



# OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

## СЕМАНТИЧЕСКАЯ ТЕХНОЛОГИЯ КОМПОНЕНТНОГО ПРОЕКТИРОВАНИЯ СИСТЕМ, УПРАВЛЯЕМЫХ ЗНАНИЯМИ

Голенков В.В., Гулякина Н.А.

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

**golen@bsuir.by**  
**guliakina@bsuir.by**

В работе рассматривается итог пятилетнего развития Проекта OSTIS, направленного на создание Открытой семантической технологии проектирования интеллектуальных систем. В основе указанной технологии лежит представление знаний в виде унифицированных семантических сетей с теоретико-множественной интерпретацией. В работе рассматриваются классы систем, основанных на знаниях, и систем, управляемых знаниями.

**Ключевые слова:** системы, управляемые знаниями, компонентное проектирование, технология проектирования, представление знаний.

### Введение

Данная работа подводит итог пятилетнего развития Проекта OSTIS (Open Semantic Technology for Intelligent Systems), целями которого являются:

- Разработка Технологии OSTIS, позволяющей быстро и качественно разрабатывать семантически совместимые компьютерные системы, управляемые знаниями, и способные создавать временные коллективы компьютерных систем для распределенного решения сложных задач.
- Определение круга актуальных приложений таких систем.
- Использование Технологии OSTIS как основы практико-ориентированной подготовки студентов и магистрантов.
- Уточнение принципов взаимодействия науки, образования, инженерии и бизнеса в развитии рынка компьютерных систем, управляемых знаниями.

Под семантической технологией компонентного проектирования компьютерных систем мы понимаем комплекс согласованных частных технологий, обеспечивающий целостное проектирование компьютерных систем, управляемых знаниями, и включающий:

- семантическую технологию компонентного проектирования баз знаний;

- семантическую технологию компонентного проектирования программ, ориентированных на обработку баз знаний;
- семантическую технологию компонентного проектирования решателей задач в компьютерных системах, управляемых знаниями;
- семантическую технологию компонентного проектирования пользовательских интерфейсов, обеспечивающих общение компьютерных систем, управляемых знаниями, с пользователями (в т.ч. и на естественных языках);
- семантическую технологию компонентного проектирования подсистем мультисенсорного восприятия и анализа информации о внешней среде компьютерных систем, управляемых знаниями;
- семантическую технологию компонентного проектирования подсистем координируемого воздействия на внешнюю среду компьютерных систем, управляемых знаниями.

### 1. Компьютерные системы, управляемые знаниями

Компьютерная система, управляемая знаниями, – система, в основе которой лежит представленная унифицированным образом база знаний, содержащая в систематизированном виде всю информацию, используемую этой системой. Таким образом, в компьютерных системах,

управляемых знаниями, вся используемая и обрабатываемая информация представляется в виде семантически структурированной целостной базы знаний, отражающей на смысловом уровне полную картину Мира, в котором "живет" эта компьютерная система. Решатель задач компьютерной системы, управляемой знаниями, (машина обработки знаний) представляет собой коллектив самостоятельных агентов, взаимодействующих между собой только через базу знаний. Некоторые агенты могут взаимодействовать и с внешней средой с помощью рецепторных и эффекторных средств. Указанный характер взаимодействия рассматриваемых агентов позволяет трактовать систему, управляемую знаниями, как систему, управляемую своей базой знаний. Память компьютерных систем, управляемых знаниями, носит структурно перестраиваемый ассоциативный характер и в перспективе может быть реализована не только программно (как в настоящий момент), но и аппаратно. Для реализации обработки знаний в такой памяти (в частности, для описания поведения агентов) необходимы языки программирования, ориентированные на обработку знаний в структурно перестраиваемой ассоциативной памяти, программы которых должны рассматриваться как частный вид знаний, хранимых в составе единой базы знаний. Интерпретатор одного из таких языков программирования (базового языка программирования) при аппаратной реализации структурно перестраиваемой ассоциативной памяти должен быть реализован тоже аппаратно. Существенно то, что при переходе на любую новую платформу реализации структурно перестраиваемой памяти и интерпретатора соответствующего базового языка программирования (в том числе, и при переходе на аппаратную их реализацию) никаких изменений в базу знаний и в решатель задач конкретной системы, управляемой знаниями, вносить не потребуется.

#### **система, управляемая знаниями**

- = система, управляемая своей базой знаний
- = система с активной базой знаний, ситуации и события в которой инициируют процессы ее обработки, выполняемые коллективом агентов
- = система, в основе которой лежит база знаний, управляющая деятельностью работающих над ней агентов
- = многоагентная компьютерная система, агенты которой управляются ее базой знаний
- = многоагентная компьютерная система, агенты которой взаимодействуют только через ее базу знаний
- = многоагентная компьютерная система, агенты которой обмениваются информацией только через ее базу знаний
- = многоагентная компьютерная система, агенты которой инициируются событиями или

*ситуациями либо в базе знаний, либо во внешней среде*

- = система, в основе которой лежит база знаний и коллектив работающих над ней и взаимодействующих через нее агентов
- = система, основанная на семантически структурированных данных и метаданных, управляющих процессом их обработки
- = система, основанная на связанных, формально представленных, хорошо структурированных данных и метаданных, управляющих процессом их обработки
- = система, в основе которой лежит база знаний, содержащая всю используемую ею информацию и организующая взаимодействие работающих над ней агентов
- = система, архитектура которой состоит из базы знаний и коллектива агентов, работающих над этой базой знаний и взаимодействующих только через нее
- = система, в которой хранящиеся в ней знания полностью управляют ее деятельностью, инициируя активность различных агентов, осуществляющих обработку этих знаний
- = система, в которой управляющими воздействиями являются события или ситуации, которые возникают в обрабатываемой базе знаний и которые инициируют деятельность соответствующих агентов, работающих над этой базой знаний (каждый агент реагирует на свой класс ситуаций или событий)
- = система, обрабатываемая база знаний которой выполняет роль среды, управляющей деятельностью и взаимодействием агентов, работающих над этой базой знаний
- = система, обрабатываемая база знаний которой выполняет роль реконфигурируемого (структурно перестраиваемого) коммутатора, обеспечивающего взаимодействие агентов, осуществляющих реконфигурацию этого коммутатора
- = система, состоящая из изменяемой (обрабатываемой) базы знаний и коллектива самостоятельных агентов, осуществляющих обработку этой базы знаний и взаимодействующих между собой только через указанную базу знаний

Не следует отождествлять понятие системы, управляемой знаниями, и понятие системы, основанной на знаниях. Понятие системы, основанной на знаниях, является более общим, т.к. база знаний здесь не обязательно должна быть активной и семантически структурированной. Это означает, что управление в системе, основанной на знаниях, может осуществляться программой, а не ситуациями и событиями в обрабатываемой базе знаний. А это, в свою очередь, означает, что обработка информации в системе, основанной на знаниях, не обязательно осуществляется коллективом агентов, работающих над базой знаний и взаимодействующих через нее. Таким

образом, системы, управляемые знаниями, следует считать важным этапом эволюции систем, основанных на знаниях, который обеспечивает существенно более высокий уровень их гибкости.

Не следует также путать понятие системы, управляемой знаниями, и понятие системы управления знаниями [Гаврилова и др., 2007], [Тузовский и др., 2005]. Система управления знаниями является частного вида системой, основанной на знаниях, которая представляет собой корпоративную систему, автоматизирующую процессы создания, распространения и использования информации внутри некоторого предприятия [Джанетто и др., 2005].

Понятие системы, управляемой знаниями, не следует путать с понятием интеллектуальной системы, т.е. системы, способной решать интеллектуальные задачи. Хотя очевидно, что, если интеллектуальную систему построить как систему, управляемую знаниями, то она приобретет способность существенно быстрее развиваться.

Переход от традиционного построения компьютерных систем к системам, построенным по архитектуре систем, управляемых знаниями, представляет собой переход к принципиально иным методологическим принципам построения компьютерных систем. В основе традиционного подхода к построению компьютерных систем лежит доминирование программ, в результате чего структуризация обрабатываемых данных "подстраивается" под программы. Поэтому при заранее не предусмотренном расширении функциональных возможностей системы нередко возникает необходимость либо в корректировке структур данных и, следовательно, корректировке уже написанных программ, либо в эклектичных, специализированных надстройках над изначально продуманными структурами данных, что приводит к усложнению понимания (разработчиками) смысла такой структуризации обрабатываемых данных. Основой построения компьютерных систем, управляемых знаниями, лежит доминирование структуризации обрабатываемой информации над программами. При этом структуризация обрабатываемой информации осуществляется независимо от программ и определяется исключительно смыслом этой информации.

Для того, чтобы в компьютерной системе в полной мере обеспечить управление знаниями, в состав базы знаний системы, в частности, должны входить:

- формулировки задач, решаемых компьютерной системой;
- планы решения этих задач, представляющие собой описание различных вариантов сведения решаемых задач к иерархической системе подзадач;

- указание активных задач, решаемых в текущий момент (как атомарных задач, так и задач, сводимых к подзадачам).

Переход современных компьютерных систем к системам, основанным на знаниях, является важнейшей тенденцией перехода к следующему поколению компьютерных систем. При этом существенно подчеркнуть то, что любую современную компьютерную систему можно реализовать по архитектуре систем, управляемых знаниями. Такая трансформация современных компьютерных систем существенно повысит их качество и конкурентоспособность. В результате трансформации современных компьютерных систем в функционально эквивалентные системы, управляемые знаниями, мы получим различные, но семантически совместимые системы, отличающиеся структурной сложностью баз знаний, функциональной сложностью агентов обработки знаний, и способные к формированию временных коллективов компьютерных систем при необходимости коллективного решения сложных задач.

Таким образом, технология проектирования компьютерных систем, управляемых знаниями – это технология, ориентированная на разработку систем, которые не обязательно должны уметь решать интеллектуальные задачи. Использование такой технологии для проектирования любых компьютерных систем на первый взгляд кажется, по сравнению с традиционными технологиями, более сложным вариантом их разработки, требующим более глубокого проникновения в семантику обрабатываемой информации и в суть реализуемых информационных процессов. Но логико-семантическая унификация архитектур компьютерных систем на основе унифицированного семантического представления обрабатываемой информации переводит эти системы на принципиально новый эволюционный уровень и создает условия для разработки более эффективных технологий их проектирования.

Актуальность систем, управляемых знаниями, обусловлена тем, что в настоящее время все большую важность приобретают задачи, для решения которых априори трудно локализовать информацию, достаточную для их решения. В ходе решения таких задач может потребоваться заранее непредсказуемая информация, хранимая в памяти системы, удовлетворяющая тем или иным требованиям. Образно говоря, исходными данными таких задач является вся информация, хранимая в памяти системы. Примерами указанных задач являются:

- информационно-поисковые задачи;
- задачи интеграции информации;
- задачи, для решения которых априори не известны способы (программы) их решения и эти способы надо найти, скомбинировать, скомбинировать из множества хранимых в памяти программ;

- задачи анализа качества всей совокупности хранимой в памяти информации;
- задачи логического вывода.

Традиционные компьютерные системы в основном ориентированы на решение задач, исходные данные которых локализованы (полностью заданы). В ходе решения таких задач потребность в дополнительной информации отсутствует. Для задач такого вида могут существовать специализированные решатели, которые у себя могут использовать удобные для них специализированные способы представления и хранения обрабатываемых ими данных. Для того, чтобы такие специализированные решатели интегрировать в состав компьютерной системы, достаточно организовать обмен конечного количества исходных (передаваемых) и возвращаемых параметров с априори известной их семантической интерпретацией или обмен конечных информационных конструкций, представленных в оговоренном формате.

К сожалению, для задач первого вида такой принцип взаимодействий компьютерной системы с решателями задач невозможен. Обмен информацией между решателями задач первого вида возможен только через общую для них память на основе унифицированного представления обрабатываемой ими информации. Более того, чтобы достаточно быстро можно было найти информацию, востребованную в ходе решения задачи, вся хранимая в системе информация должна быть структурирована, систематизирована, т.е. преобразована в целостный информационный контент.

Актуальность трансформации традиционных компьютерных систем в функционально эквивалентные системы, управляемые знаниями, обусловлена также тем, что переход на унифицированное представление различного вида знаний создает реальные условия для устранения дублирования информации в различных системах, т.е. для ликвидации "вавилонского столпотворения", обусловленного многообразием форм представления одной и той же информации в разных системах. Это, в свою очередь, дает возможность унифицировать и средства обработки знаний.

Многообразие форм решения одних и тех же задач в разных системах, обусловленное многообразием форм представления обрабатываемой информации, является важнейшим препятствием совершенствования технологий проектирования компьютерных систем.

Методика компонентного проектирования, основанная на постоянно расширяемых библиотеках многократно используемых компонентов, является для компьютерных систем, управляемых знаниями, значительно более широкого применима, чем для традиционных

компьютерных систем. Это обусловлено следующими причинами:

- Унификация логико-семантической архитектуры компьютерных систем, управляемых знаниями, и унификация представления знаний дает возможность существенно расширить библиотеки многократно используемых (reusable) компонентов. Проектирование компьютерных систем становится более "крупноблочным" и следовательно, значительно более быстрым. В частности, компьютерные системы, управляемые знаниями, дают возможность создавать библиотеки семантически целостных и достаточно крупных многократно используемых компонентов баз знаний. Отсутствие семантически унифицированного представления обрабатываемой информации в традиционных компьютерных системах такой возможности не предоставляет.
- Агенты, работающие над базой знаний компьютерных систем, управляемых знаниями, часто имеют предметно независимый характер, что делает их многократно используемыми компонентами в самых различных компьютерных системах, управляемых знаниями.
- Компьютерные системы, управляемые знаниями, могут легко интегрировать новые полезные для них компоненты, вводимые в состав постоянно расширяемых библиотек многократно используемых компонентов. Таким образом, каждая компьютерная система, управляемая знаниями, может эволюционировать в ходе своей эксплуатации как в результате деятельности своих разработчиков, которые должны постоянно ее совершенствовать, так и на основе постоянно расширяемых библиотек многократно используемых компонентов.

Актуальность трансформации традиционных компьютерных систем в системы, управляемые знаниями, обусловлена также тем, что для обеспечения их эффективной эксплуатации и совершенствования они должны в своей памяти хранить хорошо систематизированную и формализованную документацию, которая должна стать важной частью информационного контента компьютерной системы. Компьютерные системы должны становиться более понятными, прозрачными, комфортными для человека – как для конечного (в том числе, начинающего) пользователя, так и для разработчика (в том числе, начинающего). Каждый пользователь должен иметь возможность получить полную информацию:

- о структуре и возможностях текущей версии системы (в т.ч. о ее пользовательском интерфейсе);
- о методике эффективной эксплуатации системы;

- об истории совершенствования системы (о прошлых ее версиях);
- о планах совершенствования системы;
- о команде разработчиков и распределении проектных задач и обязанностей;
- о методике и организации коллективной разработки (совершенствовании) системы.

Качество каждой компьютерной системы можно оценивать в трех аспектах:

- как уровень приобретенных ею знаний и навыков;
- как скорость повышения уровня приобретаемых знаний и навыков (имеется ввиду максимально возможная скорость при благоприятных обстоятельствах);
- как набор и степень ограничений, накладываемых на уровень знаний и навыков, которые может приобрести компьютерная система в процессе своего совершенствования (в том числе, и руками ее разработчиков).

Для того, чтобы компьютерная система могла обладать высокой скоростью повышения уровня приобретаемых ею знаний и навыков (т.е. высоким уровнем обучаемости), она должна быть гибкой (легко наращиваемой, легко модифицируемой). Гибкость системы обеспечивается такой ее архитектурой, компоненты которой могут модифицироваться в достаточной степени независимо друг от друга. При этом те факторы, которые обеспечивают связь (взаимодействие) компонентов и требуют четкого согласования для обеспечения целостности системы, четко регламентируются.

Уровень приобретенных системой знаний и навыков может достигнуть способности решать интеллектуальные задачи. Такие системы обычно называют интеллектуальными.

Таким образом, следует отличать

- Общий объем и тематику приобретенных системой знаний и навыков.
- Способность системы быстро приобретать (усваивать, схватывать) новые знания и навыки.

Уровень интеллекта системы определяется скоростью ее обучения (обучаемостью), а качество (уровень) приобретенных знаний и навыков определяется уровнем сложности задач, которые способна решать система. К задачам высшего уровня сложности относятся интеллектуальные задачи.

Можно говорить о классификации компьютерных систем, управляемых знаниями, по следующим признакам:

- уровень интеллектуальности средств обработки знаний;
- уровень развития рецепторно-эффекторных средств взаимодействия с внешней средой.

По первому признаку классификации можно выделить:

- компьютерные системы, управляемые знаниями, не имеющие интеллектуальных решателей задач;
- интеллектуальные системы, управляемые знаниями.

По второму признаку классификации выделим:

- компьютерные системы, управляемые знаниями, не имеющие развитых рецепторно-эффекторных средств взаимодействия с внешней средой;
- робототехнические компьютерные системы, управляемые знаниями.

Особо подчеркнем то, что массовый перевод современных компьютерных систем на уровень систем, управляемых знаниями, необходимо начинать с систем, управляемых знаниями, не имеющих интеллектуальных решателей задач и развитых рецепторно-эффекторных средств. Система, управляемая знаниями, не обязана на каждом этапе своей эволюции уметь решать интеллектуальные задачи, но обязана быть гибкой (легко модифицируемой, легко совершенствуемой) и обязана при необходимости легко приобрести умение решать интеллектуальные задачи.

## **2. Принципы внутреннего представления знаний в системах, управляемых знаниями**

В разработке компьютерных систем, управляемых знаниями, ключевую роль играет способ внутреннего представления знаний. Качество представления знаний компьютерной системы определяется тем, насколько приблизилось это представление к смысловому (семантическому) представлению. Наш подход к формализации смысла представляемых знаний основан на следующих положениях:

- Все известные языки выполняют две функции – коммуникативную (как средство обмена сообщениями между субъектами) и когнитивную (как средство представления информационной модели описываемого Мира).
- Язык внутреннего представления знаний в памяти компьютерной системы не обязан выполнять коммуникативную функцию. От языка внутреннего представления знаний требуется только то, чтобы он обеспечил хранение знаний в удобном для их обработки виде. Удобство обработки хранимых в памяти знаний определяется:
  - (1) простотой процедур информационного поиска фрагментов хранимой базы знаний, удовлетворяющих заданным требованиям;
  - (2) простотой процедур интеграции новых знаний, добавляемых в базу знаний;
  - (3) простотой реализации процедур логического вывода.

Таким образом, все, что обеспечивает только коммуникативные функции языка, из языка внутреннего представления знаний может быть исключено. Язык внутреннего представления знаний в памяти компьютерной системы, основанный на формализации смысла этих знаний, должен выполнять только когнитивную функцию – быть средством представления внутренней информационной модели некоторого описываемого Мира (в том числе и внешней среды соответствующей компьютерной системы).

- Знаки, входящие в состав внутреннего представления знаний, не должны иметь внутренней структуры, т.е. не должны быть представлены в виде некоторого имени соответствующей (обозначаемой) сущности. Смысл каждого знака определяется исключительно его связями с другими знаками, входящими в состав внутреннего представления базы знаний компьютерной системы. В отличие от этого для семантического анализа и понимания сообщений (внешних текстов) нужна структурированность и легкая распознаваемость знаков. По аналогичности структур, изображающих знаки (например, строк символов), определяется синтаксическая эквивалентность знаков, хотя в неформальных языках она не всегда совпадает с их семантической эквивалентностью (т.е. с их синонимией).
- В рамках внутреннего смыслового представления базы знаний компьютерной системы исключается синонимия (дублирование) знаков. Внутренние знаки, обозначающие одну и ту же сущность, должны быть склеены. Как следствие этого в рамках каждой базы знаний исключается семантическая эквивалентность (дублирование) входящих в ее состав фрагментов, т.е. фрагментов, несущих одну и ту же информацию. При этом сохраняется возможность существования логически эквивалентных фрагментов баз знаний, когда один фрагмент является логическим следствием второго и наоборот.
- Тексты языка внутреннего смыслового представления знаний должны быть нелинейными в отличие от привычных текстов, т.к. нелинейные тексты способны более адекватно отразить описываемый ими Мир, который по своей природе нелинеен. Каждая описываемая сущность может быть связана неограниченным числом связей с другими сущностями. Более того, для любой группы сущностей всегда существует связывающая их связь – все в Мире взаимосвязано. Вопрос в том, какие связи целесообразно, а какие нецелесообразно явно представлять в базе знаний. Линейность

привычных текстов – это результат проецирования нелинейного Мира на линейное (одномерное) пространство, что требует дополнительных специальных языковых средств, обеспечивающих не само описание этого Мира, а его проецирование на линейное пространство.

- В рамках языка внутреннего смыслового представления знаний вводятся мощные и простые средства перехода от информации к метainформации (в частности, от слабо структурированных данных к связанным данным). Для этого тексты, входящие в состав базы знаний, также рассматриваются как описываемые сущности, для обозначения которых вводятся соответствующие знаки, каждый из которых трактуется как знак множества всех знаков, входящих в состав обозначаемого текста, включая и входящие в него знаки всевозможных связей.
  - Язык внутреннего смыслового представления знаний должен быть универсальным, т.е. должен обеспечить представление всевозможного вида знаний:
    - спецификаций самых различных сущностей;
    - документаций различных технических систем (в т.ч. систем, построенных по технологии OSTIS);
    - различных предметных областей (как статических, так и динамических);
    - различного вида онтологий предметных областей;
    - текстов высказываний;
    - текстов доказательства теорем;
    - формулировок задач;
    - формулировок классов задач;
    - текстов решений конкретных задач;
    - способов решения различных классов задач;
    - описаний историй эволюции различных систем (в т.ч. систем, разрабатываемых по технологии OSTIS);
    - описаний проектов, направленных на создание и совершенствование различных технических систем (в т.ч. систем, разрабатываемых по технологии OSTIS).
- Использование универсального языка внутреннего представления знаний с возможностью его неограниченного расширения, если возникает необходимость для представления новых видов знаний, создает условия для неограниченного расширения областей применения компьютерных систем, построенных на основе такого внутреннего языка представления знаний.
- В рамках языка внутреннего смыслового представления знаний имена, термины, обозначения, используемые при передаче и

приеме внешних сообщений, также являются самостоятельными описываемыми сущностями, имеющими свои внутренние знаки, которые соответствующими отношениями связываются с внутренними знаками тех сущностей, которые именуются этими внешними обозначениями. Все внешние языки являются для базы знаний частью описываемого ею внешнего Мира.

- Атомарными фрагментами внутреннего смыслового представления базы знаний являются только знаки. При этом каждый внутренний знак сам может быть описываемой сущностью (имеется в виду сам знак, а не сущность, обозначаемая этим знаком). Кроме того, каждая связь между описываемыми сущностями сама также является описываемой сущностью, которая во внутреннем представлении имеет свой внутренний знак и которая трактуется как множество, элементами которого являются знаки сущностей, связываемых описываемой связью. Таким образом, из внутреннего представления знаний исключается все, что имеет отношение не к представлению смысла, а к используемой форме представления. Так, например, во внутреннем представлении знаний нет не только букв, слов, словосочетаний, но и разделителей, ограничителей, предлогов, союзов, местоимений, склонений, спряжений и т.п.
- Язык внутреннего смыслового представления знаний должен быть не только удобным для обработки знаний в компьютерной системе, но должен быть понятным и "прозрачным" как для разработчика компьютерной системы, так и для ее конечного пользователя. Для этого, наряду с разработкой языка внутреннего смыслового представления знаний, должны быть разработаны близкие ему внешние языки, в основе которых должны лежать простые и быстро усваиваемые процедуры трансляции текстов с внутреннего во внешние формы их представления.
- Типология знаков, входящих в состав внутреннего смыслового представления знаний, полностью определяется типологией сущностей, обозначаемых этими знаками. При этом особо выделяется базовая типология описываемых сущностей, которая задает синтаксическую типологию (алфавит) внутренних знаков.

Способ внутреннего представления знаний, используемый в технологии OSTIS, назван SC-кодом (Semantic Code). Знаки, входящие в состав текстов SC-кода, называются sc-элементами. Каждый sc-элемент можно считать инвариантом всего многообразия форм представления (во всевозможных языках и знаковых конструкциях) той сущности, которая

обозначается этим sc-элементом. Таким инвариантом является только то, что указанный sc-элемент обозначает соответствующую ему сущность. Поэтому sc-элемент не имеет формы. В этом смысле он абстрагируется от формы своего представления в рамках той или иной знаковой конструкции.

Каждому sc-элементу взаимно однозначно соответствует некоторый класс синонимичных знаков – множество всевозможных вхождений знаков обозначаемой сущности в самые различные внешние знаковые конструкции (тексты).

Текст SC-кода (sc-текст) с синтаксической точки зрения – это абстрактная знаковая конструкция, представляющая собой множество sc-элементов, связанных между собой отношениями инцидентности. С семантической точки зрения sc-текст – это абстрактная знаковая конструкция, являющаяся инвариантом всего многообразия форм представления соответствующей информации, т.е. инвариантом всего соответствующего класса семантически эквивалентных знаковых конструкций. Следовательно, каждому sc-тексту взаимно однозначно соответствует некоторый класс всевозможных семантически эквивалентных внешних текстов, представляющих заданную информацию в самых различных языках.

Рассмотрим типологию *sc-элементов* как типологию сущностей, обозначаемых *sc-элементами*.

Подчеркнем, что универсальный характер *SC-кода* означает то, что с помощью *sc-элемента* можно обозначить абсолютно любую сущность – как реальную, так и абстрактную.

Приведем описание основных классов *sc-элементов* средствами *SC-кода*, а точнее, средствами *SCg-кода* и *SCn-кода*, которые являются языками эквивалентными и близкими *SC-коду*.

#### **Алфавит SC-кода**

= Семейство классов синтаксически выделенных *sc-элементов*

= Семейство классов *sc-элементов*, элементы каждого из которых имеют одинаковые синтаксические метки

= Семейство классов эквивалентности для неявно заданного отношения эквивалентности "быть *sc-элементами*, имеющими одинаковую синтаксическую метку"

На рис. 1 представлен sc-текст, описывающий связь между *SC-кодом* и *Алфавитом SC-кода* с помощью отношения *алфавит\** и отношения *множество атомарных фрагментов текстов\**.



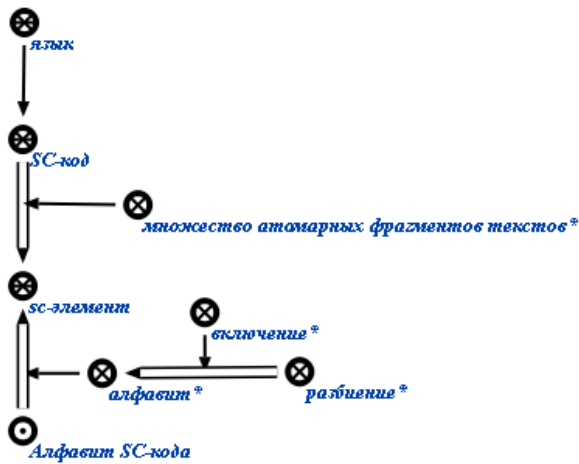


Рисунок 1 – Связь между SC-кодом и Алфавитом SC-кода

Поясним некоторые используемые в рис. 1 понятия.

**язык**

= множество знаковых конструкций, удовлетворяющих общим для этого множества синтаксическим и семантическим правилам

**SC-код**

= Множество текстов SC-кода, являющихся синтаксически и семантически унифицированными семантическими сетями, имеющих базовую теоретико-множественную интерпретацию

= Множество sc-текстов

= sc-текст

В SC-коде синонимия\* между sc-элементами отсутствует, а, точнее, должна выявляться и устраняться путем склеивания синонимичных sc-элементов, либо путем явного указания факта их синонимии.

**синонимия\***

= быть синонимичными sc-элементами\*

= синонимия sc-элементов\*

= отношение, с помощью которого явно указывается факт синонимии некоторых sc-элементов\*

∈ ключевой sc-элемент\*:

...

∈ sc-примечание

<= трансляция sc-текста\*:

...

С Русский язык

∃ хранимый файл\*:

[Явно указываемая синонимия sc-элементов может использоваться для повышения живучести системы, построенной на основе SC-кода.

Это, своего рода, резервное копирование (дублирование) sc-элементов, которое должно учитываться при информационном поиске.

Заметим также, что синонимичными друг другу могут быть не только константные, но и переменные sc-элементы. ]

Кроме того существует синонимия между синтаксически выделенными классами sc-элементов и соответствующими им семантически выделенными классами sc-элементов. Дело в том, что sc-элементы, принадлежащие разным синтаксически выделенным классам, не могут пересекаться, то есть не могут иметь сразу несколько синтаксических меток. Тогда, как соответствующие им семантически выделенные классы sc-элементов могут не просто пересекаться, а могут входить один в другой.

На рис. 2 представлены элементы Алфавита SC-кода, т.е. синтаксически выделяемые классы sc-элементов и синонимичные им семантически выделяемые классы sc-элементов.

Здесь, например, синонимия синтаксически задаваемого класса sc-узлов и класса всех sc-элементов означает то, что любой sc-элемент может быть представлен в форме sc-узла.

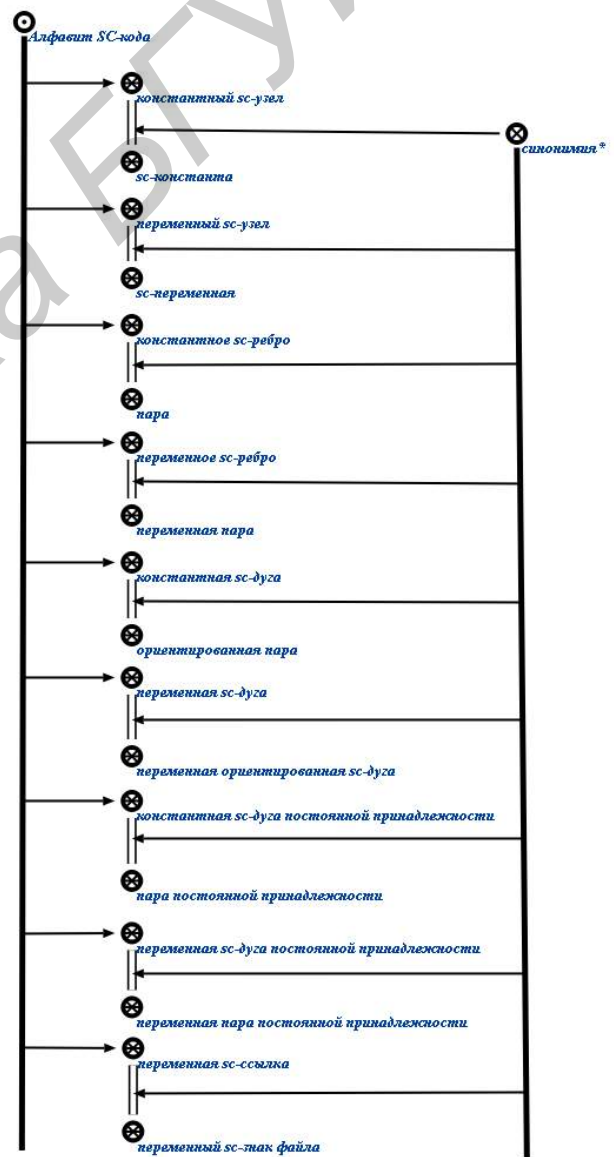


Рисунок 2 – Алфавит SC-кода



### **sc-коннектор**

= синтаксически атомарная связка sc-элементов

<= разбиение\*:

- {
- sc-ребро  
<= разбиение\*:
  - {
  - константное sc-ребро
  - переменное sc-ребро
  - }
- sc-дуга  
<= разбиение\*:
  - {
  - константная sc-дуга общего вида
  - переменная sc-дуга общего вида
  - константная sc-дуга постоянной принадлежности
  - переменная sc-дуга постоянной принадлежности
  - }
- }

### **sc-элемент**

= sc-знак

= знак сущности

= знак, входящий в текст SC-кода

= сущность

= sc-знак сущности

= знак, входящий в состав текста SC-кода

= Множество всевозможных атомарных фрагментов текста

<= разбиение\*:

- {
- sc-константа
- sc-переменная
- }

<= разбиение\*:

- {
- материальная или квазиматериальная сущность
- абстрактная сущность
- }

### **sc-константа**

= константный sc-элемент

<= разбиение\*:

- {
- терминальная sc-константа
- множество
- }

<= разбиение\*:

- {
- вымышленная сущность  
= придуманная сущность
- невымышленная сущность  
= действительно существующая сущность
- }

<= разбиение\*:

- {
- постоянная сущность  
= постоянно существующая сущность  
= стационарная сущность
- }

- временная сущность  
= временно существующая сущность  
= нестационарная сущность

### **терминальная sc-константа**

= терминальная сущность

= первичная sc-константа

= первичная сущность

= sc-константа, не являющаяся знаком множества

= терминальная константная сущность

<= разбиение\*:

- {
- материальная или квазиматериальная сущность
- sc-знак файла
- абстрактная терминальная сущность  
= абстрактная константная (конкретная, фиксированная) сущность, не являющаяся множеством
  - ⊃ действительное число
  - ⊃ геометрическая точка
  - ⊃ точка маршрута в графовой структуре
- }

### **материальная или квазиматериальная сущность**

<= разбиение\*:

- {
- материальная сущность
- квазиматериальная сущность  
= вымышленная сущность, подобная материальной  
= псевдоматериальная сущность  
= sc-знак квазиматериальной сущности  
= придуманная сущность, подобная материальной
- }

### **материальная сущность**

= естественная сущность

= объект исследования естественной науки

= физическая сущность

= материальный объект

= реальная сущность

= знак материальной сущности

= sc-знак материальной сущности

= sc-элемент, обозначающий конкретную материальную сущность

### **sc-знак файла**

= константный sc-знак файла

= sc-константа, обозначающая файл, хранимый либо во внешней файловой памяти sc-системы, либо в другом месте

= sc-знак конкретного (фиксированного) файла

= конкретный файл

= файл

= электронный образ внешней информационной конструкции

= внешняя информационная конструкция

⊃ sc-идентификатор

= *sc*-знак файла, в котором хранится в определенном формате электронный образ одного из экземпляров класса синтаксически эквивалентных конструкций, все или многие из которых, входя во внешние тексты, обозначают ту же сущность, что и соответствующий им *sc*-элемент

= *sc*-знак внешнего идентификатора некоторого *sc*-элемента

= внешний идентификатор *sc*-элемента

= *sc*-знак *sc*-идентификатора

#### **множество**

= *sc*-множество

= множество *sc*-элементов

= *sc*-знак множества *sc*-элементов

= множество знаков описываемых сущностей

= текст SC-кода

= *sc*-текст

= SC-код

<= разбиение\*:

- {
- *связка*
- *класс*
- *структура*
- }

∈ *ключевой sc-элемент*':

...

∈ *sc-примечание*

<= *трансляция sc-текста*\*:

...

⊂ *Русский язык*

∃ *хранимый файл*':

[Существенно подчеркнуть то, что любой текст SC-кода представляет собой множество *sc*-элементов, связанных между собой отношениями инцидентности *sc*-коннекторов с их компонентами. Таким образом, понятия текста SC-кода и множества *sc*-элементов являются синонимами.]

∈ *ключевой sc-элемент*':

...

∈ *sc-примечание*

<= *трансляция sc-текста*\*:

...

⊂ *Русский язык*

∃ *хранимый файл*':

[Заметим, что все семантические виды множеств (и связки, и классы, и структуры) могут быть компонентами связок любых теоретико-множественных отношений (включение\*, объединение\*, разбиение\*, пересечение\* и т.д.).]

#### **связка**

= *связь между сущностями*

= *sc-связка*

= *связка sc-элементов*

= *sc-знак связи*

<= *разбиение*\*:

- {
- *небинарная связка*

- *пара*

}

<= *разбиение*\*:

{

- *постоянная связка*  
= *стационарная связка*
- *временная связка*  
= *нестационарная связка*

}

#### **небинарная связка**

= *небинарная sc-связка*

= *небинарная связка sc-элементов*

= *sc-связка, имеющая мощность либо равную единице, либо большую, чем два*

⊃ *синглетон*

= *унарная связка*

= *одномощное множество sc-элементов*

⊃ *тройка*

= *тернарная связка*

#### **пара**

= *бинарная связка*

= *пара sc-элементов*

= *sc-пара*

= *константная пара sc-элементов*

= *двухмощное множество*

<= *разбиение*\*:

{

- *неориентированная пара*
- *ориентированная пара*

}

#### **ориентированная пара**

= *ориентированная бинарная связка*

= *ориентированная пара sc-элементов*

= *константная ориентированная пара sc-элементов*

⊃ *пара постоянной принадлежности*

= *пара стационарной принадлежности*

= *sc-пара стационарной принадлежности*

= *Отношение постоянной принадлежности*

= *Отношение стационарной принадлежности*

⊃ *пара временной принадлежности*

= *пара нестационарной принадлежности*

= *Отношение временной принадлежности*

⊃ *пара постоянной непринадлежности*

⊃ *пара временной непринадлежности*

⊃ *пара постоянной нечеткой принадлежности*

⊃ *пара временной нечеткой принадлежности*

⊃ *пара принадлежности или непринадлежности*

<= *разбиение*\*:

{

- *пара постоянной принадлежности или непринадлежности*
- *пара временной принадлежности или непринадлежности*

}

<= *разбиение*\*:

{

- *пара принадлежности*
- *пара непринадлежности*

}

- пара нечеткой принадлежности

}

### **класс**

- = sc-класс
- = класс sc-элементов
- = sc-понятие
- = sc-концепт
- = множество всех тех и только тех sc-элементов, которые обозначают сущности, эквивалентные в заданном смысле, т.е. удовлетворяющие соответствующим требованиям
- ⊃ класс терминальных сущностей
- ⊃ класс материальных или квазиматериальных сущностей
- ⊃ класс абстрактных терминальных сущностей
- ⊃ отношение
- ⊃ класс структур
- ⊃ класс классов
- ⊃ параметр
- ⊃ величина

### **отношение**

- = sc-отношение
- = класс связей
- = класс связей, эквивалентных в заданном смысле
- = отношение, заданное на множестве sc-элементов
- = класс связей, связывающих sc-элементы
- <= разбиение\*:

{

- бинарное отношение
  - = класс пар sc-элементов
  - ⊃ квазибинарное отношение
    - ⊃ сложение\*
    - ⊃ умножение\*
    - ⊃ объединение\*
    - ⊃ пересечение\*
    - ⊃ соответствие\*
      - ⊃ изоморфизм\*
      - ⊃ гомоморфизм\*
      - ⊃ биекция\*
  - ⊃ ролевое отношение
- класс небинарных связей одинаковой мощности
- класс связей разной мощности

}

### **ролевое отношение**

- = отношение, являющееся подмножеством множества пар постоянной и-или временной принадлежности
- ⊃ пара постоянной принадлежности
- ⊃ ключевой sc-элемент'
  - = ключевой элемент sc-структуры'
- ⊃ образец'
  - = контрольный экземпляр'
- ⊃ пример'
- ⊃ хранимый файл'
  - = экземпляр класса синтаксически эквивалентных информационных конструкций, хранимый в виде доступного файла'

### **параметр**

- = характеристика
- = свойство
- = семейство классов сущностей, имеющих одинаковое или почти одинаковое значение заданного параметра
- = признак классификации сущности некоторого класса
- <= разбиение\*:

{

- измеряемый параметр
  - = количественный параметр
- неизменяемый параметр
  - = качественный параметр

}

### **величина**

- = значение какого-либо параметра
- = класс сущностей, имеющих одинаковое или почти одинаковое значение какого-либо параметра
- ∈ ключевой sc-элемент':
  - ...
  - ∈ sc-примечание
  - <= трансляция sc-текста\*:
  - ...
  - ⊃ Русский язык
  - ⊃ хранимый файл':
    - [Каждая величина связывается с соответствующим ей параметром с помощью следующих ролевых отношений:
      - точная величина'
      - неточная величина'
      - величина в фиксированном интервале']

### **структура**

- = sc-структура
- = структура sc-элементов
- <= разбиение\*:

{

- тривиальная структура
  - = дискретная структура
  - = структура, не содержащая связей, связывающих элементы этой структуры
  - ⊃ геометрическая фигура
- нетривиальная структура
  - = структура, содержащая связи, связывающие элементы этой структуры

}

<= разбиение\*:

{

- связная структура
- несвязная структура
  - ⊃ тривиальная структура

}

### **sc-переменная**

- = sc-знак переменной сущности
- = переменная сущность
- = обозначение произвольной сущности, которое пробегает некоторое дополнительно уточняемое множество знаков
- = sc-обозначение произвольной сущности,

которому соответствует некоторое дополнительно уточняемое множество sc-элементов, являющееся областью возможных значений этого sc-обозначения  
∈ ключевой sc-элемент':

...

∈ sc-сравнение

<= трансляция sc-текста\*:

...

⊂ Русский язык

∃ хранимый файл':

[В отличие от переменной, область возможных значений каждой константы состоит только из самой этой константы]

⊃ sc-переменная первого порядка

= sc-переменная, значениями которой являются только sc-константы

⊃ sc-переменная второго порядка

= sc-метапеременная

= sc-переменная, значениями которой являются только sc-переменные первого порядка

⊃ sc-переменная смешанного порядка

= sc-переменная, значениями которой являются как sc-константы, так и sc-переменные первого порядка

### **абстрактная сущность**

= sc-знак абстрактной сущности

<= разбиение\*:

{

• sc-переменная

= sc-знак произвольной сущности из некоторого дополнительно уточняемого множества сущностей

• множество

= sc-множество

= множество sc-элементов

= sc-знак множества sc-знаков

• абстрактная терминальная сущность

= первичный элемент абстрактной структуры

}

### **временная сущность**

= временно существующая сущность

= нестационарная сущность

= процесс

<= разбиение\*:

{

• материальная сущность

• квазиматериальная сущность

• временная связка

= временно существующая связь

= нестационарная связка

• ситуация

= временно существующая структура

= нестационарная структура

= состояние динамической системы

• событие

= переходный процесс

= процесс перехода от одного состояния к другому

⊃ действие

= воздействие

= событие, осуществляемое неким целенаправленным активным субъектом

}

<= разбиение\*:

{

• сущность прошлого времени

= прошлая сущность

= сущность, прекратившая свое существование

= сущность, существовавшая в прошлом времени

= сущность "жившая" в прошлом времени

• сущность настоящего времени

= сущность, существующая в текущий момент времени

= Множество всех сущностей, существующих в текущий момент

= Текущее состояние окружающего Мира

• сущность будущего времени

= будущая сущность

}

При классификации знаков по признаку постоянства обозначаемых ими сущностей следует отличать постоянство (вечность и временность) обозначаемых сущностей и постоянство самих знаков, хранимых в памяти компьютерной системы.

В общем случае можно утверждать, что ничего вечного в базе знаний нет. Просто одни знаки "живут" (присутствуют, хранятся) в базе знаний дольше, другие имеют короткую "жизнь" (входя, например, в состав временной, вспомогательной информации, используемой для решения конкретной задачи и теряющей ценность после решения этой задачи).

Таким образом, не следует путать временный характер существования некоторой описываемой сущности и временный характер присутствия знака этой сущности в рамках текущего состояния базы знаний. Из состава текущего состояния базы знаний может быть удален любой знак, обозначающий как временно существующую, так и постоянно существующую сущность.

Для описания бывшей, настоящей и будущей принадлежности того или иного sc-элемента текущему состоянию базы знаний той или иной компьютерной системы вводятся следующие понятия.

### **текущее состояние базы знаний**

= множество sc-элементов, входящих в состав текущего состояния базы знаний

= Семейство текущих состояний баз знаний различных компьютерных систем

= множество настоящих знаков базы знаний соответствующей системы

∋ Текущее состояние базы знаний IMS

∋ Текущее состояние базы знаний ИСС по геометрии

**множество sc-элементов, удаленных из текущего состояния базы знаний**

- = множество знаков, логически удаленных из базы знаний соответствующей системы
- = множество бывших знаков базы знаний
- = множество прошлых знаков базы знаний
- ⊃ Множество sc-элементов, удаленных из текущего состояния базы знаний IMS
- ⊃ Множество sc-элементов, удаленных из текущего состояния базы знаний ИСС по геометрии

**множество sc-элементов, которые планируется ввести в текущее состояние базы знаний**

- = множество будущих знаков базы знаний
- ⊃ Множество sc-элементов, которые планируется ввести в текущее состояние базы знаний IMS
- ⊃ Множество sc-элементов, которые планируется ввести в текущее состояние базы знаний ИСС по геометрии

На рис. 3 в качестве примера приведен фрагмент спецификации Метасистемы IMS, в котором используются введенные выше понятия.

В этом рисунке особое место занимает sc-узел с идентификатором "Я". Этот узел является sc-знаком синглтона, единственным элементом которого является sc-знак той компьютерной системы, в базу знаний которой указанный sc-узел входит. Таким образом, рассматриваемый синглетон есть не что иное, как указатель на самого себя, на собственное Я.

Рассмотрим кратко типологию знаний, входящих в состав внутреннего представления базы знаний, которая полностью определяется (дается) типологией знаков, обозначающих эти знания. К числу основных видов знаний, представленных в SC-коде, относятся предметные области, онтологии, семантические окрестности, sc-тексты высказываний, sc-тексты доказательств, задачи, планы решения конкретных задач, sc-тексты обоснования решений задач, обобщенные формулировки задач соответствующих классов, способы решения задач соответствующих классов, разделы баз знаний.

**знание**

- = семантически осмысленная и целостная sc-структура
- = sc-знание
- = знание, представленное в SC-коде
- ⊂ структура
- ⊃ предметная область
- ⊃ онтология
- ⊃ логическая система понятий и их определений
- ⊃ логическая система утверждений и их доказательств
- ⊃ семантическая окрестность
- ⊃ sc-сравнение
- ⊃ sc-текст высказывания
- ⊃ sc-текст доказательства

⊃ задача

- = sc-текст формулировки задачи
- = sc-формулировка задачи
- ⊃ план решения задачи
- = sc-текст плана решения задачи
- ⊃ sc-текст обоснования решения задачи
- ⊃ обобщенная формулировка задач соответствующего класса
- = sc-текст обобщенной формулировки задач некоторого класса
- ⊃ способ решения задач соответствующего класса
- = sc-текст способа решения задач
- = sc-текст обобщенного плана решения задач некоторого класса
- ⊃ раздел базы знаний

Важнейшим отношением, заданным на множестве знаний является отношение *быть метазнанием\**, описывающее переход от знаний к описывающим их метазнаниям (в частности, от данных – к связанным данным, от неструктурированных данных – к структурированным).

**метазнание\***

- = *быть метазнанием\**
- = Бинарное ориентированное отношение, связки которого связывают некоторое исходное знание со знанием, которое является метаотношением исходного знания, его спецификацией, описанием его структуры\*
- ⊃ метазнание, основанное на выделении тривиальных структур из терминальных элементов исходного знания\*
- ⊃ метазнание, основанное на выделении нетривиальных структур, являющихся фрагментами исходного знания\*

Примером связи между знанием и соответствующим ему метазнанием является переход от некоторого исходного знания к описанию его декомпозиции (сегментации) на некоторые части с указанием связей между этими частями.

На рис. 4 приведены примеры использования отношения *метазнание\** (*быть метазнанием\**), а также частных видов этого отношения.

Важнейшим ролевым отношением, связывающим sc-знаки знаний с компонентами этих знаний, является отношение *ключевой sc-элемент'* (*быть ключевым sc-элементом'*).

**предметная область**

- = sc-модель предметной области
- ⊂ знание
- <= разбиение\*:

- {
  - статическая предметная область
  - = предметная область, в состав которой входят постоянно существующие объекты исследования, постоянно существующие связи и структуры
  - = стационарная предметная область

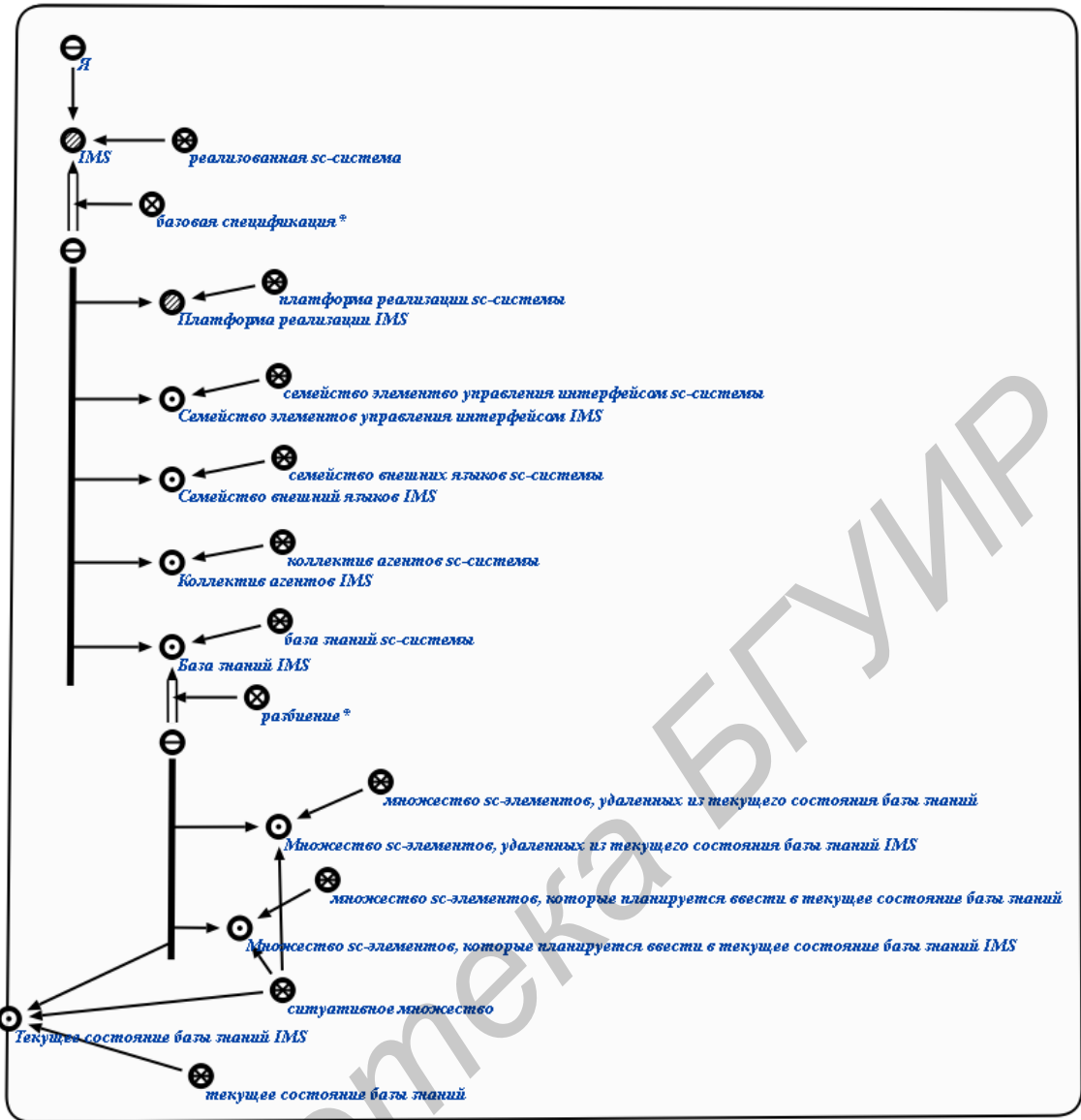


Рисунок 3 – Текст SC-кода, описывающего базовую спецификацию IMS

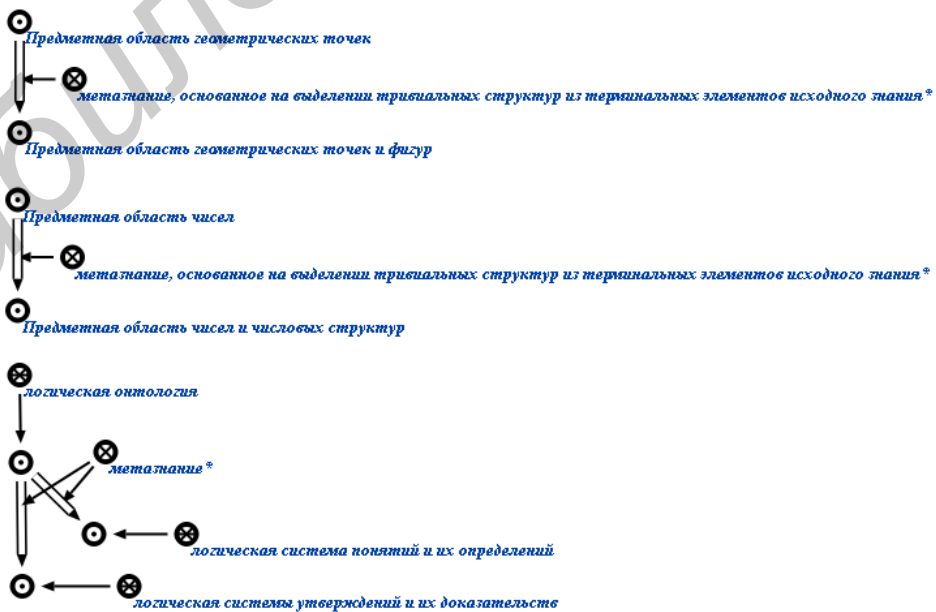


Рисунок 4 – Примеры связей, связывающих знания и соответствующие им метазнания

- динамическая предметная область  
= нестационарная предметная область
- }  
 ∈ ключевой sc-элемент':  
 ...  
 ∈ sc-примечание  
 <= трансляция sc-текста\*:  
 ...  
 C Русский язык  
 ∃ хранимый файл':  
 [Каждая предметная область задает соответствующий специализированный язык (sc-язык), представляющий собой множество всевозможных фрагментов (подструктур), включаемых в состав этой предметной области. Так, например, каждая теоретико-множественная онтология является текстом языка, задаваемого Предметной областью множеств.]
- ∃ Предметная область предметных областей  
= Предметная область, задающая sc-язык структурных спецификаций предметных областей
  - ∃ Предметная область множеств  
= Предметная область, которая задает теоретико-множественный sc-язык, используемый, в частности, для записи теоретико-множественных онтологий
  - ∃ Предметная область sc-связок и отношений
  - ∃ Предметная область логических формул
  - ∃ Предметная область sc-идентификаторов
  - ∃ Предметная область параметров, величин и шкал
  - ∃ Предметная область scp-программ

**Предметная область логических формул**  
 = Предметная область, определяющая специализированный sc-язык логических формул, высказываний и формальных теорий (логических онтологий)  
 = Предметная область, являющаяся объединением всевозможных текстов логического sc-языка (Языка SCL)

**Предметная область scp-программ**  
 = Предметная область, определяющая специализированный sc-язык SCP

**SCP**  
 = Semantic Code Programming  
 = Специализированный sc-язык, являющийся базовым языком программирования, ориентированным на обработку sc-текстов

**Предметная область параметров, величин и шкал**  
 = Предметная область, задающая sc-язык параметрических моделей всевозможных сущностей  
 = Предметная область параметров и их значений  
 = Предметная область, представляющая собой результат объединения всевозможных параметрических моделей  
 = Предметная область параметрического

- описания всевозможных сущностей  
 ∃ максимальный класс объектов исследования':
- параметр  
= признак классификации  
= характеристика
  - величина  
= значение параметра
  - шкала  
= множество результатов измерения величины
- ∃ исследуемое отношение':
- точность\*  
= Отношение, связывающее каждую неточную величину с той точной величиной, которая задает половину интервала, в рамках которого указанная неточная величина находится
  - измерение\*
  - единица измерения\*
  - шкала измерения\*
- ∃ немаксимальный класс объектов исследования':
- количественный параметр
  - качественный параметр
  - точная величина
  - неточная величина
  - величина в фиксированном интервале
  - дискретная шкала
  - непрерывная шкала

На множестве предметных областей заданы следующие неровные отношения:

- частная предметная область\*  
∈ антитранзитивное отношение
- онтология\*  
= быть онтологией заданной предметной области
- включение\*

На рис. 5 приведен пример использования указанных отношений.

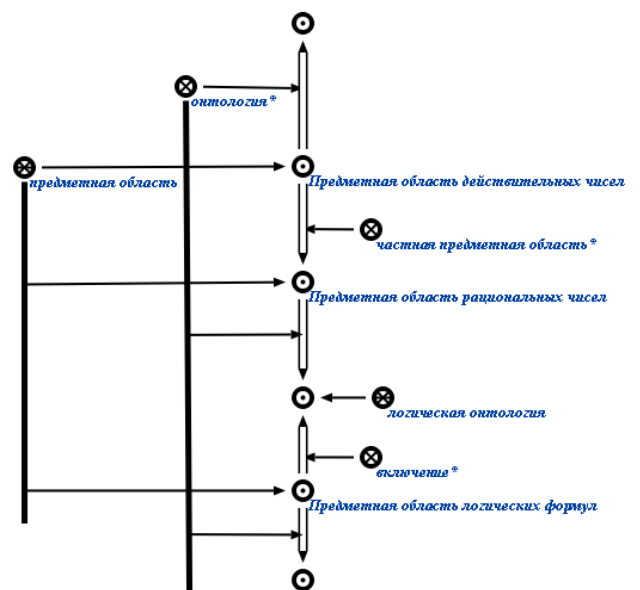


Рисунок 5 – Пример использования отношений, заданных на множестве предметных областей



### **частная предметная область\***

= быть частной предметной областью\*

<= разбиение\*:

- {
- частная предметная область, являющаяся результатом сужения класса исследуемых объектов\*
- частная предметная область, являющаяся результатом сужения набора отношений, заданных на объектах исследования или вторичных им объектах\*
- }

К числу ролевых отношений, указывающих роли компонентов предметных областей, относятся:

- *максимальный класс объектов исследования'*
- *немаксимальный класс объектов исследования'*
- *исследуемое понятие'*  
= понятие, исследуемое в данной предметной области'
- *понятие, исследуемое в другой предметной области'*  
▷ *понятие, исследуемое в частной предметной области'*  
= *понятие, определяемое и исследуемое в частной предметной области'*  
= *понятие, исследуемое в предметной области, которая связана с заданной связкой отношения быть частной предметной областью'*
- *ключевой объект исследования'*  
= *объект исследования, обладающий особыми свойствами'*

К числу ключевых объектов исследования', например, в рамках Предметной области комплексных чисел относятся числа ноль, единица, мнимая единица.

### **максимальный класс объектов исследования'**

⊂ *ключевой sc-элемент предметной области'*

∈ *ключевой sc-элемент'*:

...

∈ *sc-примечание*

<= *трансляция sc-текста\**:

...

⊂ *Русский язык*

∃ *хранимый файл'*:

[Предметная область может иметь несколько максимальных классов объектов исследования, если нет смысла объединять эти классы в одно общее понятие.]

### **онтология**

= *sc-онтология*

= *sc-модель онтологии предметной области*

= *онтология предметной области*

= *онтология предметной области, представленная в SC-коде*

= *sc-текст онтологии*

= *спецификация предметной области*

⊂ *знание*

<= *разбиение\**:

{

- *интегрированная онтология*  
= *объединение онтологий различного вида, специфицирующих одну и ту же предметную область*
- *структурная спецификация предметной области*
- *теоретико-множественная онтология*
- *терминологическая онтология*
- *логическая онтология*
- *онтология задач и решений задач*
- *онтология классов задач и способов решения задач*

}

Связь между предметной областью и ее онтологией задается отношением *онтология\** (*быть онтологией\**), которое является частным видом отношения *метазнание\** (*быть метазнанием\**).

### **семантическая окрестность**

= *sc-текст, посвященный описанию некоторой сущности, знак которой является ключевым sc-элементом указанного sc-текста*

= *sc-окрестность*

= *описание (спецификация) некоторой сущности, выполненное с той или иной степенью детализации и в том или ином ракурсе*

▷ *базовая семантическая окрестность*

= *базовая семантическая окрестность в рамках текущего состояния базы знаний*

▷ *общая семантическая окрестность*

= *общая семантическая окрестность в рамках текущего состояния базы знаний*

▷ *sc-текст определения*

▷ *sc-пояснение*

▷ *sc-примечание*

▷ *параметрическая модель*

▷ *документация*

▷ *документация технической системы*

▷ *документация проекта*

▷ *документация технологии*

▷ *биография*

▷ *медицинская карта пациента*

Каждая базовая семантическая окрестность включает в себя:

- все классы, которым принадлежит знак описываемой сущности;
- все связки, компонентами которых является знак описываемой сущности с указанием (1) ролей этого знака, (2) классов, которым принадлежат указанные связки и (3) остальных компонентов указанных связок;
- все структуры, элементами которых является знак описываемой сущности с указанием (1) ролей этого знака в рамках указанных структур, (2) классам, которым принадлежат эти структуры.

Кроме того:

- если описываемая сущность является связкой (связкой), то в ее базовую семантическую

окрестность включаются все компоненты этой связи с указанием их ролей;

- если описываемая сущность является конечной структурой, то в ее базовую семантическую окрестность включаются все компоненты этой структуры с указанием их ролей, если таковые имеются;
- если описываемая сущность является бесконечной структурой (например, бесконечной предметной областью), то в ее базовую семантическую окрестность включаются все ключевые sc-элементы этой структуры с уточнением их ролей в рамках этой структуры;
- если описываемая сущность является классом, то в ее базовую семантическую окрестность включаются все ключевые элементы этого класса, отмеченные такими ролевыми отношениями, как *пример*, *образец*, *контрольный экземпляр*.

Каждая *общая семантическая окрестность* заданной описываемой сущности полностью включает в себя *базовую семантическую окрестность* этой сущности, но, кроме этого, в *общую семантическую окрестность* включаются:

- *параметрическая модель* описываемой сущности;
- все *sc-тексты высказываний*, описывающих свойства данной сущности;
- идентификация описываемой сущности в различных внешних информационных конструкциях и, если описываемая сущность является классом, то и правила построения таких идентификаторов для элементов указанного класса;
- история исследования описываемой сущности или история ее создания, если она является артефактом, с указанием конкретных персон, принимавших участие в этих процессах, и, в первую очередь, авторов соответствующих исследований или разработок.

#### **параметрическая модель**

= *параметрическая модель некоторой сущности*

= *спецификация заданной сущности через указание ее "координат" в абстрактном параметрическом пространстве (в пространстве признаков)*

= *параметрическая спецификация*

⊆ *семантическая окрестность*

Множество всевозможных параметрических моделей можно считать специализированным языком параметрических моделей. Объединением всех sc-текстов этого языка является *Предметная область параметров, величин и шкал*.

#### **sc-сравнение**

= *sc-текст, посвященный сравнительному анализу конечного числа (чаще всего, двух) сущностей, знаки которых являются ключевыми элементами указанного sc-текста*

#### **раздел базы знаний**

= *sc-текст раздела базы знаний*

= *фрагмент базы знаний, совершенствование которого может осуществляться в достаточной степени независимо от других специально выделенных фрагментов базы знаний*

= *специально выделенный фрагмент базы знаний, которому однозначно ставится в соответствие персона, администрирующая этот фрагмент и несущая полную персональную ответственность за его актуализацию, обновление и совершенствование*

⊃ *база знаний*

= *целостная база знаний компьютерной системы, управляемой знаниями*

= *sc-модель базы знаний*

= *база знаний компьютерной системы, представленная в SC-коде*

∈ *ключевой sc-элемент*:

...

∈ *sc-примечание*

<= *трансляция sc-текста* \*:

...

⊆ *Русский язык*

∋ *хранимый файл*':

[Разделом базы знаний может быть предметная область, онтология, документация, обоснование разработки, документация. Частным видом разделов баз знаний являются сами базы знаний, которые должны удовлетворять целому набору требований, обеспечивающих их целостность.]

### **3. Основные положения предлагаемой технологии, ориентированной на проектирование систем, управляемых знаниями, и принципы организации проекта, направленного на создание и использование этой технологии**

В настоящее время одним из главных направлений развития искусственного интеллекта является создание общей, доступной и эффективной технологии комплексного проектирования систем, управляемых знаниями. К основным принципам, лежащим в основе любой эффективной технологии (не только проектирования компьютерных систем) можно отнести:

- Четкое отделение этапа проектирования от этапа производства и соответствующее этому четкое отделение технологии проектирования от технологии производства
- Унификация результатов проектирования и, соответственно этому, независимость проектирования от производства (для компьютерных систем – платформенная независимость логического проектирования)

- Компонентное (модульное) проектирование, на основе постоянно пополняемых библиотек многократно используемых компонентов
- Постоянное совершенствование технологии как со стороны науки, так и со стороны инженерии (должен накапливаться и систематизироваться проектный опыт), поэтому технология должна быть доступной, открытой и должна быть результатом широкого сотрудничества всех участников.

В соответствии с этим, к основным положениям технологии OSTIS, ориентированной на проектирование систем, управляемых знаниями, можно отнести следующее.

### **3.1. Базовая графодинамическая модель обработки информации**

В качестве формальной основы проектируемых интеллектуальных систем предлагается использовать графодинамические модели обработки информации специального вида, ориентированные на параллельную асинхронную обработку информации.

Графодинамическая модель обработки информации трактует процесс обработки информации как процесс преобразования графовой структуры, в ходе которого меняется не только состояние элементов этой графовой структуры, но и конфигурация этой структуры (появляются или удаляются её вершины, а также связи между ними). Заметим, что для создания графодинамических моделей обработки информации недостаточно тех видов графовых структур, которые в настоящее время исследуются в теории графов [Касьянов, 2003]. Общее определение графовой структуры, на основе которого можно строить практически полезные графодинамические модели обработки информации приведено в работе [Голенков и др, 2001].

Без организации параллельной обработки информации невозможно рассчитывать на необходимую производительность подавляющего числа практически полезных интеллектуальных систем. 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;
- коллектива агентов, работающих над общей для них графодинамической памятью и обменивающихся информацией только 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. Другими словами, процесс обработки информации в графодинамической памяти сводится не только к изменению состояния элементов памяти, но и к изменению конфигурации связей между ними.

Агенты, работающие над 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 интеллектуальной системы, 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.

### **3.2. Унификация семантических моделей обработки знаний**

В основе предлагаемого подхода к созданию технологии проектирования систем, управляемых знаниями, лежит принцип унификации семантических моделей интеграции знаний и семантических моделей интеграции целых систем, управляемых знаниями.

На основе унифицированных семантических сетей необходимо 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.

Унифицированный способ кодирования различных видов знаний необходим для обеспечения совместимости различных видов знаний и различных языков представления знаний, что, в свою очередь необходимо для интеграции знаний [Ивашенко, 2012], а также для интеграции различных языков представления знаний. Без приведения интегрируемых знаний к одинаковой (общей, унифицированной) форме интеграция невозможна.

Унификация семантических моделей обработки знаний предполагает:

- унификацию структуризации баз знаний и типологии знаний;
- унификацию моделей информационного поиска;
- унификацию моделей интеграции знаний;
- унификацию моделей решения задач;
- унификацию моделей пользовательских интерфейсов;
- унификацию визуализации баз знаний.

### **3.3. Компонентное проектирование интеллектуальных систем**

Проектируемые системы, управляемые знаниями, обладают высоким уровнем гибкости, их разработка в предлагаемой технологии осуществляется поэтапно, переходя от одной целостной версии системы к другой. При этом стартовая версия системы может быть ядром соответствующего класса систем, входящим в библиотеку многократно используемых компонентов. Предлагаемая технология концентрирует внимание не на создании стартовых версий разрабатываемых систем (это превращается в достаточно простую их сборку из имеющихся многократно используемых компонентов), а на расширении и совершенствовании текущих версий разрабатываемых систем в ходе их эксплуатации.

Проектирование по предлагаемой технологии имеет компонентный характер (модульный), основанный на библиотеке многократно используемых компонентов. Это многократное использование типовых компонентов уже разработанных компьютерных систем и модульное проектирование, заключающееся в сборке (синтезе) проектируемых систем из крупных ранее разработанных компонентов. Организация такого сборочного модульного проектирования требует создания, постоянного пополнения и сопровождения специальной библиотеки многократно используемых типовых ip-компонентов (компонентов интеллектуальной собственности – Intellectual Property Core).

Библиотека ip-компонентов – это, своего рода, систематизированный каталог комплектации для сборочного проектирования компьютерных систем, формальное описание коллективного опыта всех разработчиков таких систем.

### **3.4. Доступность и открытость технологии**

В предлагаемой технологии осуществляется постоянное накопление и систематизация накапливаемого проектного опыта по разработке различных классов и различных видов компонентов систем, управляемых знаниями. Такое накопление проектного опыта осуществляется самими разработчиками прикладных систем. Конструктивным результатом этого процесса является расширение соответствующих разделов библиотеки многократно используемых компонентов.

Предлагаемая технология носит открытый и доступный характер как для ее пользователей (разработчиков прикладных систем, управляемых знаниями), так и для тех, кто желает участвовать в ее развитии.

Доступность предлагаемой технологии для ее понимания и усвоения обеспечивается тем, что документация по технологии оформляется как раздел базы знаний метасистемы, благодаря чему метасистема может выполнять функцию интеллектуальной справочной системы по предлагаемой технологии.

Реализация предлагаемой технологии осуществляется в виде метасистемы, управляемой знаниями, которая осуществляет поддержку проектирования систем по предлагаемой технологии, которая сама реализуется по этой же технологии и которая, как и другие системы, разрабатываемые по предлагаемой технологии, постоянно расширяется и совершенствуется. Таким образом, от гибкости (модифицируемости) проектируемых систем, управляемых знаниями, мы переходим к гибкости самой технологии их проектирования. Этой является важнейшим достоинством предлагаемой технологии, поскольку для любой технологии важен не столько текущий уровень ее развития, сколько трудоемкость и скорость ее перехода на новый уровень.

### **3.5. Принципы организации проекта OSTIS**

Проект OSTIS, направленный на создание семантической технологии компонентного проектирования компьютерных систем, управляемых знаниями, и на применение этой технологии имеет следующие особенности:

- Предлагаемая технология является универсальной, т.к. любую компьютерную систему можно построить на ее основе, сутью предлагаемой технологии является новая парадигма построения компьютерных систем,

в которой доминирующую роль играют не программы, а обрабатываемая информация, имеющая в памяти системы онтологически унифицированное смысловое представление, основанное на согласованной системе понятий. При этом хранимая в памяти информация включает в себя как непосредственно обрабатываемую часть, так и часть, обеспечивающую управление процессом обработки информации, т.е. управление агентами, работающими над памятью (их инициирование и организацию взаимодействия).

- Развитие и расширение сфер применения предлагаемой технологии проектирования систем, управляемых знаниями, есть не что иное, как переход к компьютерным системам принципиально нового поколения, обладающих существенными преимуществами по сравнению с системами, построенными по традиционным технологиям.
- Развитие и расширение сфер применения предлагаемой технологии проектирования систем, управляемых знаниями, и, в частности, интеллектуальных систем (ИС), требует комплексного подхода и согласованного взаимодействия науки, образования, инженерии и бизнеса, см. рис.6.

Подчеркнем, что технология проектирования технических систем любого класса является ключевым фактором взаимодействия и продуктом сотрудничества соответствующей области науки, образования, инженерии и бизнеса.

Разработка большинства компьютерных систем сталкивается с недостаточностью развития используемых технологий. Поэтому разработка каждой такой системы должна включать в себя совершенствование используемой технологии – это необходимое условие создания конкурентоспособной системы. Из этого следует, что технологии проектирования сложных компьютерных систем должны быть гибкими (легко расширяемыми и модифицируемыми) и должен быть организован открытый массовый процесс развития каждой такой технологии, охватывающий всех ее пользователей. Если разработчики прикладных компьютерных систем сами не участвуют в развитии соответствующей технологии, то их квалификация и, следовательно, конкурентоспособность разрабатываемых ими систем будет падать.

Если обучение студентов и магистрантов разработке прикладных компьютерных систем на основе заданных технологий не сочетается с формированием у них навыков совершенствования этих технологий, то у молодых специалистов не появится способность быстро адаптироваться к новым тенденциям и новым уровням развития информационных технологий. Для того, чтобы студентов и магистрантов можно было учить не только использовать, но и совершенствовать различные технологии, эти технологии должны быть открытыми. Можно выделить три уровня подготовки молодых специалистов по разработке сложных компьютерных систем:

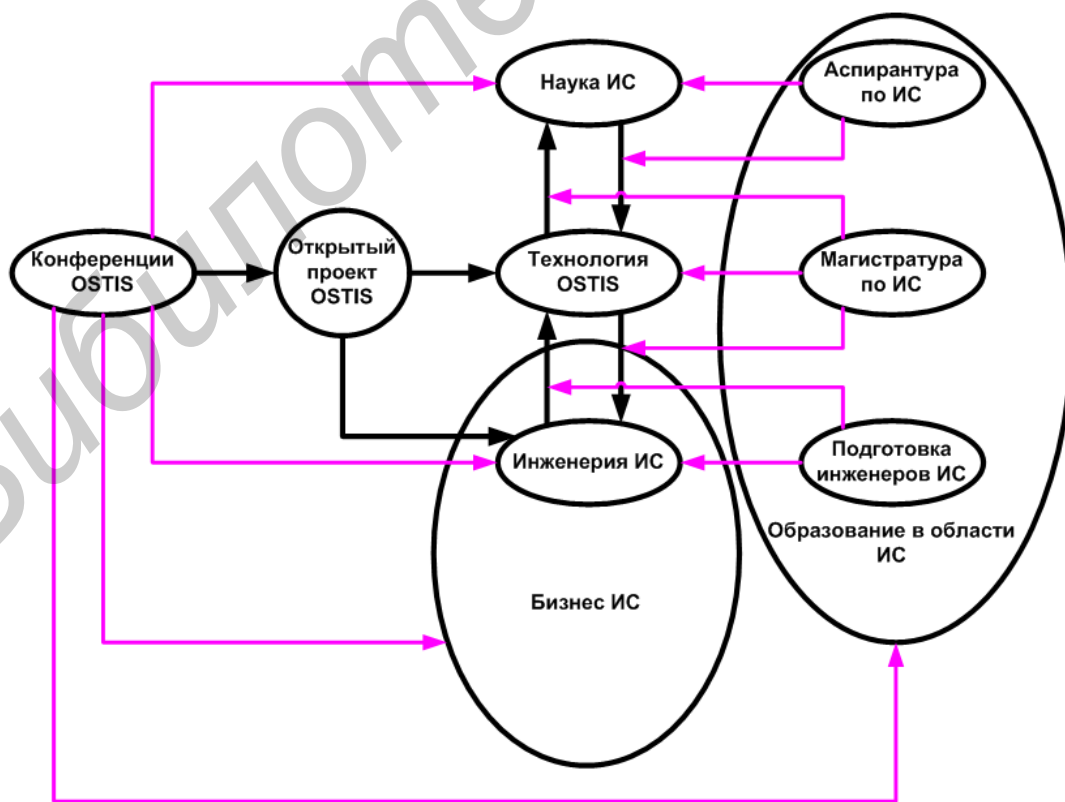


Рисунок 6 – Проект OSTIS как модель взаимодействия науки, образования, инженерии и бизнеса, направленного на развитие рынка совместимых компьютерных систем, управляемых знаниями

- (1) способность выполнять четко поставленные и не всегда простые индивидуальные проектные задания;
- (2) способность эффективно участвовать в коллективных решениях нечетко поставленных проектных задач;
- (3) способность эффективно участвовать в совершенствовании используемой технологии, как минимум, путем передачи в нее своего приобретенного проектного опыта в виде различных типовых решений.

#### 4. Актуальные направления применения компьютерных систем, управляемых знаниями

Как уже было отмечено, любую компьютерную систему можно построить по архитектуре систем, управляемых знаниями, что даже без всякой интеллектуализации решателей задач переведет эти системы на качественно новый уровень, позволяющий существенно увеличить темпы их совершенствования в ходе эксплуатации и обеспечить их полную семантическую совместимость. Есть целый ряд областей применения компьютерных систем, настоятельно требующих того, чтобы эти системы были построены на основе знаний. Прежде всего, это те области применения компьютерных систем, в которых качество, структуризация и легкая модифицируемость контента играет решающую роль. Перечислим некоторые из таких областей.

- Мир вещей
- Умный дом
- Интеллектуальные робототехнические системы различного назначения
- Системы ситуационного управления, интеллектуального управления
- Мир Интернета
- Персональные агенты (ассистенты), живущие в интернете и способные вступать во взаимодействие с другими интернет-ориентированными агентами
- Интеллектуальные справочные системы различного назначения, способные отвечать на любые вопросы и решать любые задачи в заданной предметной области:
  - web-сайты с семантически структурированным контентом и с полным решением проблемы поиска информации, содержащейся в этом сайте;
  - интеллектуальные справочные системы учебного назначения, способные выполнить роль консультанта по заданной учебной дисциплине. В частности, актуальным является создание комплекса семантически совместимых интеллектуальных справочных систем по всем школьным дисциплинам и по общеобразовательным дисциплинам технических вузов (теория множеств, теория графов, теория отношений,

абстрактная алгебра, математическая логика, теория алгоритмов и программ и др.);

- интеллектуальные help-системы (семантически структурированные руководства пользователей различных технических систем);
- порталы научных знаний, обеспечивающие постоянно совершенствуемую систематизацию и структуризацию научных знаний в соответствующих областях;
- семантически структурированный аналог Википедии, которая может полностью сохранить все существующие естественно-языковые статьи, сделав над ними семантически структурированную надстройку путем частичного или полного перевода этих статей на внутренний язык;
- семантически структурированные электронные научные журналы, актуализируемые соответствующими редакционными коллегиями, и автоматизирующие решение таких трудоемких задач, как рецензирование (в т.ч. и верификация доказательств) различных научных статей, которые должны быть представлены в формализованном виде (близком к внутреннему смысловому представлению знаний)
- Интеллектуальные корпоративные системы, управляемые знаниями
- Интеллектуальные системы автоматизации проектирования различных классов технических систем, управляемые знаниями, и способные к систематизации постоянно накапливаемого проектного опыта
- Интеллектуальные обучающие системы, включающие в себя соответствующие интеллектуальные справочные системы и имеющие возможности управлять деятельностью обучаемых, учитывая их индивидуальные особенности.

#### Заключение

Переход от традиционных принципов и технологий построения компьютерных систем, в которых доминируют программы, управляющие функционированием систем к принципам и технологиям, направленным на построение систем, в которых хранящиеся знания полностью управляют их деятельностью, инициируя активность различных агентов, осуществляющих обработку этих знаний, - неизбежен. Чем раньше мы станем активнее накапливать опыт в разработке систем, управляемых знаниями, и совершенствовать соответствующие технологии, тем конкурентоспособнее мы будем на следующем этапе развития информационных технологий. Опыт создания комплексной

технологии OSTIS и опыт разработки прикладных систем на ее основе создает необходимые предпосылки для начала перехода от традиционных компьютерных систем к системам, управляемым знаниями, в которых не всегда требуется умение решать интеллектуальные задачи. Процесс развития технологий проектирования систем, управляемых знаниями, должен быть перманентным, а фундаментом его должно быть активное сотрудничество науки, образования, инженерии и бизнеса.

## Библиографический список

[Андреева, 2006] Андреева О.А., Боровикова О.И., Булгаков С.В. и др. Археологический портал знаний: содержательный доступ к знаниям и информационным ресурсам по археологии // Тр. 10-й национальной конференции по искусственному интеллекту с международным участием КИИ-2006 (25–28 сентября 2006 г., Обнинск). М.: Физматлит, 2006. Т. 3. С. 832–840.

[Боровикова, 2006] Боровикова О.И., Загорюлько Ю.А., Загорюлько Г.Б. и др. Разработка портала знаний по компьютерной лингвистике // Тр. 11-й национальной конференции по искусственному интеллекту с международным участием КИИ-2008 (Дубна, 2008 г.). М.: ЛЕНАНД, 2008. Т. 3. С. 380–388.

[Гаврилова и др., 2007] Гаврилова, Т.А. Интеллектуальные технологии в менеджменте: инструменты и системы: Учеб. пособие / Т. А. Гаврилова, Д.М. Муромцев; Высшая школа менеджмента СПбГУ. – СПб.: Изд-во «Высшая школа менеджмента»; Издат. дом С.-Петерб. гос ун-та, 2007. – 488 с.

[Гладун и др., 2006] Гладун А.Я., Рогушина Ю.В. Онтологический подход к поиску веб-сервисов в распределенной среде Интернет // Информатика, Минск, № 4, 2006. – С.116-127.

[Гладун и др., 2013] Гладун А.Я., Рогушина Ю.В. Репозитории онтологий как средство повторного использования знаний для распознавания информационных объектов // Онтология проектирования, № 1 (7), 2013. – С.35-50.

[Глоба, 2014] Л.С. Глоба, Р.Л. Новогрудская, Подход к построению формальной алгебраической системы порталов знаний, Онтология проектирования. - 2014. - №2(11). – ISSN 2223-9537- С.40-59

[Голенков и др, 2001] Голенков, В.В. Представление и обработка знаний в графодинамических ассоциативных машинах /В. В. Голенков [и др.] – Мн. : БГУИР, 2001.

[Ефименко, 2011] Онтологическое моделирование экономики предприятий и отраслей современной России: Часть 1. Онтологическое моделирование: подходы, модели, методы, средства, решения: препринт WP7/2011/08 (ч.1) [Текст] / И.В. Ефименко, В.Ф. Хорошевский; Нац. Исслед. ун-т «Высшая школа экономики». – М.: Изд. дом Высшей школы экономики, 2011. – 76 с.

[Джанетто и др., 2005] Джанетто, К. Управление знаниями: Руководство по разработке и внедрению корпоративной системы управления знаниями / К. Джанетто, Э. Уилер. – М.: Добрая книга, 2005.

[Загорюлько, 2008] Загорюлько Ю.А., Боровикова О.И. Подход к построению порталов научных знаний // Автоматика. № 1, 2008, т. 44. –с. 100–110.

[Ивашенко, 2012] Ивашенко, В. П. Семантические модели и средства интеграции и отладки баз знаний / В. П. Ивашенко // Материалы международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем» – Минск, 2012. –С. 193-204

[Касьянов, 2003] Касьянов, В.Н. Графы в программировании: обработка, визуализация и применение/ В. Н. Касьянов, В. А. Евстигнеев // ВHV–Санкт-Петербург, 2003.–1104 с.

[Клещев и др., 2011] Клещев, А.С. Облачная платформа для разработки и управления интеллектуальными системами / А.С. Клещев, В.В. Грибова, Д.А. Крылов, Ф.М. Москаленко, С.В. Смагин, В.А. Тимченко, М.Б. Тютюнник, Е.А. Шалфеева // материалы междунар. научн.-техн. конф. OSTIS-2011 (Минск, 10-12 февраля 2011г.). – Минск: БГУИР, 2011. С. 5-14.

[Кузнецов, 2007] Семантико-ориентированные системы на основе баз знаний / И.П. Кузнецов, А.Г. Мацкевич. – М.: Связьиздат, 2007. – 173с.

[Левыкин и др., 2014] Левыкин, М.В. Паттерны проектирования требований к информационным системам: моделирование и применение / В.М. Левыкин, М.В. Евланов, М.А. Керносов. Харьков: ООО «Компанія «СМІТ»», 2014. – 320 с.

[Лукашевич, 2011] Лукашевич, Н.В. Тезаурусы в задачах информационного поиска. – М.: Издательство Московского университета, 2011. – 512 с.

[Рогушина, 2014] Рогушина Ю.В. Знание-ориентированные средства поддержки семантического поиска в Web. – LAP LAMBERT Academic Publishing. 2014. – 214 с. – ISBN 978-3-659-56520-5.

[Рубашкин, 2012] Рубашкин, В.Ш. Онтологическая семантика. Знания. Онтологии. Онтологически ориентированные методы информационного анализа текстов. – М.: ФИЗМАТЛИТ, 2012. – 348с.

[Тузовский и др., 2005] Тузовский, А.Ф. Системы управления знаниями (методы и технологии) / А.Ф. Тузовский, С.В. Чириков, В.З. Ямпольский. – Томск: Изд-во НТЛ, 2005. – 260 с.

[Хорошевский, 2009] Хорошевский В.Ф. Пространства знаний в сети Интернет и Semantic Web (ч. 2) // Искусственный интеллект и принятие решений. 2009. №4. С. 15-36.

[Шведин, 2010] Онтология предприятия: экспириентологический подход: Технология построения онтологической модели представления / Шведин Б.Я. – М.:Ленард, 2010. – 210с.

## SEMANTIC TECHNOLOGY OF COMPONENT DESIGN OF SYSTEMS, MANAGED BY KNOWLEDGES

Golenkov V.V., Guliakina N.A.

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

[golenkov@bsuir.by](mailto:golenkov@bsuir.by)

[guliakina@bsuir.by](mailto:guliakina@bsuir.by)

The article considers a result of five-year development of the Project OSTIS, oriented on the creation of Open semantic technology for intelligent systems design. The basis of that technology is the knowledge representation using semantic networks with theoretical-set interpretation. The article considers such system classes, as knowledge based systems and systems, managed by knowledges.