абстрактной **sc-моделью** интеллектуальной системы или, сокращённо, **sc-системой**. Абстрактная *sc*-система включает в себя **sc-модель базы знаний** — интегрированную базу *всех* знаний, которые необходимы для функционирования интеллектуальной системы и представлены в виде интегрированного *sc*-текста, и абстрактную **sc-машину**, в памяти которой хранится *sc*-текст базы знаний абстрактной системы.

Нетрудно заметить, что *sc*-текст базы знаний интеллектуальной системы является формальным и унифицированным уточнением того, что должна знать проектируемая интеллектуальная система. А *sc*-машина интеллектуальной системы и, в первую очередь, набор входящих в её состав *sc*-агентов, является формальным и унифицированным уточнением того, что должна эта система уметь делать со своими знаниями.

Подчеркнём, что чёткое выделение абстрактного семантического уровня интеллектуальной системы позволяет не только обеспечивать их семантическую совместимость, но и сформировать критерии сравнения интеллектуальных систем по уровню их возможностей. Очевидно, что уровень возможностей интеллектуальной системы определяется качеством (корректностью, полнотой, многообразием) её *знаний* и эффективностью её *умений* (т.е. эффективностью используемых её моделей решения задач).

Заметим, что абстрактную логико-семантическую модель, в принципе, можно построить для любой компьютерной системы, обеспечивая их семантическую совместимость на абстрактном логико-семантическом уровне.

Принцип 10. Унификация семантических моделей информационного поиска

На основе унифицированных семантических сетей (т.е. на основе *SC*-кода) обеспечить построение **унифицированных семантических моделей информационного поиска** — унифицированных семантических моделей ассоциативного доступа к информации, хранимой в семантической памяти.

Ассоциативный доступ — это доступ, основанный не на знании того, где находится искомая информация (в частности, на знании адреса или имени соответствующей области памяти), а на знании того, как искомая информация связана с известной информацией, хранимой в памяти.

Эффективность организации информационного поиска в базе знаний интеллектуальной системы во многом определяет эффективность самой интеллектуальной системы. Это обусловлено тем, что время, затрачиваемое интеллектуальной системой на поиск нужных в текущий момент знаний и навыков, занимает, мягко говоря, не меньше половины времени, затрачиваемого на решение задачи в целом.

Унифицированная семантическая модель информационного поиска (**sc-модель информационного поиска**) включает в себя, во-первых, **SC-язык вопросов**, с помощью которого в виде *sc*-текстов осуществляется описание (спецификация) запрашиваемых (искомых) фрагментов всего интегрированного *sc*-текста, хранимого в текущий момент в *sc*-памяти (т.е. *sc*-текста, который является *sc*-моделью базы знаний). Тексты, принадлежащие *SC*-языку вопросов, будем называть **sc-вопросами**. Во-вторых, **SC-язык оформления ответов**, с помощью которого осуществляется явное выделение *sc*-текстов, являющихся ответами, и явное описание их связи с явно выделенными *sc*-текстами, которые представляют вопросы, соответствующие указанным ответам. В-третьих, семейство информационно-поисковых *sc*-агентов, каждый из которых реагирует на соответствующий ему

тип инициированного sc-вопроса и выполняет соответствующую поисковую процедуру в sc-памяти.

Семантическая типология вопросов — предмет отдельного рассмотрения. Приведём фрагмент такой типологии, чтобы проиллюстрировать семантическую мощьность sc-языка вопросов.

Прежде всего, по аналогии с логическими формулами множество вопросов разбивается на **атомарные вопросы** и **неатомарные вопросы**, каждый из которых представляет собой конечное множество вопросов.

Компонентами неатомарного вопроса могут быть как атомарные, так и неатомарные вопросы. При этом, если построить оргграф, вершинами которого будут знаки всех вопросов, входящих в состав заданного неатомарного вопроса, а дуги будут связывать знаки неатомарных вопросов, входящих в состав заданного неатомарного вопроса, с их компонентами, то этот оргграф будет деревом, все конечные вершины которого являются знаками атомарных вопросов. Частным видом неатомарного вопроса является конъюнктивный вопрос, ответом на который является конъюнкция (интеграция) ответов на все вопросы, являющиеся компонентами этого конъюнктивного вопроса.

Поскольку в общем случае вопросу может соответствовать *несколько* правильных ответов (т.е. ответов, удовлетворяющих, релевантных, соответствующих заданному вопросу), множество вопросов разбивается на вопросы, запрашивающие:

- все правильные ответы;
- один (или, по крайней мере, один) правильный ответ;
- вопросы, запрашивающие несколько разнообразных правильных ответов;
- вопросы, запрашивающие точно указанное число правильных ответов.

Специальным видом неатомарных вопросов являются **сколько-вопросы**, запрашивающие не сами правильные ответы некоторого вопроса (который может быть атомарным и единственным компонентом сколько-вопроса), а количество правильных ответов.

Типы атомарных вопросов

Какой-вопрос атомарного вида. Каждый такой вопрос запрашивает фрагменты базы знаний, *изоморфные* заданному **образцу**, который может иметь *произвольный* размер, произвольную конфигурацию и может быть представлен не только *логической формулой существования*, в которой квантор существования действует на конъюнкцию *атомарных логических формул*, но также и логической формулой существования, в которой квантор существования действует на логическую формулу *произвольного вида*. Суть атомарного *какой-вопроса* заключается в поиске знаков таких объектов, которые заданным образом связаны с другими известными и неизвестными (искомыми) объектами, т.е. в поиске знаков таких объектов, которые удовлетворяют заданным требованиям. На основе *какой-вопросов* атомарного вида строится важный класс неатомарных вопросов, ключевыми компонентами которых являются атомарные *какой-вопросы*, а остальными компонентами — вопросы любого вида, в формулировках которых используются переменные, входящие в состав соответствующих ключевых *какой-вопросов*.

Запрос всех элементов заданного конечного множества (чаще всего это множество из элементов некоторой структуры);

Запрос внешней информационной конструкции, представленной некоторым файлом в том или ином формате.

Запрос полного текста заданного высказывания.

Ли-вопрос, запрашивающий факт истинности или ложности заданного высказывания в рамках заданной формальной теории.

Вопрос выбора альтернатив, запрашивающий одно или несколько истинных высказываний из заданного множества высказываний.

Почему-вопрос, запрашивающий обоснование (доказательство) истинности заданного высказывания.

Что-это-вопрос, запрашивающий основные сведения об указываемом объекте. Фактически, речь идёт о выделении из базы знаний семантической окрестности, «центром» которой является знак указываемого объекта. Таким объектом может быть все, что угодно — понятие, предметная область, формальная теория, высказывание, любая структура, материальный объект.

Запрос общих свойств объектов, принадлежащих заданному классу.

Запрос идентифицирующих признаков заданного объекта. Здесь запрашиваются фрагменты базы знаний, каждый из которых *однозначно* определяет (устанавливает, идентифицирует) заданный объект. Если заданным объектом является понятие, то таким идентифицирующим признаком является либо определение этого понятия, либо соответствующая теорема о необходимости и достаточности.

Запрос связей между заданными объектами.

Запрос сравнительного анализа заданных объектов.

Запрос сходств заданных объектов (сходства, аналогии — частный вид связей между объектами).

Запрос отличий заданных объектов (отличия объектов — частный вид связей между ними).

Запрос плана решения заданной конкретной задачи, т.е. плана достижения заданной цели в заданных конкретных условиях.

Запрос обобщённого способа решения любой задачи из заданного класса задач. Таким обобщённым способом может быть алгоритм, декларативная (непроцедурная) программа, нестрогое предписание (рекомендация).

Зачем-вопрос, запрашивающий то, какой надцели соответствует заданная цель, которая может быть сформулирована как в декларативной, так и в процедурной форме.

Список типов атомарных вопросов можно продолжить, но почти все они будут подтипами (подмножествами) перечисленных типов вопросов. В основе sc-языка вопросов лежит построение онтологии вопросов, в рамках которой чётко прописываются все теоретико-множественные (в первую очередь, родо-видовые) связи между *всеми* выделенными типами и подтипами вопросов. При этом в формулировке *каждого* конкретного sc-вопроса *явным образом* отражаются иерархия *всех* типов вопросов, которым принадлежит данный конкретный sc-вопрос. Для этого каждому типу вопросов ставится в соответствие ключевой sc-узел, обозначающий этот тип вопросов.

В заключение заметим, что в **SC-языке оформления ответов** кроме отношения релевантности, связывающего вопрос с правильными на него ответами, используются языковые средства, описывающие качество, полноту ответов. Это вызвано тем, что некоторые типы вопросов предполагают наличие целого множества правильных ответов, но разного качества.

Принцип 11. УНИФИКАЦИЯ СЕМАНТИЧЕСКИХ МОДЕЛЕЙ ИНТЕГРАЦИИ ЗНАНИЙ И СЕМАНТИЧЕСКИХ МОДЕЛЕЙ ИНТЕГРАЦИИ ЦЕЛЫХ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ

На основе унифицированных семантических сетей обеспечить построение **унифицированных семантических моделей интеграции знаний** (понимания знаний) и использовать эти модели, во-первых, как основу процесса приобретения интеллектуальной системой новых знаний как со стороны конечных пользователей, так и со стороны разработчиков; во-вторых, как основу интеграции программ и различных семантических моделей расширения задач; в-третьих, как основу интеграции абстрактных логико-семантических моделей интеллектуальных систем.

Главное свойство интеллектуальной системы — не те интеллектуальные знания и навыки, способности, которые она имеет в текущий момент, а *мета-способность* приобретать *любые* необходимые ей *новые* знания и навыки. А для этого интеллектуальная система, как минимум, должна уметь интегрировать эти приобретаемые знания и навыки. Следовательно, проблема формализации интеграции знаний и навыков — центральная для деятельности интеллектуальных систем.

Принципиальные отличия интеграции двух интеллектуальных систем заключается в следующем: пусть имеются две интеллектуальных системы $s1$ и $s2$, первая из которых способна решать задачи из множества $q1$, вторая — $q2$. В результате простого соединения этих систем получаем систему, которая способна решать задачи из множества $(q1 \cup q2)$. Тогда как в результате интеграции мы получаем систему, которая способна решать значительно большее число задач, чем $(q1 \cup q2)$. Таким образом, при интеграции интеллектуальных систем происходит приобретение нового качества «на стыке» интегрируемых систем, когда для решения некоторых задач одна часть необходимых знаний и/или умений находится в одной интегрируемой системе, а другая часть — в другой системе.

Процесс интеграции двух семантических сетей рассмотрим как систему следующих взаимодействующих подпроцессов, некоторые из которых могут выполняться параллельно:

- приведение интегрируемых семантических сетей к унифицированному виду, т.е. представление (запись) их в SC-коде;
- согласование ключевых узлов и онтологий, используемых в интегрируемых sc-текстах. Очевидно, что полностью автоматизировать такое согласование невозможно, поэтому разработчикам интегрируемых фрагментов баз знаний и целых баз знаний необходимо *уметь договариваться* друг с другом;
- выделение в интегрируемых sc-текстах таких sc-элементов, которые имеют глобальные (уникальные) идентификаторы (внешние имена);
- выделение в интегрируемых sc-текстах sc-элементов, имеющих локальные идентификаторы вместе с областью действия каждого такого идентификатора. Область действия локального идентификатора — такой фрагмент базы знаний, в рамках которого разные sc-элементы считаются синонимичными;
- склеивание sc-элементов, имеющих одинаковые глобальные идентификаторы;
- склеивание sc-элементов, имеющих одинаковые локальные идентификаторы, если каждый из этих sc-элементов принадлежит области действия своего и другого локальных идентификаторов;
- склеивание sc-элементов на основании однозначности используемых алгебраических операций;

- склеивание sc-элементов на основании логических высказываний о существовании единственности;
- склеивание кратных связей, принадлежащих отношениям:
- не имеющих кратных связей;
- имеющих кратные связи, но не для заданных типов компонентов (например, кратные связи принадлежности не могут выходить из знаков канторовских множеств).

Таким образом, интеграция семантических сетей, т.е. процесс погружения (понимания) одной семантической сети в другую — нетривиальный процесс рассуждений, направленный на выявление пар синонимичных элементов семантической сети на основе определённых знаний, имеющихся в базе знаний интеллектуальной системы.

От унифицированной семантической модели интеграции знаний (точнее, sc-текстов) можно достаточно легко перейти к интеграции sc-моделей интеллектуальных систем, поскольку после интеграции sc-моделей баз знаний интегрируемых интеллектуальных систем интеграция соответствующих им наборов sc-агентов сводится к простому теоретико-множественному объединению указанных множеств sc-агентов.

Принцип 12. Унификация и интеграция различных семантических моделей решения задач

Обеспечить в рамках проектируемой интеллектуальной системы использование не только различных видов знаний, но и **различных моделей и стратегий решения задач**.

Для этого необходимо акцентировать внимание не столько на разработке новых моделей решения задач, сколько на унификации и интеграции в рамках проектируемых интеллектуальных систем уже разработанных и хорошо зарекомендовавших себя моделей (дедуктивных, индуктивных, абдуктивных, чётких, нечётких, универсальных, специализированных). Подчеркнём то, что в разных проектируемых интеллектуальных системах могут быть востребованы самые разные сочетания известных моделей и стратегий решения задач. Подавляющее число моделей представления знаний и решения задач не являются альтернативными и *дополняют друг друга*. Не составляют исключение и такие классы моделей, как фреймовые, логические, продукционные.

Рассмотренное выше понятие **вопроса** и его формализация являются основой не только для информационно-поисковых моделей, но и для различных моделей решения задач. С точки зрения решателя задач, вопрос — **непроцедурная формулировка информационной цели**, т.е. декларативная формулировка некоторой информационной цели, которая описывает спецификацию (свойства) той информации, которую *требуется* либо найти, если она уже присутствует в текущем состоянии базы знаний, либо построить (сгенерировать, вывести), если она отсутствует в текущем состоянии памяти. Таким образом, вопрос можно считать описанием целевого (требуемого) состояния обрабатываемой базы знаний (а точнее, определённого фрагмента этой базы знаний). Вопрос также можно считать одним из видов *метазнаний*, описывающих (специфицирующих) наше *незнание*, т.е. наше знание о том, что мы не знаем, но хотели бы знать.

Вопросы могут инициироваться (задаваться) как пользователями, так и самой системой. Это означает, что в процессе обработки информации интеллектуальная система сама себе может задавать вопросы. Если инициирован некоторый sc-вопрос, то сначала активизируются соответствующие агенты информационного поиска в «надежде» на то, что запрашиваемый ответ (или ответы) на указанный sc-вопрос уже присутствует в текущем состоянии базы знаний. И только после того, как информационно-поисковые sc-агенты обнаружат отсутствие ответа в текущем состоянии базы знаний, начинается работа решателя задач, направленная на генерацию (построение, порождение, вывод) требуемого ответа.

Кроме вопроса используется также и **процедурная формулировка информационной цели** — описание (спецификация) некоторого действия, которое требуется выполнить

и которое направлено на преобразование базы знаний, хранимой в некоторой памяти. Указанное действие, выполняемое одним или несколькими sc-агентами, порождает определённое событие (изменение состояния sc-памяти).

Для унификации различных моделей решения задач необходимо уточнить не только понятие **информационной цели**, но и понятие **информационной задачи**. Информационная задача задаётся, во-первых, формулировкой информационной цели (*что требуется*), и, во-вторых, той хранимой в памяти информацией, которая семантически связана с заданной информационной целью (*что дано*). В пределе, контекстом информационной цели можно считать текущее состояние всей хранимой базы знаний.

Формальное рассмотрение контекстов различных информационных задач требует разработки специальных языковых средств, предназначенных для описания *текущего состояния хранимой* базы знаний, а точнее, для описания «границ» между тем, что в текущем состоянии базы знаний известно, и тем, что неизвестно. К числу таких языковых средств, в частности, относятся следующие ключевые узлы, являющиеся знаками нестационарных множеств (т.е. множеств, которые в разные моменты времени могут иметь разные элементы):

- быть sc-дугой нечёткой принадлежности (такая sc-дуга связывает sc-узел, обозначающий некоторое множество, с sc-элементом, о котором в текущий момент времени неизвестно, принадлежит он указанному множеству или нет);
- быть построенным конечным множеством (у каждого такого множества в текущем состоянии базы знаний известны и явно указаны *все* его элементы);
- быть построенным высказыванием (для каждого такого высказывания в текущем состоянии базы знаний представлен не только его знак, но и полный текст);
- быть построенной внешней информационной конструкцией (файлом);
- быть аксиоматизированной формальной теорией;
- быть построенным рассуждением (обоснованием, доказательством, решением);
- быть построенной программой.

Более подробно унифицированные семантические модели решения задач и технология их проектирования рассмотрены в работе [26].

Принцип 13. УНИФИКАЦИЯ ВИЗУАЛИЗАЦИИ СЕМАНТИЧЕСКИХ СЕТЕЙ

Использовать язык унифицированного визуального представления абстрактных унифицированных семантических сетей в виде, близком к изоморфному в качестве основы организации графического пользовательского интерфейса.

Указанный язык графического изображения sc-текстов назван SCg-кодом (Semantic Code graphical). Подчёркнём, что следует чётко отличать язык абстрактных унифицированных семантических сетей (SC-код), который абстрагируется от того, как должны быть физически представлены узлы и коннекторы текстов этого языка (sc-текстов), и язык графического изображения таких семантических сетей. Т.е. абстрактная семантическая сеть и её рисунок — принципиально разные вещи.

С помощью SCg-кода осуществляется отображение на экране не только пользовательских сообщений, адресуемых системе, и сообщений, адресуемых пользователю, но и всей остальной информации, необходимой для организации работы пользователя (прежде всего, элементы управления интерфейсом). Такая унификация отображаемой пользователю информации даёт возможность организовать взаимодействие пользователя с help-системой точно так же, как и его взаимодействие с основной (предметной) системой.

Трактовка элементов управления пользовательским интерфейсом как элементов отображаемого на экране SCg-текста позволяет унифицировать представления *любой* информации, отображаемой на экране, и способы инициирования различных вопросов, касающихся *любой* отображаемой на экране информации (в том числе, и элементов управления).

Для того, чтобы чётко отделить те средства SCg-кода, которые обусловлены самим SC-кодом, от тех средств, которые обусловлены стремлением повысить наглядность SCg-текстов, введём ядро SCg-кода (или просто SCg-ядро), алфавит которого взаимно однозначно соответствует алфавиту SC-кода и, соответственно, тексты которого *изоморфны* семантически эквивалентным текстам SC-кода.

Переход от SCg-ядра к SCg-коду заключается в ослаблении требований, предъявляемых к изображениям семантических сетей, в целях обеспечения удобства для человеческого восприятия. Такое ослабление осуществляется в следующих направлениях: вводится приписывание идентификаторов изображаемых sc-элементов, расширяется алфавит графических примитивов, допускается уникальное изображение некоторых sc-узлов, допускается синонимия sc.g-элементов, но при этом синонимичным элементам должны быть приписаны одинаковые идентификаторы, вводятся специальные графические средства, направленные на повышение наглядности (шинные линии, контуры).

Заметим также, что кроме SCg-кода для внешнего представления абстрактных унифицированных семантических сетей используются также и другие языки:

- SCs-код, обеспечивающий представление унифицированных абстрактных семантических сетей (sc-текстов) в виде, близком к традиционным текстам;
- SCn-код, обеспечивающий гипертекстовое представление абстрактных sc-текстов, предназначенное для оформления исходных текстов баз знаний.

Более подробно различные языки внешнего представления абстрактных sc-текстов вместе с большим количеством примеров рассмотрены в работах [12 — 14].

Принцип 14. УНИФИКАЦИЯ СЕМАНТИЧЕСКИХ МОДЕЛЕЙ РАЗЛИЧНЫХ ПОЛЬЗОВАТЕЛЬСКИХ ИНТЕРФЕЙСОВ

Пользовательский интерфейс интеллектуальной системы, построенной на основе предлагаемой технологии, рассматривать как **специализированную интеллектуальную систему**, предназначенную для **трансляции адресуемых пользователю сообщений** с внутреннего абстрактного семантического языка представления знаний (SC-кода) на тот или иной внешний язык, тексты которого отображаются пользователю в удобном для него виде, а также для **трансляции пользовательских сообщений** с внешнего языка на внутренний семантический язык интеллектуальной системы (SC-код).

Трактовка пользовательских интерфейсов как интеллектуальных систем и унификация семантических моделей таких систем дают возможность, во-первых, унифицировать проектирование пользовательских интерфейсов. Во-вторых, легко наращивать возможности пользовательских интерфейсов. В-третьих, легко интегрировать пользовательские интерфейсы с предметными (основными) интеллектуальными системами. И наконец, неограниченно использовать базу знаний предметных интеллектуальных



систем для семантического анализа и понимания вводимой пользователем информации (в частности, естественно-языковых текстов).

Более подробно унифицированные семантические модели пользовательских интерфейсов и технология их проектирования рассмотрены в работе [34].

Принцип 15. Библиотека типовых семантически совместимых компонентов интеллектуальных систем и методика модульного проектирования интеллектуальных систем

Создать общую библиотеку многократно используемых семантически совместимых компонентов интеллектуальных систем, на основе которой разработать методику модульного (компонентного, сборочного) проектирования интеллектуальных систем, в целях ускорения процесса проектирования интеллектуальных систем.

В указанной библиотеке можно выделить следующие разделы (частные библиотеки):

- библиотека многократно используемых компонентов баз знаний. Прежде всего, в эту библиотеку входят самые различные по содержанию, но семантически совместимые онтологии. Кроме того, сюда попадают различные «джентльменские наборы» знаний, которыми должны владеть «образованные» интеллектуальные системы. К таким знаниям, в частности, относятся базовые знания по арифметике, теории множеств, теории отношений, логике и многие другие знания, востребованность которых может быть самой разной;
- библиотека компонентов семантических моделей информационного поиска. Сюда, прежде всего, попадают различные информационно-поисковые агенты;
- библиотека компонентов семантических моделей интеграции знаний и машин обработки знаний;
- библиотека интерпретаторов программ, соответствующих различным языкам программирования;
- библиотека различных стратегий решения задач, различных моделей решения задач и агентов, входящих в состав таких моделей;
- библиотека компонентов пользовательских интерфейсов.

Все компоненты, включаемые в состав общей библиотеки компонентов интеллектуальных систем, оформляются как компоненты интеллектуальной собственности (intellectual property), поэтому будем их также называть ip-компонентами.

Особо подчеркнём, что это модульное проектирование интеллектуальных систем возможно только в том случае, если отбор компонентов, включаемых в состав рассмотренной библиотеки, будет осуществляться на основе тщательного анализа качества этих компонентов. Одним из важнейших критериев такого анализа является семантическая совместимость анализируемых компонентов со всеми компонентами, имеющимися в текущей версии библиотеки.

Для обеспечения семантической совместимости таких компонентов интеллектуальных систем, которые являются унифицированными семантическими моделями (sc-моделями знаний, sc-моделями машин обработки знаний, sc-агентов, sc-моделями интеллектуальных подсистем), необходимо согласовать семантику (смысл) всех используемых ключевых узлов и глобальные идентификаторы ключевых узлов, используемых в разных компонентах.

После этого интеграция всех компонентов, входящих в состав библиотеки, и в любых комбинациях осуществляется автоматически, без вмешательства разработчика.

Принцип 16. ПЛАТФОРМЕННО-НЕЗАВИСИМЫЙ ХАРАКТЕР ПРОЕКТИРОВАНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ

Для максимальной платформенной независимости технологии обеспечить чёткое разделение процесса проектирования формального описания логико-семантической модели разрабатываемой интеллектуальной системы от процесса реализации (интерпретации) этой модели на той или иной платформе.

Подчеркнём при этом следующее. Если каждой интеллектуальной системе соответствует своя уникальная логико-семантическая модель, то каждый интерпретатор абстрактных логико-семантических моделей интеллектуальных систем должен обеспечивать интерпретацию целого класса таких моделей, а в идеале — интерпретацию любой такой модели. Следовательно, разработка указанных интерпретаторов может осуществляться абсолютно независимо от разработки логико-семантических моделей конкретных интеллектуальных систем.

Таким образом, SC-код, обеспечивающий унификацию семантического представления любых знаний, вместе с языком SCP, обеспечивающим унификацию формального описания агентов, работающих над семантической памятью, являясь средством унификации логико-семантических моделей интеллектуальных систем, выполняют в рамках предлагаемой технологии роль, аналогичную той, которую выполняет язык VHDL в современных микропроцессорных технологиях. В лице SC-кода и языка SCP мы имеем *стандарт полного* формального описания логико-семантических моделей интеллектуальных систем, обеспечивающий независимость проектирования абстрактных логико-семантических моделей конкретных интеллектуальных систем от разработки различных вариантов реализации (различных вариантов их интерпретации на различных платформах). Такой стандарт является своего рода «водоразделом» между полным платформенно-независимым описанием интеллектуальной системы (абстрактной логико-семантической моделью) и платформенно-зависимой реализацией (интерпретацией) этой абстрактной модели.

Полностью построенная абстрактная логико-семантическая модель проектируемой интеллектуальной системы:

- является открытой, поскольку её можно легко пополнять новыми знаниями и навыками, интегрируя их в текущую версию модели;
- концентрирует внимание на семантических аспектах функционирования интеллектуальной системы и не содержит никаких лишних деталей, обусловленных способами её технической реализации (интерпретации);
- является абстрактным инвариантом целого множества различных способов её технологической реализации (в том числе, и с помощью принципиально новых компьютеров).

Разработка прототипа интеллектуальной системы завершается разработкой полной sc-модели этой системы, которая записывается в виде исходного текста с использованием таких языковых средств, как SCg-код, SCs-код, SCn-код. После этого разработчик выбирает один из *универсальных* вариантов интерпретации (реализации) sc-моделей, загружает разработанные им исходные тексты в выбранный интерпретатор и получает прототип, пригодный для опытной эксплуатации и последующего совершенствования.

Если же после этого разработчиков интеллектуальной системы что-то не устраивает в выбранном варианте интерпретации sc-моделей (в частности, производительность), должна существовать достаточно продуманная методика совершенствования выбран-

ного варианта интерпретатора sc-моделей интеллектуальных систем. Очевидно, что для каждого такого варианта указанная методика будет иметь свои особенности.

Следовательно, нижние уровни детализации проектируемых интеллектуальных систем, в отличие от верхнего (логико-семантического), являются платформенно-зависимыми. Следовательно, можно говорить о различных модификациях технологии проектирования интеллектуальных систем, соответствующих разным платформам. Напомним при этом, что основная трудоёмкость проектирования интеллектуальных систем, полностью определяющая уровень её возможностей (уровень знаний и навыков), концентрируется именно на первом этапе проектирования — на разработке её абстрактной логико-семантической модели.

Таким образом, проектирование интеллектуальной системы можно организовать как два следующих самостоятельных процесса, выполняемых одновременно и независимо друг от друга:

- 1) процесс разработки абстрактной унифицированной логико-семантической модели проектируемой интеллектуальной системы;
- 2) процесс совершенствования выбранного интерпретатора абстрактных унифицированных логико-семантических моделей интеллектуальных систем.

Заметим, что сама идея обеспечения кросс-платформенной разработки компьютерных систем путём внедрения формального языка, обеспечивающего описание абстрактных (логических) моделей этих систем, не нова. Существует целый ряд кросс-платформенных технологий. Вопрос в том, о каком классе разрабатываемых компьютерных систем идёт речь; какими свойствами обладают используемые абстрактные модели компьютерных систем; какими достоинствами обладает технология разработки самих этих абстрактных моделей.

Вопросы программной реализации и, в частности, web-ориентированной реализации унифицированных логико-семантических моделей интеллектуальных систем рассмотрены в работе [32].

Принцип 17. СЕМАНТИЧЕСКИЙ АССОЦИАТИВНЫЙ ПАРАЛЛЕЛЬНЫЙ КОМПЬЮТЕР

Обеспечить возможность реализации унифицированных логико-семантических моделей интеллектуальных систем на **семантических ассоциативных компьютерах**, специально ориентированных на аппаратную реализацию таких моделей.

Очевидно, что для указанных компьютеров базовый графовый язык программирования (язык SCP) является их ассемблером, т. е. аппаратно интерпретируемым языком программирования.

В связи с проблемой создания компьютеров, ориентированных на обработку знаний, необходимо отметить следующее:

- 1) в таких компьютерах принципиально важна поддержка именно параллельной обработки знаний;
- 2) опыт использования параллельных компьютеров показывает, что эффективное их использование, предполагающее разработку качественных параллельных программ, требует особой профессиональной подготовки и высокой квалификации. Мир параллельного программирования требует особой культуры, стиля мышления. Ещё более серьёзная профессиональ-

ная подготовка требуется для разработки параллельных программ, ориентированных на обработку знаний и использующих ассоциативный доступ к обрабатываемой информации;

3) уровень развития микроэлектронных технологий в настоящее время позволяет достаточно быстро реализовывать самые смелые компьютерные архитектуры и модели обработки информации;

4) созданию параллельных компьютеров для обработки знаний должно предшествовать создание **технологии** проектирования интеллектуальных систем, в основе которой лежат те модели параллельной обработки знаний, которые будут аппаратно поддерживаться в указанных компьютерах. Иначе мы получим «грудю» талантливо сделанного «железа», эффективность использования которого будет весьма низкой;

5) предлагаемая технология проектирования интеллектуальных систем как раз и предполагает последовательное выполнение следующих этапов:

- разработка технологии проектирования абстрактных унифицированных логико-семантических моделей интеллектуальных систем;
- разработка нескольких вариантов программной реализации абстрактных унифицированных логико-семантических моделей интеллектуальных систем, выполненных на современных компьютерах;
- разработка и эксплуатация достаточно большого количества прикладных интеллектуальных систем, совершенствование технологии проектирования интеллектуальных систем на основе приобретённого опыта;
- разработка семантического ассоциативного компьютера, появление которого не отменит ранее созданного. Просто появится ещё один, но уже аппаратный вариант реализации абстрактных унифицированных логико-семантических моделей интеллектуальных систем. Его применение для разработанных абстрактных унифицированных логико-семантических моделей различных прикладных систем для их конечных пользователей абсолютно ничего не изменит, кроме существенного повышения быстродействия.

Рассматривая абстрактную sc-машину обработки знаний на самом верхнем уровне, не уточняя (не детализируем) «внутреннее устройство» sc-агентов обработки знаний. Разработав язык SCP, мы получили возможность формально описывать (детализировать) поведение sc-агентов обработки знаний. Если трактовать язык SCP как ассемблер семантического ассоциативного компьютера, то проектирование этого компьютера можно рассматривать как формальный переход к sc-машинам более низкого уровня, обеспечивающим *интерпретацию sc-машин* более высокого уровня. Существенным здесь является то, что при этом мы не выходим за пределы класса абстрактных sc-машин. Просто вводится последовательность sc-языков программирования всё более низкого уровня, каждый из которого обеспечивает формальное описание sc-агентов, входящих в состав sc-машины, интерпретирующей программы непосредственно предшествующего ему sc-языка программирования более высокого уровня (см. Принцип 9). При этом таких уровней должно быть столько, сколько необходимо для доведения формального описания sc-машин до такого уровня детализации, который позволяет перейти от соответствующего абстрактного языка микропрограммирования к формальному описанию цифровой аппаратуры на языке VHDL.

Аппаратная интерпретация абстрактных sc-машин предполагает создание реконфигурируемой памяти с распределёнными в ней процессорными элементами. Такую интеграцию памяти и процессора будем называть **процессоро-памятью**. Реконфигурируемость (структурная перестраиваемость) памяти может быть обеспечена коммутационной средой для процессорных элементов. Можно рассматривать целый ряд подходов к реализации реконфигурируемой семантической ассоциативной процессоро-памяти. В частности, процессорным элементам можно ставить в соответствие узлы обрабатываемых унифицированных семантических сетей, а коммутируемым каналам связи между процессорными элементами — коннекторы этой семантической сети. В этом случае теку-

щее состояние конфигурации коммутируемых каналов связи будет полностью соответствовать текущему состоянию конфигурации обрабатываемой семантической сети. Следовательно, память «превращается» из пассивного хранилища байтов в коммутационную среду между процессорными элементами.

Принцип 18. Встроенные подсистемы интеллектуальных систем, обеспечивающие их эффективную эксплуатацию и эволюцию

Каждую проектируемую интеллектуальную систему трактовать как **результат интеграции следующих интеллектуальных подсистем:**

- предметной (основной) интеллектуальной системы;
- интеллектуального пользовательского интерфейса;
- интеллектуальной подсистемы адаптивного управления диалогом с конечным пользователем;
- интеллектуальной help-системы для информационного обслуживания и обучения конечных пользователей предметной интеллектуальной системы, которые, начиная работать с системой, не обязаны сразу иметь высокую квалификацию;
- интеллектуальные системы управления проектированием интеллектуальной системы, которая координирует деятельность разработчиков предметной интеллектуальной системы [Грибова, 2010];
- интеллектуальные системы управления информационной безопасностью предметной интеллектуальной системы.

Подчеркнём, что для обеспечения интегрируемости (семантической совместимости) перечисленных интеллектуальных систем они должны проектироваться на основе *одной и той же технологии*.

Таким образом, проектируя каждую интеллектуальную систему, необходимо одновременно проектировать:

- подсистему, которая осуществляет информационное обслуживание и обучение конечных пользователей данной интеллектуальной системы, т.е. фактически является оформлением документации по эксплуатации системы в виде интеллектуальной справочной и обучающей системы. Это существенно расширит контингент конечных пользователей, повысит эффективность эксплуатации системы и существенно упростит эту эксплуатацию;
- подсистему, которая обеспечивает координацию разработчиков проектируемой интеллектуальной системы, поскольку разработка (совершенствование) системы продолжается в ходе её эксплуатации и требует создания специальных методов и компьютерных средств постоянного совершенствования предметной интеллектуальной системы непосредственно в ходе её эксплуатации. Это существенно отодвинет срок её морального старения;
- подсистему, обеспечивающую управление информационной безопасностью проектируемой интеллектуальной системы.

Если подсистема управления проектированием интеллектуальной системы будет создаваться действительно как интеллектуальная система, интегрируемая с основной (предметной) интеллектуальной системой, то в перспективе она может стать не только координатором деятельности разработчиков, но и самостоятельным субъектом проектирования, способным тестировать, диагностировать, анализировать как основную проектируемую интеллектуальную систему, так и себя.

Принцип 19. Доступность и открытость технологии

Обеспечить максимально возможное **расширение контингента разработчиков** интеллектуальных систем, использующих предлагаемую технологию, за счёт максимальной доступности этой технологии и открытого характера её развития.

Если технология проектирования интеллектуальных систем ориентируется на широкое, массовое распространение и на интенсивное собственное развитие, опирающееся на накапливаемый опыт её использования, она должна быть доступной и открытой. Это означает:

- свободный доступ ко всей документации и основанным средствам автоматизации (компьютерной поддержке) проектирования интеллектуальных систем;
- открытость исходных текстов всех основных средств компьютерной поддержки проектирования интеллектуальных систем, всех основных многократно используемых (типовых) компонентов интеллектуальных систем;
- открытость исходных текстов всех «пилотных» проектов прикладных интеллектуальных систем, выполняющих роль «образцово-показательных» проектов;
- открытый характер организации (project-менеджмента) процесса развития технологии, имеющий форму открытого (open source) проекта, участником которого может быть любой желающий, в том числе, и любой пользователь этой технологии, указывающий на различные ошибки и высказывающий различные пожелания.

Завершая рассмотрение открытого характера предлагаемой технологии, сделаем следующие замечания:

- открытый характер технологии не является препятствием для реализации коммерческих интересов, связанных с этой технологией. Так, например, на коммерческой основе могут создаваться и предоставляться, во-первых, различные прикладные интеллектуальные системы; во-вторых, некоторые варианты реализации различных многократно используемых (типовых) компонентов интеллектуальных систем; в-третьих, некоторые варианты реализации интерпретатора абстрактных семантических логико-семантических моделей интеллектуальных систем, в частности, варианты построения семантических ассоциативных параллельных компьютеров;
- открытый характер технологии, при грамотном использовании фактора её открытости, способствует обеспечению информационной безопасности как самой технологии, так и прикладных интеллектуальных систем, созданных на её основе;
- открытый характер предлагаемой технологии проектирования интеллектуальных систем может быть эффективно реализован только на базе **технологии облачных вычислений**, в рамках которой вся предлагаемая технология проектирования интеллектуальных систем рассматривается как некий Internet-сервис [Грибова и др., 2011]

Принцип 20. Эволюционная методика проектирования

Использовать **методику поэтапного эволюционного проектирования** интеллектуальных систем.

Указанная методика предполагает:

- быстрое проектирование;
- скорейшее введение в эксплуатацию первых версий проектируемой системы с минимальными, но практически полезными возможностями;
- эволюционное поэтапное совершенствование проектируемой интеллектуальной системы путём её расширения новыми знаниями и навыками непосредственно в ходе её эксплуатации и активным привлечением конечных пользователей.

С формальной точки зрения, проектирование унифицированной логико-семантической модели (sc-модели) интеллектуальной системы, в конечном счёте, сводится к проектиро-

ванию sc-модели **базы знаний** этой интеллектуальной системы, поскольку scr-программы, описывающие поведение sc-агентов, можно рассматривать как часть базы знаний. Таким образом, проектируемая база знаний включает в себя:

- базу знаний предметной (основной) интеллектуальной системы;
- тексты всех scr-программ, описывающих поведение sc-агентов;
- текст документации, представленный в виде базы знаний интеллектуальной help-системы, обеспечивающей всестороннее информационное обслуживание пользователей проектируемой интеллектуальной системы.

Начальный этап проектирования базы знаний интеллектуальной системы — уточнение иерархической системы предметных областей, которые должны быть описаны в проектируемой базе знаний. Каждой такой предметной области ставится в соответствие определённый раздел проектируемой базы знаний. Среди выделенных разделов проектируемой базы знаний имеются разделы, которые делятся (декомпозируются) на подразделы, а также атомарные (недекомпозируемые) разделы. Далее процесс проектирования всей базы знаний сводится к проектированию каждого её атомарного раздела с последующей их интеграцией в единую базу знаний.

В целом начальную стадию проектирования всей интеллектуальной системы на основе предлагаемой технологии можно условно разбить на четыре этапа:

1. Разработка первой версии интеллектуальной системы, которая включает в себя:
 - первую версию её базы знаний;
 - *типовое ядро* интеллектуальной информационно-поисковой машины, которое входит в состав библиотеки многократно используемых компонентов интеллектуальных систем;
 - *типовое ядро* интеллектуального решателя;
 - *типовое ядро* пользовательского интерфейса.
2. Разработанная первая версия интеллектуальной системы уже обладает определённой целостностью, её можно тестировать и запускать в предварительную опытную эксплуатацию.
3. Разработка второй версии интеллектуальной системы, которая включает в себя:
 - вторую версию её базы знаний;
 - первую версию её информационно-поисковой машины;
 - типовое ядро её интеллектуального решателя;
 - типовое ядро её пользовательского интерфейса.
4. Разработка третьей версии интеллектуальной системы, включающей в себя:
 - третью версию её базы знаний;
 - вторую версию её информационно-поисковой машины;
 - первую версию её интеллектуального решателя;
 - типовое ядро её пользовательского интерфейса.
5. Разработка четвертой версии интеллектуальной систем, включающей в себя:
 - четвертую версию её базы знаний;
 - третью версию её информационно-поисковой машины;
 - вторую версию её интеллектуального решателя;
 - первую версию её пользовательского интерфейса.

Дальнейшее развитие проектируемой интеллектуальной системы может акцентировать внимание на самых разных направлениях, приоритетность которых определяется самим приложением.

Более подробно методика эволюционного коллективного проектирования унифицированных семантических моделей интеллектуальных систем, на основе содержательной структуризации знаний (см. Принцип 6), описана в работе [20].

Принцип 21. РЕАЛИЗАЦИЯ ПРЕДЛАГАЕМОЙ ТЕХНОЛОГИИ В ВИДЕ ИНТЕЛЛЕКТУАЛЬНОЙ МЕТАСИСТЕМЫ

Реализовать предлагаемую технологию как **интеллектуальную метасистему, ориентированную на поддержку проектирования интеллектуальных систем**, построенную по тем же самым принципам (т. е. по той же технологии), что и интеллектуальные системы, разрабатываемые на её основе.

Указанная интеллектуальная система должна включать в себя:

- теорию (принципы построения) проектируемых интеллектуальных систем, которая входит в состав базы знаний метасистемы;
- библиотеку типовых многократно используемых компонентов (ip-компонентов) интеллектуальных систем, которая также входит в состав базы знаний рассматриваемой метасистемы;
- средства автоматизации синтеза, анализа и имитационного моделирования проектируемых интеллектуальных систем и их компонентов;
- интеллектуальную help-систему, являющуюся подсистемой рассматриваемой интеллектуальной метасистемы, ориентированной на информационное обслуживание и обучение разработчиков интеллектуальных систем;
- методику проектирования интеллектуальных систем, которая оформляется как часть базы знаний метасистемы;
- методику обучения проектированию интеллектуальной системы, которая также является частью базы знаний метасистемы;
- интеллектуальную подсистему управления проектированием самой метасистемы;
- интеллектуальную подсистему управления информационной безопасностью метасистемы;
- семейство различных вариантов реализации интерпретаторов унифицированных абстрактных логико-семантических моделей интеллектуальных систем.

Учитывая рассматриваемые выше принципы построения предлагаемой нами технологии, она названа Открытой Семантической Технологией проектирования Интеллектуальных Систем (Open Semantic Technology for Intelligent Systems — OSTIS). Можно было бы назвать её **SC-технологией**, поскольку основой этой технологии является SC-код. Соответственно этому, интеллектуальную метасистему, ориентированную на поддержку проектирования интеллектуальных систем, будем называть **метасистемой OSTIS**.

В интеллектуальной метасистеме OSTIS можно выделить целый ряд подсистем, ориентированных на поддержку проектирования различных компонентов интеллектуальных систем, таких как:

- базы знаний и различные фрагменты баз знаний (онтологии, формальные теории, программы);
- информационно-поисковые машины, машины интеграции знаний, решатели задач;
- пользовательские интерфейсы (графические, естественно-языковые, мультимодальные).

В интеллектуальной метасистеме OSTIS можно также выделить семейство интеллектуальных подсистем, ориентированных на поддержку проектирования различных классов интеллектуальных систем:

- интеллектуальные справочные системы (системы информационного обслуживания);

- интеллектуальные обучающие системы (имеющие подсистемы интеллектуального управления обучением);
- интеллектуальные help-системы для пользователей различных компьютерных систем;
- интеллектуальные системы автоматизированного проектирования;
- интеллектуальные системы управления проектами.

Заключение

В предлагаемой технологии OSTIS существенными являются весь целостный комплекс этих принципов и их максимально возможная согласованность.

Ключевые проблемы, лежащие в основе предлагаемой технологии:

- обеспечение *семантической совместимости* (интегрируемости) различных моделей представления и обработки знаний;
- создание *общей теории* абстрактных семантических моделей интеллектуальных систем, не противопоставляя, а интегрируя самые различные подходы;
- обеспечение максимальной возможной *независимости* интеллектуальных систем от многообразия вариантов и платформ их технической реализации (в т. ч. и от будущих компьютеров, специально ориентированных на аппаратную поддержку обработки знаний).

Список литературы

1. Айзерман М.А. Динамический подход к анализу структур, описываемых графами (основы графодинамики) / М.А. Айзерман, Л.А. Гусев, С.В. Петров, И.М. Смирнова, Л.А. Тененбаум // Исследования по теории структур. М.: Наука, 1988. С. 5–76.
2. Амосов Н.М. Активные семантические сети в роботах с автономным управлением / Н.М. Амосов, А.М. Касаткин, Л.М. Касаткина // Труды IV Международной объединённой конференции по искусственному интеллекту: Т. 9. М., 1975.
3. Бениаминов Е.М. Основания категорного подхода к представлению знаний. Категорные средства / Е.М. Бениаминов // Изв. АН СССР. Техн. кибернет. 1988. № 2. С. 21–33.
4. Борщев В.Б. Схемы на клубных системах и вегетативная машина / В.Б. Борщев // Семиотика и информатика. 1983. Вып. 22. С. 3–44.
5. Берштейн Л.С. Использование расплывчатых ориентированных гиперграфов второго рода для представления фреймовых моделей / Л.С. Берштейн, Д.М. Башмаков // Методы построения алгоритмических моделей сложных систем. Таганрог, 1988. Вып. 7. С. 64–68.
6. Вагин, В.Н. Достоверный и правдоподобный вывод в интеллектуальных системах / В.Н. Вагин [и др.]. М.: ФИЗМАТЛИТ, 2008.
7. Вольфенгаген В.Э. Системы представления знаний с использованием семантических сетей / В.Э. Вольфенгаген, О.В. Воскресенская, Ю.Г. Горбанев // Вопросы кибернетики. Интеллектуальные банки данных. М.: АН СССР, 1979. С. 49–69.
8. Гаврилова Т.А. Базы знаний интеллектуальных систем / Т.А. Гаврилова, В.Ф. Хорошевский. СПб.: Питер, 2000.
9. Гастев Ю.А. Гомоморфизмы и модели. Логико-алгебраические аспекты моделирования / Ю.А. Гастев. М.: Наука, 1975.
10. Георгиев В.О. Модели представления знаний предметных областей диалоговых систем (обзор) / В.О. Георгиев. // Изв. АН СССР. Техн. кибернет. 1993. № 5. С. 24–44.

10. *Гладун В.П.* Процессы формирования новых знаний / В.П. Гладун. София: Педагог, 1994.
11. *Голенков В.В.* Представление и обработка знаний в графодинамических ассоциативных машинах / В.В. Голенков [и др.]. Мн.: БГУИР, 2001.
12. *Голенков В.В.* Графодинамические ассоциативные модели и средства параллельной обработки информации в системах искусственного интеллекта / В.В. Голенков, Н.А. Гулякина // Доклады БГУИР. 2004. № 1(5). С. 92–101.
13. *Голенков В.В.* Принципы построения массовой семантической технологии компонентного проектирования интеллектуальных систем / В.В. Голенков, Н.А. Гулякина // Материалы международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем». Минск, 2011. С. 21–58.
14. *Гостев Ю.Г.* Описание семантики программ с помощью подстановок на графах // Программирование. 1981. № 5. С. 11–17.
15. *Гостев Ю.Г.* Описание структур данных с помощью графопорождающих грамматик / Ю.Г. Гостев // Программирование. 1981. № 2. С. 44–51.
16. *Гуляева Д.М.* Решение прикладных задач на расширенных семантических сетях. / Д.М. Гуляева // Математическое обеспечение ЭВМ и систем программирования. М., 1989.
17. *Гулякина Н.А.* Языки и технологии программирования, ориентированные на обработку семантических сетей / Н.А. Гулякина, Д.А. Лазуркин, О.В. Пивоварчик // Материалы международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем». Минск, 2012.
18. *Гусаков В.Я.* Динамические алгебраические системы как математическая модель банка данных / В.Я. Гусаков, С.М. Гусакова // Семиотика и информатика. 1981. Вып. 17. С. 43–52.
19. *Давыденко И.Т.* Комплексная методика эволюционного коллективного проектирования семантических моделей интеллектуальных справочных систем / И.Т. Давыденко // Материалы международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем». Минск, 2012.
20. *Евгеньев Г.Б.* Технология создания многоагентных прикладных систем / Г.Б. Евгеньев // Одиннадцатая национальная конференция по искусственному интеллекту: Труды конференции. Т. 2. М., 2008. С. 306–312.
21. *Епифанов М.Е.* Индуктивное обобщение в ассоциативных сетях / М.Е. Епифанов // Известия АН СССР. Техническая кибернетика. № 5, 1984. С. 132–146.
22. *Ефимова С.М.* П-графы для представления знаний / С.М. Ефимова. М: Вычислительный центр АН СССР, 1985.
23. *Ефимова С.М.* Поиск в базах знаний, опирающихся на модель П-графов, и его аппаратная реализация на основе метода M^3 / С.М. Ефимова, Е.В. Суворова. М: Вычислительный центр АН СССР, 1988.
24. *Загорюлько Ю.А.* Технология конструирования средств обработки знаний на основе семантических сетей. Средства спецификации и настройки / Ю.А. Загорюлько. Новосибирск, 1988.
25. *Заливако С.С.* Семантическая технология компонентного проектирования интеллектуальных решателей задач / С.С. Заливако, Д.В. Шункевич // Материалы международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем». Минск, 2011.
26. *Ивашенко В.П.* Семантические модели и средства интеграции баз знаний / В.П. Ивашенко // Материалы международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем». Минск, 2011.
27. *Калиниченко Л.А.* Методы и средства интеграции неоднородных баз данных / Л.А. Калиниченко. М.: Наука, 1983.
28. *Кандрашина Е.Ю.* Представление знаний о времени и пространстве в интеллектуальных системах / Е.Ю. Кандрашина, Л.В. Литвинцева, Д.А. Поспелов. М.: Наука, 1989.
29. *Карабеков Б.А.* Система «Бинарная Модель Знаний» как инструмент для концептуального моделирования бизнес-процессов // Одиннадцатая национальная конференция по искусственному интеллекту: Труды конференции. Т. 2. М., 2008. С. 282–291.

30. Клещев А.С. Семантические порождающие модели. Общая точка зрения на фреймы и продукции в экспертных системах / А.С. Клещев. Владивосток, 1986.
31. 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, 2011.
32. Колмогоров А.Н. К определению алгоритма / А. Н. Колмогоров // Успехи математических наук. 1958. Т. 13. № 4(82). С. 3–28.
33. Корончик Д.Н. Семантические модели мультимодальных пользовательских интерфейсов и семантическая технология их проектирования / Д.Н. Корончик // Материалы международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем». Минск, 2011.
34. Котов В.Е. Асинхронные вычислительные процессы над общей памятью / В.Е. Котов, А.С. Нариньяни // Кибернетика. 1966. № 3. С. 64–71.
35. Касьянов В.Н. Графы в программировании: обработка, визуализация и применение / В.Н. Касьянов, В.А. Евстигнеев // ВHV — Санкт-Петербург, 2003.
36. Кузнецов В.Е. Представление в ЭВМ неформальных процедур / В.Е. Кузнецов. М.: Наука, 1989.
37. Кузнецов И.П. Семантические представления / И.П. Кузнецов. М.: Наука, 1986.
38. Лозовский В.С. Семантические сети / В. С. Лозовский // Представление знаний в человеко-машинных и робототехнических системах. М.: ВИНТИ, 1984. С. 84–121.
39. Лозовский В.С. Экстенциональная база знаний на основе семантических сетей / В.С. Лозовский // Известия АН СССР. Техн. кибернет. 1982. № 5. С. 23–42.
40. Любарский Ю.Я. Интеллектуальные информационные системы / Ю.Я. Любарский. М.: Наука, 1980.
41. Люгер Дж.Ф. Искусственный интеллект: стратегии и методы решения сложных проблем / Дж.Ф. Люгер. М.: Вильямс, 2003.
42. Мальцев А.И. Алгебраические системы / А.И. Мальцев. М.: Наука, 1970.
43. Марковский А.В. Анализ структуры знаковых ориентированных графов / А.В. Марковский. // Известия РАН: Теория и системы управления. № 5. 1997.
44. Мартынов В.В. Универсальный семантический код / В.В. Мартынов. Минск: Наука и техника, 1977.
45. Мельчук И.А. Опыт теории лингвистических моделей «Смысл-Текст». Семантика, синтаксис / И.А. Мельчук. М.: Наука, 1974.
46. Месарович М. Общая теория систем: математические основы / М. Месарович, Я. Тахакара. М.: Мир, 1978.
47. Молокова О.С. Методология анализа предметных знаний / О.С. Молокова. // Новости искусственного интеллекта. 1992. № 3. С. 11–60.
48. Нариньяни А.С. НЕ-факторы и инженерия знаний: от наивной формализации к естественной программатике / А.С. Нариньяни // КИИ-94. Сборник трудов Национальной конференции с международным участием по ИИ. «Искусственный интеллект-94»; в 2-х т. Т. 1. Тверь: АИИ, 1994. С. 9–18.
49. Осипов Г.С. Построение моделей предметных областей. Неординарные семантические сети / Г.С. Осипов. // Известия АН СССР. Техническая кибернетика. № 5. 1990.
50. Петров С.В. Графовые грамматики и автоматы (обзор) / С.В. Петров. // Автоматика и телемеханика. № 7. 1978. С. 116–136.

51. Петрушкин В.А. Экспертно-обучающие системы / В.А. Петрушкин. Киев: Наукова думка. 1992.
52. Плесневич Г.С. Денотационная семантика ассоциативных сетей / Г.С. Плесневич // Семиотика и информатика. 1983. Вып. 21.
53. Плесневич Г.С. Представление знаний в ассоциативных сетях / Г.С. Плесневич // Изв. АН СССР. Техн. кибернет. 1982. № 5. С.6–22.
54. Плесневич Г.С. Бинарные модели знаний / Г.С. Плесневич // Труды Международных научно-технических конференций «Интеллектуальные системы» (AIS'08) и «Интеллектуальные САПР» (CAD-2008). Научное издание в 4-х томах. Т. 2. М.: Физматлит, 2008.
55. Попков В.К. Гиперсети и их характеристики связности / В.К. Попков. // Исследования по прикладной теории графов. Новосибирск: Наука, 1986. С. 25–58.
56. Поспелов Д.А. Представление знаний. Опыт системного анализа / Д.А. Поспелов. // Системные исследования. Методологические проблемы. Ежегодник. М.: Наука, 1986. С. 83–102.
57. Поспелов Д.А. Ситуационное управление. Теория и практика / Д.А. Поспелов. М.: Наука, 1986.
58. Рабинович З.Л. О концепции машинного интеллекта и её развитии / З.Л. Рабинович // Кибернетика и системный анализ. 1995. № 2. С. 163–173.
59. Рассел С. Искусственный интеллект: современный подход / С. Рассел, П. Норвиг. М.: Вильямс, 2006.
60. Резанов С.Н. Об одном методе обобщения на семантических сетях в системе управления энергообъединением / С.Н. Резанов // Изв. АН СССР. Техн. кибернет. 1989. № 5. С. 55–62.
61. Рубашкин В.Ш. Представление и анализ смысла в интеллектуальных информационных системах / В.Ш. Рубашкин. М.: Наука, 1989.
62. Рыбина Г.В. Основы построения интеллектуальных систем: учеб. пособ. / Г.В. Рыбина. М.: Финансы и статистика, 2010.
63. Сапатый П.С. Об эффективности структурной реализации операций над семантическими сетями / П.С. Сапатый // Техн. кибернет. 1983. № 5. С. 128–134.
64. Семёнов В.В. Семантические фреймовые сети как модели предметной области для САПР САУ / В.В. Сапатый // Представление знаний в системах искусственного интеллекта. М.: МДНТП, 1980. С. 117–122.
65. Скороходько Э.Ф. Семантические сети и автоматическая обработка текста. / Э.Ф. Скороходько. Киев: Наук. думка, 1983.
66. Скрэгг Г. Семантические сети как модели памяти / Г. Скрэгг // Новое в зарубежной лингвистике. Вып. 12. М.: Радуга, 1983. С. 228–271.
67. Соловьёв В.А. Формирование на семантической сети понятий и суждений с помощью рассуждений по аналогии / В.А. Соловьёв // II Всесоюзная конференция «Искусственный интеллект-90». Секционные и стендовые доклады. Минск, 1990. Т. 1. С. 166–169.
68. Тузов В.А. Математическая модель языка / В.А. Тузов. Л.: Изд-во ленингр. ун-та, 1984.
69. Тузов В.А. О формализации понятия задачи / В.А. Тузов. М.: Наука, 1986. С. 73–83.
70. Тыгу Э.Х. Интеграция знаний / Э.Х. Тыгу // Изв. АН СССР. Техн. кибернет. 1989. № 5. С. 3–13.
71. Финн В.К. Многозначные логики и их применения / ред. В.К. Финн. М.: ЛКИ, 2008. Т. 1, Т.2.
72. Уварова Т.Г. Формальное описание операционного языка для семантических сетей. / Т.Г. Уварова, Л.Л. Лифшиц. М.: ВЦ АН СССР, 1987.
73. Хельбиг Г. Семантическое представление знаний в вопросно-ответной системе FAS-80 / Г. Хельбиг. // Представление знаний и моделирование процессов понимания. Новосибирск, 1980. С. 97–123.
74. Хендрикс Г. О расширении применимости семантических сетей введением разбиений / Г.О. Хендрикс // Труды IV Международной объединённой конференции по искусственному интеллекту. М., 1975. Т. 1. С. 190–206.

75. 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.
76. Цаленко М.Ш. Моделирование семантики в базах данных. / М.Ш. Цаленко. М.: Наука, 1989.
77. Шенк Р. Обработка концептуальной информации / Р. Шенк. Москва: Энергия, 1980.
78. Шрейдер Ю.А. Системы и модели / Ю.А. Шрейдер, А.А. Шаров. М.: Радио и связь, 1982.
79. Шуберт Л. Усиление выразительной мощности семантических сетей / Л. Шуберт // Кибернетический сборник. Новая серия. 1979. Вып. 16. С. 171–212.
80. Naidenova X. Machine Learning Methods for Commonsense Reasoning Processes: Interactive Models (Premier Reference Source)/ Xenia Naidenova // Information Science Reference; 1 edition (October 31, 2009).
81. Открытая семантическая технология проектирования интеллектуальных систем [Электронный ресурс]. 2011. Режим доступа: <http://ostis.net>
82. Russell S. Artificial Intelligence. A Modern Approach / S Russell, P Norvig. New Jersey: Prentice Hall, 1995.
83. Sowa J. Semantic networks / John F. Sowa // Encyclopedia of Artificial Intelligence, edited by S. C. Shapiro. New York: Wiley, 1992.
84. Sowa J. Conceptual Graphs/ John F. Sowa, F. van Harmelen, V. Lifschitz, B. Porter// eds., Handbook of Knowledge Representation, Elsevier, 2008, pp. 213–237.
85. Wooldridge M. Agent Theories, Architectures and Languages: A Survey / M. Wooldridge, N. Jennings // Intelligent Agents. Languages. Amsterdam: Springer Verlag, August, 1994. P. 3–39.

Сведения об авторах

Голенков В.В. —

доктор технических наук, профессор, заведующий кафедрой «Искусственный интеллект» Белорусского государственного университета информатики и радиоэлектроники. Golen@bsuir.ru

Гулякина Н.А. —

кандидат физико-математических наук, доцент, заместитель заведующего кафедрой «Искусственный интеллект» Белорусского государственного университета информатики и радиоэлектроники. Guliakina@bsuir.ru