

УДК 004.03:004.9

ПРОЕКТ ОТКРЫТОЙ СЕМАНТИЧЕСКОЙ ТЕХНОЛОГИИ КОМПОНЕНТНОГО ПРОЕКТИРОВАНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ. ЧАСТЬ 1: ПРИНЦИПЫ СОЗДАНИЯ

В.В. Голенков¹, Н.А. Гулякина²

Белорусский государственный университет информатики и радиоэлектроники, Минск, Белоруссия
¹*golen@bsuir.by*, ²*guliakina@bsuir.by*

Аннотация

В работе рассматривается открытый проект, направленный на создание и развитие технологии компонентного проектирования интеллектуальных систем. Первая часть статьи содержит разработанные авторами принципы построения технологии проектирования интеллектуальных систем, основанных на онтологиях и семантическом представлении знаний. Обсуждается использование накопленного опыта современных информационных технологий, применение графодинамических моделей обработки информации. В частности, предлагается ориентироваться на параллелизм и асинхронность, на смысловые модели представления и обработки знаний, на графовые языки программирования. Авторы рекомендуют делать упор на унификацию абстрактного смыслового представления знаний, унификацию структуризации баз знаний и формального описания поведения агентов обработки информации в графодинамической семантической памяти.

Ключевые слова: интеллектуальная система, технология проектирования, семантическая сеть, SC-код, предметная область, онтология, графовый язык программирования, язык SCP.

Введение

Основным результатом исследований в области искусственного интеллекта (ИИ) является не только разработка различного вида интеллектуальных систем (ИС), но и создание технологий, позволяющих быстро разрабатывать ИС, имеющие практическую ценность. Составляющими таких технологий являются:

- формальная теория ИС;
- методы проектирования ИС;
- средства автоматизации проектирования ИС;
- средства информационной поддержки разработчиков ИС;
- средства компьютерной поддержки управления коллективной разработкой ИС.

Анализ современных технологий ИИ показывает, что наряду с впечатляющими достижениями они имеют целый ряд серьезных недостатков. К числу таких недостатков, в частности, относятся:

- технологии ИИ не ориентированы на широкий круг разработчиков ИС и, следовательно, не получили массового распространения;
- велики сроки разработки ИС и велика трудоемкость их сопровождения;
- высока степень зависимости технологий ИИ от платформ, на которых они реализованы, что является причиной высокой трудоемкости переноса ИС на новые платформы;
- для эффективной реализации существующих моделей представления знаний и моделей решения трудно формализуемых задач современные компьютеры оказываются плохо приспособленными, что требует разработки принципиально новых компьютеров;
- современное состояние в области проектирования ИС представляет собой «вавилонское столпотворение» самых различных моделей, методов, средств, платформ;

- отсутствуют подходы, позволяющие на универсальной основе интегрировать научные и практические результаты в области искусственного интеллекта, что порождает высокую степень дублирования результатов;
- отсутствует общее унифицированное решение проблемы семантической совместимости компьютерных систем (КС), что порождает высокую трудоемкость создания комплексных интегрированных КС. Наиболее остро проблема совместимости КС проявляет себя при разработке web-ориентированных систем;
- отсутствует методика конструктивного использования опыта завершенных разработок ИС, что порождает высокую степень дублирования разработок различных компонентов этих систем;
- отсутствует унифицированная методика совершенствования различных структурных уровней КС в процессе их сопровождения. Обновление КС часто сводится к разработке различного рода «заплаток», которые устраняют не причины выявленных недостатков обновляемых КС, а только некоторые следствия этих причин.

Перечислим общие подходы и шаги, направленные на устранение указанных выше недостатков современных информационных технологий (ИТ) и в том числе технологий ИИ.

Во-первых, всё, что в настоящее время делается в ИТ, необходимо привести в стройную систему и сделать доступным пользователю, не являющемуся специалистом в области ИТ, но желающему самостоятельно разработать КС в своей предметной области (Про).

Во-вторых, минимизировать участие специалистов по информатике в разработке прикладных КС и полностью исключить такое участие в разработке прототипов этих систем. Специалисты в области ИТ должны заниматься не разработкой приложений, а разработкой и совершенствованием технологий (моделей, средств и методов), создающих комфортные условия для разработчиков приложений.

В-третьих, ориентироваться на разработку таких моделей представления и обработки информации, которые были бы комфортны для пользователя.

В основе предлагаемого подхода к созданию технологии проектирования ИС лежат описанные ниже принципы.

1 Использование опыта современных технологий

В первую очередь, имеется в виду технология проектирования микросхем, которая за последнее время обеспечила сокращение времени и повышение качества разработок:

- благодаря созданию языковых средств формального описания проектируемых микросхем на разных уровнях детализации;
- благодаря четкому разделению процесса разработки формальных описаний микросхем и процесса их реализации по заданным формальным описаниям;
- благодаря созданию мощных и доступных библиотек формальных описаний типовых многократно используемых компонентов микросхем.

Для того чтобы построить *технология проектирования ИС*, необходимо:

- 1) создать языковые средства полного унифицированного формального описания ИС;
- 2) отделить разработку полного унифицированного формального описания проектируемой ИС от разработки различных вариантов интерпретации таких формальных описаний ИС;
- 3) создать библиотеку формальных описаний типовых компонентов ИС. Для этого необходимо обеспечить интегрируемость указанных компонентов ИС.

2 Ориентация на графодинамические модели обработки информации

В качестве формальной основы проектируемых ИС предлагается использовать *графодинамические модели* представления и обработки информации.

В графодинамической модели обработки информации процесс обработки информации рассматривается как процесс преобразования *графовой структуры* (ГС), в ходе которого меняется не только состояние элементов этой ГС, но и конфигурация этой структуры: появляются или удаляются её вершины, а также связи между ними. Для создания графодинамических моделей обработки информации в ИС недостаточно тех видов графовых структур, которые в настоящее время исследуются в теории графов. Здесь требуются не только рёбра, дуги, гиперрёбра (многокомпонентные неориентированные связки), но и *гипердуги*, представляющие собой многокомпонентные ориентированные связки, являющиеся ролевыми структурами, компоненты которых выполняют в рамках этих связей дополнительно указываемые роли. Кроме связей, связывающих вершины, потребуются также связки, компонентами которых являются другие связки. Также потребуются *подструктуры*, компонентами которых являются вершины вместе с инцидентными им (со связывающих их) связками, т.е. структуры, являющиеся фрагментами заданной ГС [1].

Приведем общее определение ГС, на основе которого можно строить практически полезные графодинамические модели обработки информации.

Графовая структура G задается пятеркой $\langle V, C, K, M, I \rangle$, где:

V – множество *вершин* (первичных элементов, терминальных элементов);

C – множество *связующих элементов* ГС, каждый из которых задает некоторый фрагмент ГС;

K – множество *ключевых вершин* ГС, каждая из которых задает некоторый класс эквивалентных (однотипных) в определенном смысле элементов ГС ($K \subset V$);

M – множество *меток* элементов (алфавит элементов) ГС, каждая из которых задает некоторый базовый класс эквивалентных элементов ГС. К таким классам элементов, в частности, относятся следующие классы:

- вершин ГС,
- связующих элементов ГС,
- ключевых вершин ГС,
- меток ГС,
- отношений инцидентности, заданных на множестве элементов ГС;

I – множество используемых в ГС *отношений инцидентности*, заданных на множестве её элементов. Все эти отношения инцидентности являются бинарными ориентированными отношениями. Среди этих отношений выделим:

- отношения инцидентности вершин,
- отношения инцидентности, каждая пара которых связывает связующий элемент ГС с элементом того фрагмента, который задается (обозначается) этим элементом,
- отношения инцидентности, каждая пара которых связывает ключевую вершину ГС с тем элементом ГС, который является экземпляром того класса элементов, который задается (обозначается) этой ключевой вершиной,
- отношения инцидентности, каждая пара которых связывает метку ГС с тем элементом ГС, который имеет указанную метку.

Связь каждого отношения инцидентности (каждого элемента множества I) с соответствующими парами инцидентности и связь каждой пары инцидентности с элементами ГС, соединяемыми этой парой инцидентности, можно условно считать неявно задаваемыми связями инцидентности более низкого уровня.

Каждую ГС G будем трактовать как множество всех элементов, входящих в её состав:

$$G = (V \cup C \cup K \cup M \cup I).$$

Таким образом, в число элементов ГС входят все её вершины (в том числе ключевые), связующие элементы, метки и отношения инцидентности.

Множество связующих элементов ГС можно разбить на множества *связок* (простых связующих элементов) и *подструктур*.

В свою очередь, по признаку ориентированности множество связок можно разбить на множества:

- *ориентированных связок*, компоненты которых выполняют в рамках этих связок в общем случае разные роли;
- *неориентированных связок*, все компоненты которых выполняют в рамках этих связок одинаковые роли.

Важным частным случаем ориентированной связки является *кортеж*. Кортеж задает такое подмножество элементов ГС, в котором роли всех элементов пронумерованы. Если в ГС имеются кортежи, то в число её отношений инцидентности должны входить следующие отношения «быть компонентом»: первым, вторым, третьим компонентом и т. д.

По количеству компонентов множество связок можно разбить на множества:

- унарных связок;
- бинарных связок;
- многокомпонентных связок, имеющих более двух компонентов.

Неориентированные бинарные связки будем называть *рёбрами*, ориентированные – *дугами*. Неориентированные многокомпонентные связки будем называть гиперребрами, а ориентированные – гипердугами.

Интерес к графодинамическим моделям обработки информации имеет длительную историю. Достаточно отметить предложенное А.Н. Колмогоровым уточнение понятия алгоритма [2], работы школы М.А. Айзермана по графодинамике [3], исследования по графовым грамматикам [4], исследования по теории программирования и CASE-технологиям [5], исследования по параллельным моделям обработки информации [6], предложенные В.Б. Борщевым и М.В. Хомяковым клубные системы и вегетативная машина [7].

Для разработки графодинамических моделей обработки информации необходимо рассматривать ГС с позиций семиотики и трактовать её как *знаковую конструкцию* (текст), представляющую собой систему взаимосвязанных *знаков*. Такая трактовка ГС позволяет «вдохнуть» семантику в теорию графов.

Говоря о *графовых языках*, следует подчеркнуть то, что ГС, являющиеся текстами таких языков, представляют собой абстрактные математические структуры, не требующие уточнения способа их материального представления (например, способа кодирования в компьютерной памяти, способа графического изображения, ориентированного на человеческое восприятие). ГС как абстрактный математический объект и её, например, графическое представление – это принципиально разные вещи. Из этого, в частности, следует, что каждому графовому языку может соответствовать несколько языков, использующих разные способы представления текстов этого графового языка.

Накопленный опыт развития и применения теории графов и все полученные в ней результаты являются хорошим математическим фундаментом для разработки различных графовых языков и различных графодинамических моделей обработки информации, а также для создания теории таких языков и моделей. На стыке теории графов и семиотики может появиться раздел семиотики – графовая семиотика (семиотика нелинейных знаковых конструкций).

2.1 Ориентация на параллелизм и асинхронность в графодинамических моделях

В качестве формальной основы проектируемых ИС предлагается использовать графодинамические модели, ориентированные на *параллельную и асинхронную обработку информации*. Без организации параллельной обработки информации невозможно рассчитывать на необходимую производительность подавляющего числа практически полезных ИС. Асинхронные модели обработки информации являются более гибкими, их легче интегрировать и наращивать новыми функциональными возможностями.

Графодинамическая модель параллельной асинхронной обработки информации, которую будем также называть *графодинамической параллельной асинхронной машиной*, трактуется нами как абстрактная *многоагентная система*, состоящая из:

- абстрактной *графодинамической памяти*, в которой хранятся обрабатываемые графовые структуры;
- *агентов*, работающих с общей для них графодинамической памятью и обменивающихся информацией через эту память (в т.ч. и для координации своих действий).

Графодинамическая память носит реконфигурируемый, структурно перестраиваемый характер, поскольку процесс обработки графовых структур, в конечном счете, сводится к генерации и удалению различных элементов графовых структур, а также к генерации и удалению пар инцидентности между этими элементами, т.е. процесс обработки информации в графодинамической памяти сводится не только к изменению состояния элементов памяти, но и к изменению конфигурации связей между ними.

Агенты, работающие с общей графодинамической памятью, делятся на три вида:

- внутренние агенты, каждый из которых реагирует на определенного вида ситуации или события в графодинамической памяти и осуществляет изменение состояния графодинамической памяти, соответствующее своему функциональному назначению;
- рецепторные агенты, каждый из которых реагирует на определенные события во внешней среде и осуществляет первичное отражение этих событий в графодинамической памяти;
- эффекторные агенты, каждый из которых реагирует на определенного вида команды, формируемые внутренними агентами в графодинамической памяти, и осуществляет соответствующее изменение материального (физического) состояния ИС, которое определенным образом влияет на изменение её внешней среды.

Агенты могут работать параллельно, если одновременно возникают условия инициирования агентов.

Асинхронность деятельности внутренних агентов заключается в том, что наличие условия инициирования агента ещё не означает начала его работы. Время реакции каждого внутреннего агента в известной мере субъективно и достаточно произвольно. В этом смысле указанные агенты обладают свободой выбора момента начала реакции на условие инициирования и свободой выбора последовательности обработки условий инициирования, если в текущий момент таких условий возникло несколько.

Для обеспечения эффективного взаимодействия агентов, работающих с общей графодинамической памятью, наряду с представляемой им свободой, необходима разработка таких правил их поведения, которые гарантируют безопасность и производительность каждого из них. В конечном счете, эти правила сводятся к двум положениям:

- забота о безопасности, точнее, об обеспечении безопасного выполнения своей задачи;
- невмешательство в работу других агентов.

Для обеспечения безопасного выполнения своей задачи агент блокирует некоторые элементы ГС, которая хранится в общей графодинамической памяти. Блокировка – это запрет, установленный заданным агентом и адресованный другим агентам, на выполнение тех или

иных действий над заданным элементом хранимой ГС. Таким образом, существует несколько видов таких блокировок, например:

- запрет на удаление заданного элемента ГС;
- запрет на удаление всех элементов хранимой ГС, инцидентных заданному (блокируемому) элементу;
- запрет на удаление/генерацию заданного вида элементов хранимой ГС, которые связаны с блокируемым элементом выходящей из него (или входящей в него) парой инцидентности, принадлежащей заданному отношению инцидентности.

Приведем некоторые правила поведения агента, работающего с общей графодинамической памятью:

- не нарушать блокировочные запреты, сформированные другими агентами;
- блокировать тот фрагмент обрабатываемой ГС, целостность которого необходимо сохранить до завершения своей работы;
- не «жадничать» – не блокировать больше, чем надо;
- снимать свои блокировки как можно быстрее, как только в них отпадает необходимость;
- удалять сгенерированные для своей работы вспомогательные структуры, как только в них отпадает необходимость;
- поиск фрагментов хранимой ГС, являющихся условиями инициирования агента, осуществлять поэтапно, начиная с поиска тех частей этих условий, которые реже появляются в памяти (это необходимо для того, чтобы скорее установить факт отсутствия условий инициирования).

В случае возникновения конфликтов между агентами используются внутренние агенты специального вида, реагирующие на возникновение таких конфликтов и обеспечивающие их разрешение (такие агенты будем называть метаагентами-судьями).

Для организации своей деятельности над графодинамической памятью каждый агент «опирается» на соответствующее ему семейство постоянно присутствующих в памяти элементов хранимой ГС. Указанные элементы будем называть *ключевыми элементами агентов*, которые соответствуют константам программ, описывающих поведение агентов.

Сложность комплексного перехода на графодинамическую парадигму параллельной асинхронной обработки информации определяется исключительно психологическими обстоятельствами. Но накопленный человечеством опыт по созданию КС и, в частности, ИС позволяет этот переход сделать достаточно быстро, так как многие проблемы, возникающие при реализации и применении графодинамических моделей, имеют достаточно близкие аналоги в традиционных КС, но решаются значительно проще. Эпицентром такого перехода является формализация смысла и разработка семантически совместимых языковых средств представления различных видов знаний.

2.2 Ориентация на смысловые модели представления и обработки знаний в графодинамических моделях

В качестве формальной основы проектируемых ИС, в качестве основы абстрактных логико-семантических моделей ИС, предлагается использовать графодинамические модели специального вида – *смысловые модели представления и обработки знаний*, в основе которых лежат *семантические сети* [8-13].

Фактически, речь идет о создании формальных средств описания *смысла* различных видов знаний и формальных средств описания обработки знаний на смысловом уровне.

Без уточнения понятия смысла невозможно:

- решить задачу понимания информации, поступающей на вход ИС по разным каналам;

- разработать такой способ внутреннего представления знаний в памяти ИС, который был бы лишен каких бы то ни было особенностей, обусловленных не смыслом хранимых знаний, а предлагаемой формой их представления.

Любой язык есть форма представления всех тех знаний, которые могут быть представлены на этом языке. Смысл же представляемого знания есть такое его абстрактное представление, которое является инвариантом всего многообразия семантически эквивалентных форм представления этого знания в самых различных языках.

Смысловое представление информации является абстрактной знаковой конструкцией, обладающей следующими свойствами.

- 1) Среди знаков, входящих в состав указанной знаковой конструкции не должно существовать пар синонимичных знаков, т.е. знаков, имеющих один и тот же денотат, знаков, обозначающих одну и ту же сущность. Если в процессе обработки смыслового представления информации появляются синонимичные знаки, они должны быть «склеены».
- 2) Не допускается дублирование информации. При этом необходимо четко отличать семантическую эквивалентность знаний от их логической эквивалентности.
- 3) Среди знаков, входящих в состав указанной знаковой конструкции, не должно существовать омонимичных знаков. Из этого, в частности, следует, что в состав рассматриваемой знаковой конструкции не могут входить местоимения. Если в процессе анализа смыслового представления информации будут выявлены омонимичные знаки, то они должны быть «расклеены» на два или более неомонимичных знака.
- 4) Все знаки, входящие в состав указанной знаковой конструкции, не должны иметь внутренней структуры, анализ которой необходим для понимания этой знаковой конструкции. В смысловом представлении информации все знаки, входящие в состав знаковой конструкции, являются ее атомарными фрагментами.
- 5) Все атомарные фрагменты рассматриваемой знаковой конструкции являются знаками, т.е. семантически значимыми фрагментами.
- 6) В рамках построения смыслового представления информации не должны использоваться не только такие конструкции, как слова, термины (словосочетания), но и такие языковые приемы, как склонение, спряжение.
- 7) С синтаксической точки зрения все знаки, входящие в состав смыслового представления информации должны четко делиться на два вида:
 - знаки связей между описываемыми сущностями (связь считается одним из видов описываемых сущностей);
 - знаки сущностей, которые связями не являются. При этом между множеством знаков связей и множеством знаков иных сущностей задается несколько соответствий инцидентности, которые определяют то, какие сущности связываются каждой связью, и то, какие роли выполняют указанные сущности в рамках соответствующих им связей.

Важным достоинством смыслового представления информации является то, что в нем явно и четко задаются связи между описываемыми сущностями (в т.ч. и связи между связями) в виде связей между знаками этих сущностей: четко указывается семантический тип связи, а также указываются компоненты связи и роли этих компонентов в рамках этой связи. Все связи каждой описываемой сущности в смысловом представлении информации представлены только множеством знаков связей, инцидентных знаку указанной сущности, т.е. множеством связей, одним из компонентов которых является знак рассматриваемой сущности. Таким образом, любую знаковую конструкцию можно представить в семантически эквивалентном виде как множество знаков описываемых сущностей и множество знаков связей, связывающих эти описываемые сущности с другими сущностями. При этом на описываемые

сущности и на связи между ними не накладывается никаких ограничений. Описываемые сущности могут быть:

- материальными (физическими) и абстрактными (виртуальными) – числами, множествами, знаками каких-либо сущностей;
- действительно существующими и вымышленными;
- фиксированными (константными) и произвольными (переменными);
- связями между сущностями.

Нетрудно заметить, что знаковая конструкция, являющаяся смысловым представлением информации, в общем случае не может быть линейной, поскольку каждая описываемая сущность, являющаяся денотатом соответствующего знака, может быть связана неограниченным числом связей с другими сущностями, описываемыми в этой же знаковой конструкции. ГС, обладающие указанными выше семантическими свойствами, называют также *семантическими сетями* [11, 14]. Приведенная трактовка семантических сетей дает возможность в полной мере использовать теорию графов для исследований синтаксических свойств семантических сетей и для построения алгоритмов обработки семантических сетей.

Важнейшим достоинством семантических сетей является то, что в них достаточно просто и наглядно выглядит процедура навигационного поиска знаков описываемых сущностей на основе априорного знания семейства отношений, заданных на этих сущностях.

Семантические сети как модели представления знаний известны давно. Но, в отличие от фреймовых, продукционных и логических моделей, для семантических сетей не были разработаны достаточно удобные и практически используемые языки представления знаний, достаточно удобные языки программирования, специально ориентированные на обработку семантических сетей. И, как следствие этого, не были созданы широко используемые комплексные технологии проектирования ИС, в основе которых лежат семантические сети. Вот в чем же хороши семантические сети и в чем достоинство семантических моделей обработки информации:

- Представление знаний в виде семантических сетей позволяет существенно упростить процедуру интеграции знаний и свести эту процедуру к выявлению и склеиванию синонимичных элементов интегрируемых семантических сетей.
- База знаний ИС, представленная в виде корректно построенной семантической сети, полностью исключает дублирование информации.
- Представление знаний в виде семантических сетей позволяет существенно упростить процедуру ассоциативного доступа к различным видам фрагментов хранимой базы знаний, а также существенно расширить типологию запросов к базе знаний.
- Семантические модели обработки знаний не только хорошо приспособлены к поддержке параллельной асинхронной обработки информации, но и обеспечивают обмен информацией через общую графодинамическую память между различными одновременно протекающими процессами.
- С помощью семантических моделей представления и обработки знаний можно проинтерпретировать все известные виды моделей представления обработки знаний (фреймовые, продукционные, логические), а также все известные модели решения задач различного вида и все известные модели рассуждений. Это дает возможность рассматривать перечисленные модели не как альтернативные, а как дополняющие друг друга модели, которые могут сосуществовать в разных сочетаниях в разных ИС.

Семантическая модель обработки знаний представляет собой абстрактную многоагентную систему, состоящую из абстрактной семантической памяти, в которой хранятся семантические сети, и из множества агентов, ориентированных на обработку семантических сетей, хранимых в указанной семантической памяти.

Семантическую память можно трактовать как абстрактную семантическую модель памяти ИС.

Семейство абстрактных агентов, работающих над семантической памятью, вместе с этой семантической памятью можно трактовать как семантическую модель решения задач, используемую в соответствующей ИС, или как операционную семантику этой ИС. Подчеркнем, что семантическую модель обработки информации можно построить для любой компьютерной системы (как для ИС, так и для КС традиционного вида), обеспечивая, тем самым, семантическую совместимость (на абстрактном уровне) не только ИС, но и КС любого уровня интеллектуальности.

Всю семантическую сеть, хранимую в семантической памяти абстрактной логико-семантической модели ИС, будем называть абстрактной *семантической моделью базы знаний* этой ИС.

База знаний должна содержать в себе всю информацию, необходимую агентам, работающим над семантической памятью, для организации коллективной деятельности по решению задач, с которыми должна справляться ИС (сюда, в том числе, входит и описание блокировок, задаваемых разными процессами в семантической памяти).

Семантическая модель базы знаний ИС – это, образно говоря, формальная трактовка «смыслового пространства» в котором «живет» эта ИС, а, точнее, такого фрагмента указанного «смыслового пространства», который в текущий момент указанной ИС известен.

В целом логико-семантическая модель ИС включает в себя:

- семантическую модель базы знаний этой ИС;
- семантическую машину обработки знаний этой ИС, которая, в свою очередь, состоит из семантической памяти и коллектива агентов над семантической памятью.

2.3 Унификация абстрактного смыслового представления знаний различного вида

Предлагается определить структуру *унифицированных семантических сетей*, обеспечивающих представление и интеграцию различных видов знаний.

Это предполагает разработку соответствующего стандарта, выделяющего из всего многообразия абстрактных *языков семантических сетей* определенный базовый универсальный язык семантических сетей, который мы назвали *SC-кодом* (*Semantic Computer code*) [15, 16].

Основными направлениями перехода от семантических сетей общего вида к текстам *SC*-кода являются следующие.

- 1) Переход от семантических сетей, имеющих унарные и многокомпонентные (многоместные) связи, к семантическим сетям, имеющим только бинарные связи. Такие сети будем называть *бинарными семантическими сетями* [17, 18].
- 2) Минимизация алфавита, т.е. минимизация числа меток, используемых в семантических сетях.
- 3) Универсальность разрабатываемого языка, т.е. возможность представления любых знаний в виде текстов этого языка.

Тексты *SC*-кода будем называть *sc-текстами*. Знаки, входящие в состав *sc*-текстов, будем называть *sc-элементами*. Переход от введенного выше общего понятия семантической сети к унифицированным семантическим сетям (*sc*-текстам) рассмотрим как задание целого ряда ограничений на семантические сети общего вида. Такие ограничения не должны снижать семантической мощности языка семантических сетей, претендующего на универсальность.

Ограничение 1. Семантическая нормализация описываемых множеств. Если элемент семантической сети (*sc*-элемент) является знаком некоторого множества, то каждый элемент этого множества, представляющий собой сущность, не являющуюся знаком (*sc*-элементом),

заменяется на знак указанной сущности (на *sc*-элемент, обозначающий эту сущность). Таким образом, все описываемые *sc*-текстом множества становятся множеством знаков, а точнее, множеством *sc*-элементов (*sc*-множествами). Такие множества будем называть семантически нормализованными. Очевидно, любое множество может быть представлено в семантически нормализованном виде. Для описания связей между *sc*-множествами и элементами этих множеств вводятся связи принадлежности. Каждая из этих связей связывает знак некоторого *sc*-множества с одним из элементов этого множества, который всегда является знаком (*sc*-элементом) благодаря семантической нормализации *sc*-множества.

Ограничение 2. Каждая связь описываемых сущностей трактуется как множество, элементами которого являются описываемые сущности, связываемые этой связью. Семантическая нормализация каждого такого множества означает то, что все связи, входящие в состав семантической сети, будут связывать не сами описываемые сущности, а знаки этих сущностей за исключением случаев, когда указанные сущности уже являются знаками.

Ограничение 3. Если в состав семантической сети входит знак небинарной связи, то связи между этим знаком и компонентами обозначаемой им связи задаются не с помощью пар инцидентности (на синтаксическом уровне), а явно с помощью явно вводимых связей принадлежности. Это означает, что в текстах *SC*-кода все небинарные связи представляются с помощью бинарных (сводятся к бинарным).

Ограничение 4. Каждая семантическая сеть и, в частности, каждая унифицированная семантическая сеть (*sc*-текст), которая описывается и, соответственно, обозначается в другой семантической сети (метасети), трактуется как семантически нормализованное множество, элементами которого являются все те и только те знаки, которые входят в состав описываемой семантической сети.

Ограничение 5. В рамках *SC*-кода четко задается и минимизируется алфавит элементов семантических сетей, т.е. алфавит знаков, входящих в состав *sc*-текстов. Указанный алфавит представляет собой семейство классов *sc*-элементов, задаваемых с помощью меток. При этом каждой такой метке взаимно однозначно соответствует свой синтаксически задаваемый класс *sc*-элементов. Семейство таких классов включает в себя классы:

- *sc*-узлов;
- *sc*-ссылок на внешние информационные ресурсы;
- *sc*-ребер;
- *sc*-дуг общего вида;
- *sc*-дуг принадлежности или непринадлежности;
- стационарных константных *sc*-дуг принадлежности.

Ограничение 6. В рамках *SC*-кода четко задаются правила перехода *sc*-элементов из одного синтаксически задаваемого класса *sc*-элементов в другой при полном сохранении его семантики. Речь идет об изменении синтаксического типа *sc*-элемента при появлении определенного вида новой информации об этом *sc*-элементе. В результате такого изменения *sc*-элемент может перейти:

- из класса *sc*-узлов в класс *sc*-ссылок;
- из класса *sc*-узлов в класс *sc*-ребер;
- из класса *sc*-ребер в класс *sc*-дуг общего вида;
- из класса *sc*-дуг общего вида в класс *sc*-дуг принадлежности или непринадлежности;
- из класса *sc*-дуг принадлежности или непринадлежности в класс стационарных константных *sc*-дуг принадлежности.

Ограничение 7. В *SC*-коде сущности, обозначаемые *sc*-элементами, делятся на два вида:

- семантически нормализованные множества, элементами которых являются *sc*-элементы;
- «внешние» описываемые сущности, не являющиеся множествами.

Таким образом, каждый *sc*-элемент есть либо знак множества (множества, элементами которого являются *sc*-элементы), либо знак «внешней» описываемой сущности (сущности, которая множеством не является). Подчеркнем, что *sc*-элементов, являющихся знаками множеств *sc*-элементов, большинство, поскольку:

- каждая связь трактуется как множество *sc*-элементов, обозначающих связываемые сущности;
- каждая структура трактуется как множество всех *sc*-элементов, входящих в эту структуру;
- каждое понятие трактуется как множество *sc*-элементов, обозначающих сущности, являющиеся экземплярами этого понятия.

SC-код имеет четкую базовую теоретико-множественную семантическую интерпретацию всех его *sc*-элементов.

Все *sc*-элементы, кроме знаков связей принадлежности или непринадлежности, можно разбить на следующие уровни иерархии:

- первичные *sc*-элементы – знаки «внешних» сущностей;
- *sc*-элементы второго уровня – знаки множеств, элементами которых являются только первичные *sc*-элементы;
- *sc*-элементы третьего уровня – знаки множеств, среди элементов которых есть, по крайней мере, один *sc*-элемент второго уровня, но нет ни одного *sc*-элемента более высокого уровня;
- и так – до бесконечности.

Ограничение 8. Все *sc*-элементы имеют описанную ниже четкую базовую семантическую типологию:

***sc*-элемент**

\leq разбиение*:

- {
 - *sc*-константа
= *sc*-знак фиксированной (конкретной) сущности
 - *sc*-переменная
= *sc*-знак нефиксированной (произвольной) сущности}

\leq разбиение*:

- {
 - первичный *sc*-элемент
= *sc*-знак внешней описываемой сущности
 - *sc*-множество
= вторичный *sc*-элемент
= знак множества *sc*-элементов
= *sc*-метазнак
= *sc*-текст
= текст *SC*-кода
= знак *sc*-текста}

***sc*-множество**

\leq разбиение*:

- {
 - *sc*-связка
 - *sc*-структура}

= *sc*-знак множества *sc*-элементов, в состав которого входят *sc*-связки или *sc*-структуры, связывающие эти *sc*-элементы

- *sc*-класс
= *sc*-элемент, обозначающий класс *sc*-элементов
= *sc*-знак множества *sc*-элементов, эквивалентных в том или ином смысле

}

***sc*-связка**

\leq разбиение*:

{

- бинарная *sc*-связка
- небинарная *sc*-связка

}

\leq разбиение*:

{

- ориентированная *sc*-связка
- неориентированная *sc*-связка

}

бинарная *sc*-связка

квазибинарная *sc*-связка

\supset *sc*-связка принадлежности или непринадлежности

\leq разбиение*:

{

- бинарная неатомарная *sc*-связка
- *sc*-коннектор
= бинарная атомарная *sc*-связка
 \leq разбиение*:

{

- *sc*-ребро
- *sc*-дуга

}

}

***sc*-связка принадлежности или непринадлежности**

\leq разбиение*:

{

- *sc*-связка принадлежности
- *sc*-связка непринадлежности
- *sc*-связка нечеткой принадлежности

}

\leq разбиение*:

{

- стационарная *sc*-связка принадлежности или непринадлежности
- нестационарная *sc*-связка принадлежности или непринадлежности

}

***sc*-структура**

\leq разбиение*:

{

- тривиальная *sc*-структура
= *sc*-структура, не содержащая *sc*-связок между своими элементами

- *двухуровневая sc-структура*
= *sc-структура, содержащая первичные элементы (вершины) и связи, связывающие только первичные элементы*
- *иерархическая sc-структура*
= *sc-структура, среди элементов которой имеются связи или структуры, связывающие другие связи или структуры этой же структуры*

}

<= разбиение*:

{

- *связная sc-структура*
- *несвязная sc-структура*

}

sc-класс

<= разбиение*:

{

- *класс первичных sc-элементов*
- *sc-отношение*
= *класс sc-связок*
- *класс sc-структур*
- *класс sc-классов*
= *sc-метакласс*
- *класс sc-элементов разного структурного типа*

}

⊃ *sc-понятие*

= *sc-концепт*

= *ключевой класс sc-элементов*

= *sc-знак ключевого класса sc-элементов*

Достоинствами SC-кода являются следующие его свойства:

- все основные семантические связи между текстами (семантическая эквивалентность, семантическое включение, семантическое пересечение) в SC-коде становятся теоретико-множественными (равенство, включение, пересечение множеств);
- неограниченная возможность перехода от sc-текстов к sc-метатекстам, содержащим знаки описываемых sc-текстов;
- тексты SC-кода (в том числе и те тексты, sc-знаки которых явно вводятся в рамках соответствующих им метатекстов) могут быть иерархическими структурами, имеющими любое число уровней иерархии, поскольку sc-элемент может обозначать множество, состоящее из любых sc-элементов. В отличие от этого, например, такие традиционные структуры, как алгебраические системы, являются трехуровневыми:
 - на первом уровне – элементы носителя (основного множества) алгебраической системы;
 - на втором уровне – кортежи, элементами которых являются элементы носителя;
 - на третьем уровне – отношения, элементами которых являются указанные кортежи.

Формализация знаний на основе SC-кода предполагает теоретико-множественную интерпретацию всех sc-элементов, не являющихся знаками «внешних» сущностей. Все такие sc-элементы являются знаками множеств sc-элементов и необходимо, прежде всего, уточнять то, какие sc-элементы являются знаками этих множеств. При этом совсем не обязательно, чтобы все эти sc-элементы были представлены в текущем состоянии sc-памяти.

2.4 Унификация структуризации баз знаний

Понятие *смыслового пространства* и представление этого пространства в виде унифицированной семантической сети имеют ключевое значение для решения таких задач, как:

- обеспечение семантической совместимости баз знаний различных ИС;
- интеграция знаний различного вида;
- обеспечение компонентного проектирования баз знаний ИС путем использования ранее разработанных и апробированных знаний, заимствованных из смыслового пространства.
- Структуризация смыслового пространства задаётся:
- типологией знаний, входящих в состав смыслового пространства, путём явного введения понятий, которые обозначают различные виды знаний, представленных семантическими сетями, являющимися фрагментами смыслового пространства;
- семейством отношений, заданных на множестве выделенных знаний, являющихся фрагментами смыслового пространства.

Подчеркнём, что не существует ни одного вида знаний, который бы не входил в состав смыслового пространства. Это значит, что рассматриваемое нами смысловое пространство носит универсальный характер. Подчеркнём также, что структуризацию смыслового пространства удобнее всего проводить в рамках унифицированной модели (*SC*-модели) смыслового пространства, поскольку *SC*-код является весьма удобным средством перехода от представления знаний к представлению метазнаний.

Для структуризации смыслового пространства ключевую роль играют такие виды *знаний*, как *ПрО*, *онтология*, *семантическая окрестность*.

ПрО – это важнейший вид знаний, входящих в состав смыслового пространства. Каждый знак, входящий в состав базы знаний, должен входить в состав хотя бы одной *ПрО*, выполняя в ней ту или иную роль. Каждой *ПрО* можно поставить в соответствие:

- множество семантических окрестностей, описывающих объекты исследования этой *ПрО*;
- семейство различного вида онтологий, описывающих свойства понятий этой *ПрО*.

ПрО – структура, фокусирующая (локализирующая, концентрирующая) внимание только на заданном классе объектов исследования и только в заданном «ракурсе», который формально задаётся набором отношений, заданных на исследуемых объектах.

При накоплении больших объёмов знаний появляется огромное количество самых различных понятий (концептов) и возникает проблема структуризации и систематизации этого многообразия понятий. Для этого все накапливаемые знания делятся на связанные между собой *ПрО*, каждой из которых соответствует свой набор понятий, каждое из которых в рамках этой *ПрО* выполняет определенную роль (либо является классом объектов исследования, либо является отношением, заданным на множестве объектов исследования, и так далее).

ПрО – это структура, в состав которой входят:

- 1) основные исследуемые (описываемые) объекты – первичные и вторичные;
- 2) различные классы исследуемых объектов;
- 3) различные связки, компонентами которых являются исследуемые объекты (как первичные, так и вторичные);
- 4) различные классы указанных выше связок;
- 5) различные подструктуры *ПрО*;
- 6) различные классы объектов, не являющихся ни объектами исследования, ни указанными выше связками, но являющихся компонентами этих связок.

ПрО – это результат интеграции семантических окрестностей, описывающих все исследуемые сущности заданного класса и имеющих общий предмет исследования (то есть один и тот же набор отношений, которым должны принадлежать связки, входящие в состав интегрируемых семантических окрестностей).

Необходимо чётко отличать:

- 1) саму ПрО, которая практически всегда бесконечна;
- 2) тот фрагмент ПрО, который в текущий момент хранится в памяти и элементы которого явно связаны парами принадлежности со знаком соответствующей ПрО, подчеркнём, что указанный фрагмент ПрО всегда должен быть конечным множеством;
- 3) различного вида онтологии указанной ПрО.

Кроме общего понятия ПрО, введём уточнение этого понятия на основе SC-кода, то есть уточнение того, как ПрО будут представлены в рамках SC-модели смыслового пространства.

sc-модель ПрО

= sc-текст ПрО

= ПрО, представленная в SC-коде

= sc-граф ПрО

= уточнение понятия ПрО на основе SC-кода

= ПрО, представленная в рамках SC-модели смыслового пространства

\subset бесконечное sc-множество

\subset ПрО

\subset sc-знание

Если ПрО рассматривать как часть смыслового пространства, то её ролевая структура будет выглядеть существенно сложнее, чем просто класс объектов исследования с семейством заданных на нём отношений. Важно учитывать не только роли элементов в рамках самой ПрО, но и их соотношение со смысловым пространством. Так, например, важно знать, все ли элементы (экземпляры) каждого понятия (концепта) ПрО входят в состав этой ПрО, то есть являются её элементами.

ПрО – это бесконечная знаковая структура, в которой явно указываются роли тех знаков, обозначающие понятия (концепты), экземпляры (элементы) которых входят в состав этой же ПрО. Роль каждого понятия в рамках ПрО определяется по двум признакам:

- по уровню исследовательского внимания;
- по структурному уровню.

По уровню исследовательского внимания понятия в рамках ПрО могут выполнять следующие роли:

- быть максимальным классом исследуемых объектов в рамках данной ПрО;
- быть классом исследуемых объектов в рамках данной ПрО (не обязательно максимальным классом);
- быть исследуемым классом не исследуемых объектов (например, связей, структур или даже классов, состоящих из исследуемых объектов);
- быть неосновным исследуемым классом (неосновным понятием, вводимым в рамках данной ПрО);
- быть не исследуемым классом (классом дополнительных, вспомогательных сущностей, классом, который исследуется в другой ПрО).

По структурному уровню понятия, рассматриваемые в рамках ПрО, могут выполнять следующие роли:

- быть максимальным классом первичных элементов данной ПрО – этот класс также должен быть либо максимальным классом исследуемых объектов, либо подклассом этого класса;
- быть отношением данной ПрО, заданным над первичными или вторичными элементами этой ПрО, то есть являющимся классом связей, входящих в состав указанной ПрО;
- быть классом структур, целиком входящих в состав данной ПрО;
- быть классом классов, целиком (полностью) входящих в состав ПрО;

- быть классом сущностей, принадлежащих разным структурным уровням данной ПрО, – в каждый такой класс могут одновременно входить и первичные элементы, и связи, и структуры, и классы.

Кроме того, для каждого не исследуемого класса элементов ПрО должно быть дополнительно уточнено теоретико-множественное соотношение этого класса с множеством элементов, входящих в состав ПрО:

- быть не исследуемым классом, который является подмножеством данной ПрО;
- быть не исследуемым классом, который строго пересекается с данной ПрО.

Подчеркнём, что все исследуемые классы ПрО (в том числе и классы непосредственно исследуемых объектов) являются подмножествами этой ПрО.

Множество ПрО развивается на два класса:

- статические (стационарные) ПрО;
- динамические ПрО.

Кроме того, можно говорить о первичных и вторичных ПрО (по аналогии с первичными и вторичными *sc*-элементами). Можно также говорить о классе рефлексивных ПрО, в каждой из которых хотя бы некоторые её ключевые элементы (исследуемые понятия) либо она сама являются объектами исследования (экземплярами исследуемых классов) в этой же ПрО.

Но во всём бесконечном множестве ПрО особо следует выделить конечное число ПрО, которые назовём общими ПрО и которые являются основой для построения различного вида спецификаций всевозможных предметных областей, а также основой для построения самих ПрО.

К числу общих ПрО относятся:

- общая ПрО предметных областей;
- общая ПрО знаков (в том числе *sc*-элементов);
- общая ПрО множеств (в том числе *sc*-множеств);
- общая ПрО отношений;
- общая ПрО структур (в том числе *sc*-структур);
- общая ПрО чисел и числовых структур;
- общая ПрО параметрических моделей (как количественных, так и качественных);
- общая ПрО онтологий;
- общая ПрО семантических окрестностей;
- общая ПрО знаний (*sc*-знаний);

Онтология – вид знаний, каждое из которых является спецификацией (описанием свойств) соответствующей ПрО, ориентированной на описание свойств и взаимосвязей понятий, входящих в состав указанной ПрО.

Из общей интегрированной онтологии ПрО можно выделить следующие частные онтологии:

- структурные спецификации ПрО, в которых описываются роли понятий, входящих в состав ПрО, а также связи специфицируемых ПрО с другими ПрО;
- теоретико-множественные онтологии ПрО, в которых описываются теоретико-множественные связи между понятиями специфицируемых ПрО, включая связи между классами объектов исследования и доменами или областями определения отношений, входящих в состав специфицируемых ПрО;
- терминологические онтологии, описывающие всевозможные термины, соответствующие понятиям специфицируемых ПрО, а также правила построения терминов, которые соответствуют экземплярам указанных понятий;

- логические онтологии, представляющие собой формальные теории, высказывания которых интерпретируются на специфицируемых ПрО и описывают свойства понятий, входящих в состав этих ПрО;
- онтологии информационных программ и задач, решаемых в рамках специфицируемых ПрО.

Каждому виду онтологий соответствует своя общая ПрО, интегрирующая все онтологии этого вида:

- структурным спецификациям (онтологиям) ПрО соответствует общая ПрО предметных областей;
- теоретико-множественным онтологиям ПрО соответствует общая ПрО множеств;
- логическим онтологиям ПрО соответствует общая ПрО формальных теорий и высказываний;
- и так далее.

Всем общим ПрО (в том числе и тем, которые соответствуют различным видам онтологий) соответствуют свои онтологии, специфицирующие эти общие ПрО. Онтологии общих ПрО являются *онтологиями верхнего уровня*. К числу таких онтологий относятся:

- онтология общей ПрО знаков (в частности, *sc*-элементов);
- онтология общей ПрО терминов (в частности, идентификаторов *sc*-элементов);
- онтология общей ПрО информационных конструкций (в т.ч. файлов, хранящих эти конструкции);
- онтология общей ПрО множеств (в частности, *sc*-множеств);
- онтология общей ПрО структур (графовых структур, *sc*-структур);
- онтология общей ПрО предметных областей;
- онтология общей ПрО онтологий;
- онтология общей ПрО формальных теорий и высказываний;
- онтология общей ПрО семантических окрестностей;
- онтология общей ПрО знаний (в частности, *sc*-знаний).

Семантическая окрестность заданного знака (который будем называть ключевым элементом этой семантической окрестности) – это текст, описывающий некоторые свойства той сущности, которая обозначается указанным ключевым знаком.

Другими словами, семантическая окрестность – это некоторого вида спецификация указанной описываемой сущности.

Точно так же, как и для онтологий, можно говорить о многообразии видов семантических окрестностей, которые определенным образом коррелируют с типологией специфицируемых сущностей.

В частности, можно выделить следующие виды семантических окрестностей:

- определение понятия;
- пояснение понятия;
- примечание (описание различных свойств специфицируемой сущности);
- описание преимуществ (достоинств);
- описание недостатков;
- сравнительный анализ;
- типичная семантическая окрестность (пример типичного использования описываемого понятия);
- описание декомпозиции заданной сущности;
- теоретико-множественная семантическая окрестность;
- логическая семантическая окрестность (семейство высказываний, описывающих свойства данного понятия).

Расширение семантической окрестности (т.е. повышение уровня детализации заданного знака) может осуществляться как путем расширения числа указываемых свойств (характеристик) и связей заданного знака (назовем это увеличением валентности специфицируемого знака), так и путем включения в состав расширяемой семантической окрестности семантических окрестностей тех знаков, которые смежны специфицируемому (центральному) знаку заданной (расширяемой) семантической окрестности (назовем это увеличением радиуса заданной семантической окрестности в соответствующем направлении).

Теперь перейдем к рассмотрению отношений, заданных на множестве знаний.

Прежде всего, рассмотрим отношения, связывающие между собой ПрО. Важнейшим из них является отношение *частная ПрО* *, с помощью которого задается иерархия ПрО путем перехода от менее детального к более детальному рассмотрению соответствующих классов объектов исследования.

Частная ПрО *

= *быть частной ПрО **

= *ПрО, детализирующая описание одного из классов объектов исследования другой (более общей) ПрО**

= *ПрО, максимальный класс объектов исследования которой является подмножеством максимального класса объектов исследования другой (более общей) ПрО**

Кроме этого, ПрО могут быть связаны между собой следующим образом:

- ПрО могут иметь совпадающие максимальные классы объектов исследования, но разные предметы исследования, в частности, в одной ПрО могут рассматриваться внешние связи исследуемых объектов, а в другой – внутренняя структура этих объектов;
- максимальный класс объектов исследования одной ПрО может быть булеаном (семейством всевозможных подмножеств) максимального класса объектов исследования другой ПрО (например, ПрО *геометрических фигур* и ПрО *геометрических точек*);
- максимальный класс объектов исследования одной ПрО может быть семейством всевозможных структур, являющихся фрагментами другой ПрО (например, ПрО конфигураций геометрических фигур и ПрО геометрических фигур);
- понятия одной ПрО могут быть объектами исследования в другой ПрО;
- каждое не исследуемое понятие одной ПрО должно быть исследуемым понятием в некоторой другой ПрО;
- ПрО могут быть изоморфными и гомоморфными.

Связь между ПрО и ее онтологией задается отношением *быть онтологией**.

2.5 Ориентация на графовые языки программирования

Для описания способов решения задач и поведения агентов над общей графодинамической памятью предлагается использовать *графовые языки программирования*, которые ориентированы на обработку унифицированных семантических сетей и программы которых сами являются унифицированными семантическими сетями.

Если все используемые в ИС графовые языки программирования привести к общему унифицированному стандарту – к *SC*-коду (это требует представления в виде *sc*-текстов не только самих программ, но и обрабатываемых ими данных), то можно достаточно эффективно решать проблему формализации семантической совместимости программ, написанных не только на одном, но и на разных языках программирования.

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

Важнейшей особенностью всех этих языков является использование ассоциативного доступа к обрабатываемым фрагментам хранимого в графодинамической памяти *sc*-текста. Операционная семантика каждого такого графового языка программирования (точнее, *sc*-языка программирования) задается коллективом агентов над общей графодинамической памятью, которые обеспечивают 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.

2.6 Унификация формального описания поведения агентов обработки информации в графодинамической семантической памяти

Из всех используемых в интеллектуальной системе графовых языков программирования (а, точнее, из всех *sc*-языков программирования) предлагается выделить базовый графовый язык программирования, ориентированный на описание агентов, работающих над общей графодинамической памятью, в которой хранятся и обрабатываются унифицированные семантические сети.

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 нем программы будем называть *scp-программами*.

Перечислим основные особенности языка *SCP*:

- язык *SCP* относится к классу графовых языков программирования;
- язык *SCP* ориентирован на обработку унифицированных семантических сетей (*sc*-текстов), хранимых в семантической памяти;
- программы языка *SCP* представляются также в виде унифицированных семантических сетей (*sc*-текстов), т. е. язык *SCP* принадлежит классу *sc*-языков;
- язык *SCP* ориентирован на описание параллельной асинхронной обработки *sc*-текстов, хранимых в семантической памяти;
- язык *SCP* использует ассоциативный доступ к фрагментам обрабатываемых *sc*-текстов;
- язык *SCP* является процедурным языком программирования низкого уровня, предназначенным для описания поведения агентов, работающих над семантической памятью;
- уникальной особенностью языка *SCP* является то, что на нем можно писать *реконфигурируемые программы*, т. е. программы, которые в процессе своего выполнения могут изменять сами себя (удалять или порождать операторы, корректировать порядок их выполнения и т. п.). Такая особенность языка *SCP* обусловлена не только тем, что *scp*-программы и обрабатываемые ими данные хранятся в общей памяти, но и тем, что они принадлежат одному и тому же, базовому языку (*SC*-коду), имеющему четко заданную семантическую интерпретацию.

Более подробно графовый язык программирования *SCP* и технология проектирования *scp*-программ рассмотрены в работах [19, 20].

Заключение

В работе предложены принципы построения технологии проектирования ИС, основанных на онтологиях и семантическом представлении знаний.

Предлагаемый в работе подход к созданию средств построения ИС нового поколения заключается в ориентации на графодинамическую парадигму обработки информации, в основе которой лежит понятие графодинамической (нелинейной структурно перестраиваемой) памяти, в которой обработка информации сводится не только к изменению состояния элементов памяти, но и к изменению конфигурации связей между ними. Выбор графодинамической парадигмы обусловлен тем, что графодинамические модели обработки информации по сравнению с другими моделями имеют ряд преимуществ:

- поддерживают развитые формы ассоциативного доступа;
- поддерживают параллельную реализацию операций;
- поддерживают асинхронную реализацию операций;
- легко интегрируются;
- легко интерпретируются друг другом;
- легко поддерживают работу со сложно структурированной информацией любого уровня сложности;
- поддерживают достаточно простой переход от информации к метаинформации и согласованную обработку информации и метаинформации;
- поддерживают гибкие (открытые, легко расширяемые, модифицируемые) и мощные стратегии и механизмы решения трудно формализуемых задач.

Список источников

- [1] **Попков, В.К.** Гиперсети и их характеристики связности / В.К. Попков // Исследования по прикладной теории графов. - Новосибирск: Наука, 1986. - С. 25-58.
- [2] **Колмогоров, А.Н.** К определению алгоритма / А.Н. Колмогоров // Успехи математических наук. - 1958. - Т.13. - №4(82). - С. 3-28.
- [3] **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.
- [4] **Петров, С.В.** Графовые грамматики и автоматы (обзор) / С.В. Петров // Автоматика и телемеханика. - 1978. - №7. - С. 116-136.
- [5] **Касьянов, В.Н.** Графы в программировании: обработка, визуализация и применение / В.Н. Касьянов, В.А. Евстигнеев. – СПб.: BHV -Петербург, 2003. – 1104 с.
- [6] **Котов, В.Е.** Асинхронные вычислительные процессы над общей памятью / В.Е. Котов, А.С. Нариньяни // Кибернетика. - 1966. – №3. - С. 64-71.
- [7] **Борщев, В.Б.** Схемы на клубных системах и вегетативная машина / В.Б. Борщев // Семиотика и информатика. - 1983. - Вып. 22. - с. 3-44.
- [8] **Кузнецов, И.П.** Семантические представления. / И.П. Кузнецов. – М: Наука, 1986.- 268 с.
- [9] **Лозовский, В.С.** Семантические сети / В.С. Лозовский // Представление знаний в человеко-машинных и робототехнических системах. – М.: ВИНТИ, 1984. – С. 84-121.
- [10] **Плесневич, Г.С.** Представление знаний в ассоциативных сетях / Г.С. Плесневич // Изв. АН СССР. Техн. кибернет. - 1982. – №5. - с.6-22.
- [11] **Скоруходько, Э.Ф.** Семантические сети и автоматическая обработка текста / Э.Ф. Скоруходько. – К.: Наук. думка, 1983. – 220 с.
- [12] **Шенк, Р.** Обработка концептуальной информации / Р. Шенк. - Москва: Энергия, 1980. - 360 с.
- [13] **Sowa, J.F.**. Conceptual Graphs / J.F. Sowa, F. van Harmelen, V. Lifschitz, B. Porter // Handbook of Knowledge Representation. - Elsevier, 2008. - P. 213-237.
- [14] **Болотова, Л.С.** Системы искусственного интеллекта: модели и технологии, основанные на знаниях / Л.С. Болотова. – М.: Финансы и статистика, 2012. – 664 с.
- [15] **Голенков, В.В.** Представление и обработка знаний в графодинамических ассоциативных машинах / В.В. Голенков, О.Е. Елисеева, В.П. Ивашенко и др. – Мн. : БГУИР, 2001. – 412 с.
- [16] **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.

- [17] **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. – С.135-146.
- [18] **Карабеков, Б.А.** Система «Бинарная Модель Знаний» как инструмент для концептуального моделирования бизнес-процессов / Б.А. Карабеков // Одиннадцатая национальная конференция по искусственному интеллекту: Труды конференции. Т.2. – М.: Физматлит, 2008. – С. 282-291.
- [19] **Голенков, В.В.** Программирование в ассоциативных машинах/ В.В. Голенков и др. – Мн.: БГУИР, 2001. – 276 с.
- [20] **Гулякина, Н.А.** Языки и технологии программирования, ориентированные на обработку семантических сетей / Н.А. Гулякина О.В. Пивоварчик, Д.А. Лазуркин // Материалы международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем» – Мн.: БГУИР, 2012. – С. 221-228.
-

PROJECT OF OPEN SEMANTIC TECHNOLOGY OF THE COMPONENTAL DESIGN OF INTELLIGENT SYSTEMS. PART 1: THE PRINCIPLES OF CREATION

Vladimir Golenkov¹, Natalia Guliakina²

Belarusian State University of Informatics and Radio-electronics, Minsk, Belarus

¹golen@bsuir.by, ²guliakina@bsuir.by

Abstract

This paper describes an open source project aimed at the creation and development of component technology of intelligent systems design. In the first part of the article contains a description of the developed principles of the technology of intelligent systems design based on ontologies and semantic representation of knowledge. The use of the accumulated experience of modern information technology, application graph dynamic models of information processing is demonstrated. In particular, it is proposed to focus on parallelism and asynchrony, on semantic model knowledge representation and processing, on the graph programming languages. The authors recommend focusing on the unification of abstract semantic knowledge representation, unification structuring knowledge bases and formal description of the behavior of agents in information processing graph dynamic semantic memory.

Key words: *intelligent system, design technology, semantic network, SC-code domain ontology, the graph programming language, the language of the SCP.*

Reference

- [1] **Popkov, V.K.** Giperseti i ikh kharakteristiki svyaznosti [Hypernetworks and their coherence characteristics] / V.K. Popkov // Issledovaniya po prikladnoj teorii grafov [Research in the applied graph theory]. – Novosibirsk: Nauka publ., 1986. - pp. 25-58. (In Russian).
- [2] **Kolmogorov, A.N.** K opredeleniyu algoritma [Towards the definition of algorithm] / A.N. Kolmogorov // Uspekhi matematicheskikh nauk [Advances of mathematical science]. - 1958. - Vol.13. - Issue 4(82). - pp. 3-28. (In Russian).
- [3] **Ajzerman, M.A.** Dinamicheskij podkhod k analizu struktur, opisyaemykh grafami (osnovy grafodinamiki) [Dynamic approach to the analysis of structures which can be described by graphs (basic graph dynamics)] / M.A. Ajzerman, L.A. Gusev, S.V. Petrov, I.M. Smirnova, L.A. Tenenbaum // Issledovaniya po teorii struktur. – Moscow: Nauka publ., 1988. - pp. 5-76. (In Russian).
- [4] **Petrov, S.V.** Grafovye grammatiki i avtomaty (obzor) [Graph grammatiks and automatons (review)] / S.V. Petrov // Avtomatika i telemekhanika [Automation and telemechanics]. - 1978. - Issue 7. - pp. 116-136. (In Russian).
- [5] **Kas'yanov, V.N.** Grafy v programmirovanii: obrabotka, vizualizatsiya i primenenie. [Graphs in programming: processing, visualization and applications] / V.N. Kas'yanov, V.A. Evstigneev. - St-Petersburg: BHV-Petersburg publ., 2003. - 1104 p. (In Russian).
- [6] **Kotov, V.E.** Asinkhronnye vychislitel'nye protsessy nad obshhej pamyat'yu. [Asynchronous computing processes over shared memory] / V.E. Kotov, A.S. Narin'yani // Kibernetika [Cybernetics]. - 1966. - Issue 3. - pp. 64-71. (In Russian).

- [7] **Borshhev, V.B.** Skhemy na klubnykh sistemakh i vegetativnaya mashina [Scheme for club systems and the vegetative machine] / V.B. Borshhev // Semiotika i informatika [Semiotics and informatics]. - 1983. - Issue 22. - pp. 3-44. (In Russian).
- [8] **Kuznetsov, I.P.** Semanticheskie predstavleniya [Semantic representations] / I.P. Kuznetsov. - Moscow. Nauka publ., 1986. - 268 p. (In Russian).
- [9] **Lozovskij, V.S.** Semanticheskie seti [Semantic networks] / V.S. Lozovskij // Predstavlenie znaniy v chelovekomashinnykh i robototekhnicheskikh sistemakh [Knowledge representation in man-machine systems and robotic]. Moscow: VINITI publ., 1984. - pp. 84-121. (In Russian).
- [10] **Plesnevich, G.S.** Predstavlenie znaniy v assotsiativnykh setyakh [Knowledge representation in associative networks]. / G.S. Plesnevich // Isv. AN SSSR Tehn. kibernet [Tidings of USSR Academy of science. Technical cybernetics]. - 1982. - Issue 5. - pp. 6-22. (In Russian).
- [11] **Skorokhod'ko, E.F.** Semanticheskie seti i avtomaticheskaya obrabotka teksta [Semantic networks and automatic text processing] / E.F. Skorokhod'ko. - Kiev: Nauk. publ., 1983. - 220 p. (In Russian).
- [12] **Shenk, R.** Obrabotka kontseptual'noj informatsii [Conceptual information processing] / R. Shenk. - Moscow. Energiya publ., 1980. - 360 p. (In Russian).
- [13] **Sowa, J.F.** Conceptual Graphs / J.F. Sowa, F. van Harmelen, V. Lifschitz, B. Porter // Handbook of Knowledge Representation. - Elsevier, 2008. - pp. 213-237.
- [14] **Bolotova, L.S.** Sistemy iskusstvennogo intellekta: modeli i tekhnologii, osnovannye na znaniyakh [Artificial intelligence systems: models and technologies] / L.S. Bolotova. - Moscow. Finance and Statistics, 2012. - 664 p. (In Russian).
- [15] **Golenkov, V.V.** Predstavlenie i obrabotka znaniy v grafodinamicheskikh assotsiativnykh mashinakh [Representation and processing of knowledge in graphdynamic systems] / V.V. Golenkov et al. - Minsk. BGYIR, 2001. - 412 p. (In Russian).
- [16] **Golenkov, V.V.** Grafodinamicheskie assotsiativnye modeli i sredstva parallel'noj obrabotki informatsii v sistemakh iskusstvennogo intellekta [Graphdynamic associative models and means of parallel processing of information in artificial intelligence systems] / V.V. Golenkov, N.A. Gulyakina // BGYIR proceedings. - 2004. - Issue 1(5). - pp. 92-101. (In Russian).
- [17] **Plesnevich, G.S.** Binarnye modeli znaniy [Binary knowledge models] / G.S. Plesnevich // Trudy Mezhdunarodnykh nauchno-tekhnicheskikh konferentsij «Intellektual'nye sistemy» (AIS'08) [Proceedings of AIS'08]. Scientific issue in 4 volumes. Vol.2. - Moscow: Fizmatlit publ., 2008. - pp.135-146. (In Russian).
- [18] **Karabekov, B.A.** Sistema «Binarnaya Model' Znaniy» kak instrument dlya kontseptual'nogo modelirovaniya biznes-protsessov [System "of the binary model of Knowledge" as a tool for conceptual modeling of business processes] / **Karabekov B.A.** // Odinnadtsataya natsional'naya konferentsiya po iskusstvennomu intellektu : Trudy konferentsii [Eleventh National Conference on Artificial Intelligence: Proceedings]. Vol.2. - Moscow, Fizmatlit publ., 2008. - pp. 282-291. (In Russian).
- [19] **Golenkov, V.V.** Programmirovaniye v assotsiativnykh mashinakh [Programming in associative machines] / V.V. Golenkov et al. - Minsk. BGUIR publ., 2001. - 276 p. (In Russian).
- [20] **Gulyakina N.A.**, Yazyki i tekhnologii programmirovaniya, orientirovannyye na obrabotku semanticheskikh setej [Programming languages and technologies focused on processing of semantic networks] / N.A. Gulyakina, O.V. Pivovarchik, D.A. Lazurkin // Materialy mezhdunarodnoj nauchno-tekhnicheskoj konferentsii «Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh sistem» [Proc. of the International Scientific and Technical Conference "Open semantic technology of intelligent systems]. - Minsk: BGUIR publ., 2012. - pp. 221-228. (In Russian).

Сведения об авторах



Голенков Владимир Васильевич, 1949 г. рождения. В 1971 г. с отличием окончил физический факультет Белорусского государственного университета, д.т.н. (1996), профессор, заведующий кафедрой интеллектуальных информационных технологий Белорусского государственного университета информатики и радиоэлектроники, инициатор международной научно-практической конференции «Открытые семантические технологии проектирования интеллектуальных систем», член Российской ассоциации искусственного интеллекта, член редколлегии журналов «Программные продукты и системы», «Онтология проектирования», «Элек-

троники ИНФО», «Речевые технологии». В списке научных трудов более 100 работ в области семантических технологий.

Vladimir Golenkov was born in 1949. In 1971 he graduated from the faculty of physics of Belarusian State University with honors diploma, Dr. of science (1996), professor. The head of the department of Intelligent information systems of Belarusian State University of Informatics and Radio-electronics, initiator of the international science and practical conference «Open Semantic technologies for Intelligent Systems» (OSTIS), member of Russian association of artificial intelligence, member of the editorial board of such journals as «Program products and systems», «Ontology of designing», «Electronics-INFO», «Speech technologies». The list of his published works consists of more than 100 articles in the sphere of semantic approach in the intelligence systems design.



Гулякина Наталья Анатольевна, 1952 г. рождения. В 1974 г. окончила факультет прикладной математики Белорусского государственного университета, к.ф.-м.н, доцент, заместитель заведующего кафедрой интеллектуальных информационных технологий Белорусского государственного университета информатики и радиоэлектроники, член программного комитета конференции «Открытые семантические технологии проектирования интеллектуальных систем», автор ряда публикаций по вопросам интеллектуальных информационных технологий, автор семи учебно-методических пособий.

Natalia Guliakina was born in 1952. In 1971 she graduated from the faculty of applied mathematics of Belarusian State University, Ph.D., and assistant professor. Deputy head of the department of Intelligent information systems of Belarusian State University of Informatics and Radio-electronics, a member of Program Committee of the international conference «Open Semantic technologies for Intelligent Systems» (OSTIS), author of several publications on artificial intelligence systems design, author of seven educational toolkits.