



УДК 004.822:514

### СЕМАНТИЧЕСКАЯ МОДЕЛЬ КОЛЛЕКТИВНОГО ПРОЕКТИРОВАНИЯ БАЗ ЗНАНИЙ

Давыденко И.Т.

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

ir.davydenko@gmail.com

В работе рассматривается унифицированная модель коллективного проектирования баз знаний. Описывается унифицированная структура базы знаний системы, управляемой знаниями. Приведена классификация пользователей систем, управляемых знаниями, а также типология действий разработчиков баз знаний.

**Ключевые слова:** база знаний, система, управляемая знаниями, проектирование баз знаний.

#### Введение

В настоящее время все более актуальной становится задача эффективного информационного обеспечения научной и производственной деятельности, связанная с бурным ростом объемов информации в различных отраслях знаний. Данная задача, как правило, рассматривается в контексте создания хранилищ знаний и их систематизации и структуризации с целью облегчения их обработки.

Ускорение и упрощение процесса разработки интеллектуальных систем различного назначения является одной из важнейших задач *Проекта OSTIS*, направленного на разработку открытой технологии проектирования *компьютерных систем, управляемых знаниями*.

**Компьютерная система, управляемая знаниями** – система, в основе которой лежит представленная унифицированным образом база знаний, содержащая в систематизированном виде всю информацию, используемую этой системой. [Голенков, 2015]. В рамках данной работы будем рассматривать понятие системы, управляемой знаниями, построенной по *Технологии OSTIS*, которую будем называть *ostis-системой*.

База знаний является одним из ключевых компонентов компьютерных систем, управляемых знаниями различного назначения [Гаврилова и др., 2001], [Гаврилова, 2008], [Хорошевский, 2008], [Кудрявцев, 2010], [Тузовский и др., 2005]. Разработка этого компонента является трудоемким и продолжительным процессом.

Под *базой знаний ostis-системы* будем понимать конечную информационную конструкцию,

являющуюся формальным представлением всех знаний, достаточных для функционирования некоторой компьютерной системы и хранимых в памяти этой системы. [Голенков, 2015].

В рамках данной работы рассмотрим унифицированную модель базы знаний любой *ostis-системы*, а также модель коллективной разработки *баз знаний ostis-систем*.

В качестве формальной основы проектируемых логико-семантических моделей баз знаний интеллектуальных систем используются графодинамические модели специального вида – семантические модели представления и обработки знаний, в основе которых лежат унифицированные семантические сети с базовой теоретико-множественной интерпретацией. Основным способом кодирования информации для таких сетей является SC-код (Semantic Code) [Голенков, 2012].

Некоторые из рассматриваемых в данной работе понятий будут сопровождаться фрагментами их спецификации на формальном языке SCn. [IMS, 2016]

Каждая *ostis-система* состоит из *sc-модели компьютерной системы*, не зависящей от платформы реализации этой модели, и *платформы интерпретации sc-моделей*, обеспечивающей интерпретацию какого-либо класса *sc-моделей*.

#### *ostis-система*

$\leq$  базовая декомпозиция\*:

- {
- платформа интерпретации *sc-моделей*
- *sc-модель компьютерной системы*
- }

В свою очередь, *sc-модель компьютерной системы* декомпозируется следующим образом:

#### **sc-модель компьютерной системы**

<= базовая декомпозиция\*:

- {
- *sc-модель базы знаний*
- *sc-модель машины обработки знаний*
- *sc-модель интерфейса компьютерной системы*
- }

В свою *sc-модель базы знаний* любой компьютерной системы, построенной по *Технологии OSTIS* имеет унифицированную структуру, которая будет подробнее рассмотрена далее.

Стоит отметить, что описываемые ниже разделы базы знаний отражают структуру базы знаний с точки зрения технологии разработки базы знаний.

## **1. Структура базы знаний систем, управляемых знаниями**

Рассмотрим структуру *sc-модели базы знаний ostis-системы*:

#### **sc-модель базы знаний**

= унифицированная семантическая модель базы знаний

∈ раздел базы знаний

<= базовая декомпозиция\*:

- {
- *предметная часть базы знаний*
- *контекст предметной части базы знаний в рамках Глобальной базы знаний*
- *документация компьютерной системы*
- *история и текущие процессы эксплуатации компьютерной системы*
- *история, текущие процессы и план развития компьютерной системы*
- }

*предметная часть базы знаний* содержит всю информацию о *предметной области* (или нескольких взаимосвязанных предметных областях в рамках этой же базы знаний) [Давыденко, 2013], для работы с которой предназначена та или иная система, управляемая знаниями (например - справочная система). Примерами разделов такого рода могут быть *Документация. Технология OSTIS* или *Документация. Геометрия Евклида*.

*контекст предметной части базы знаний в рамках Глобальной базы знаний* содержит спецификацию объектов, которые не исследуются непосредственно в предметной части базы знаний данной системы, но имеют к ней отношение, то есть, упоминаются при описании каких-либо понятий, исследуемых в предметной части базы знаний. Например, для системы *IMS* это могут быть такие понятия, как *искусственный интеллект* или *интеллектуальная система*, для системы по *Геометрии Евклида* – историческая справка о жизни Евклида, математики и т.д.

Раздел *документация компьютерной системы* содержит документацию самой *ostis-системы*, как минимум, спецификацию ее базы знаний, машины обработки знаний и интерфейса, а также все необходимые руководства, обеспечивающие возможность обучения работе с системой.

#### **история и текущие процессы эксплуатации компьютерной системы**

<= базовая декомпозиция\*:

- {
- *история эксплуатации компьютерной системы*
- *текущие процессы эксплуатации компьютерной системы*
- }

В разделе *история эксплуатации компьютерной системы* хранится история диалога системы с ее пользователями, то есть спецификация всех действий, выполненных (уже ставших *прошлыми сущностями*) *системой* с целью удовлетворения информационных потребностей пользователей, в том числе, ответов на вопросы и осуществления указанных ими преобразований в базе знаний, в том числе – последовательность выполнения этих действий и полученный результат. При необходимости, часть таких описаний может удаляться, например, по истечении какого-либо установленного срока.

В разделе *текущие процессы эксплуатации компьютерной системы* хранятся спецификации всех действий, выполняемых *ostis-системой* в данный момент (входящих во множество *настоящих сущностей*), а также все временные вспомогательные конструкции, сгенерированные *sc-агентами* в процессе работы и пока еще не удаленные. После выполнения указанных действий их знаки и спецификации переносятся в раздел *история эксплуатации компьютерной системы*.

#### **история, текущие процессы и план развития компьютерной системы**

<= базовая декомпозиция\*:

- {
- *структура и организация проекта компьютерной системы*
- *история развития компьютерной системы*
- *текущие процессы развития компьютерной системы*
- *план развития компьютерной системы*
- }

В разделе *структура и организация проекта компьютерной системы* описываются структура проекта, направленного на развитие *ostis-системы*, в том числе, указываются его подпроекты и роли разработчиков, ответственных за каждый проект.

В разделе *история развития компьютерной системы* находятся спецификации проектных действий, выполненных в процессе разработки системы (*прошлых сущностей*), с обязательным указанием исполнителей, последовательности и

результата выполнения.

В разделе *текущие процессы развития компьютерной системы* находятся спецификации утвержденных и инициированных проектных действий, выполняемых разработчиками системы в данный момент времени (*настоящих сущностей*), с обязательным указанием исполнителей, последовательности и цели выполнения, а также вся информация, описывающая предложения по редактированию *предметной части базы знаний и истории эксплуатации компьютерной системы* и их обсуждение администраторами, менеджерами и экспертами.

В разделе *план развития компьютерной системы* находятся спецификации проектных действий, которые утверждены к выполнению, но пока еще не выполняются по каким-либо причинам, а также вся информация, описывающая предложения по редактированию раздела *история, текущие процессы и план развития компьютерной системы* и их обсуждение администраторами, менеджерами и экспертами.

В соответствии с указанной структурой построена база знаний как самой системы IMS [IMS, 2016], так и всех прикладных систем, построенных по *Технологии OSTIS*.

## 2. Типология пользователей систем, управляемых знаниями

Процесс разработки *sc-модели базы знаний ostis-системы* по сути сводится к формированию разработчиками предложений по редактированию того или иного раздела базы знаний и последующему рассмотрению этих предложений администраторами, при необходимости - экспертами, а также в отдельных случаях - менеджерами соответствующих проектов.

Рассмотрим типологию пользователей *ostis-системы*.

Все пользователи любой *ostis-системы* делятся на *зарегистрированных пользователей\** и *незарегистрированных пользователей\**.

*пользователь\**

= *пользователь ostis-системы\**

∈ *бинарное отношение*

∈ *неролевое отношение*

≤ *разбиение\**:

- {
- *незарегистрированный пользователь\**
- *зарегистрированный пользователь\**
- }

*незарегистрированный пользователь\** - это *бинарное отношение*, связывающее *ostis-систему* и *sc-элемент*, обозначающий *персону*, не прошедшую процедуру регистрации в системе.

Незарегистрированный пользователь имеет доступ на чтение *предметной части базы знаний*

*ostis-системы*. Данный тип пользователей может работать с *ostis-системой* в режиме эксплуатации, т.е. может только задавать запросы, адресуемые *предметной части базы знаний* (т.е. предметные задачи).

*зарегистрированный пользователь\** - это *бинарное отношение*, связывающее *ostis-систему* и *sc-элемент*, обозначающий *персону*, прошедшую процедуру регистрации в системе.

Зарегистрированный пользователь имеет доступ на чтение всей базы знаний и внесение предложений ко всей базе знаний, может выполнять роль конечного пользователя *ostis-системы*, т.е. работать в режиме эксплуатации, а также роль ее разработчика. При этом независимо от роли, которую выполняет тот или иной пользователь, он может делать предложения по редактированию любой из частей базы знаний, которые в зависимости от его уровня будут либо приняты автоматически, либо будут отдельно рассматриваться.

Среди зарегистрированных пользователей выделен отдельный тип пользователей - *разработчик\**.

*разработчик\** - это *бинарное отношение*, связывающее какой-либо *раздел базы знаний ostis-системы* (в пределе - всю базу знаний) и *sc-элемент*, обозначающий *персону*, которая может быть разработчиком данного *раздела базы знаний*, т.е. выполнять проектные задачи в рамках данного раздела.

Помимо эксплуатации *ostis-системы разработчик\** может вносить предложения по изменению какой-либо части базы знаний, оставлять комментарии к предложениям по редактированию базы знаний. Среди разработчиков выделены такие роли, как *администратор\**, *менеджер\** и *эксперт\**.

*разработчик\**

=> *включение\**:

- *администратор\**
- *менеджер\**
- *эксперт\**

*администратор\** - это *бинарное отношение*, связывающее какой-либо *раздел базы знаний ostis-системы* (в общем случае - всю базу знаний) и *sc-элемент*, обозначающий *персону*, которая является администратором данного *раздела базы знаний*.

Задачами *администратора\** являются:

- контроль целостности и непротиворечивости всей базы знаний;
- определение уровней доступа других пользователей;
- принятие решения относительно принятия или отклонения предложений в различные части базы знаний, в том числе при необходимости отправка их на экспертизу;

• самостоятельное внесение изменений в различные части базы знаний, путем использования соответствующих команд редактирования (при этом изменения автоматически оформляются как предложения и заносятся в *историю развития ostis-системы*).

**менеджер\*** - это бинарное отношение, связывающее какой-либо раздел базы знаний *ostis-системы* (в общем случае – всю базу знаний) и *sc-элемент*, обозначающий *персону*, которая является менеджером данного раздела базы знаний.

Задачами **менеджера\*** являются:

- планирование объемов работ по проектированию базы знаний;
- детализация проектных задач на подзадачи, непосредственно формулирование проектных задач, назначение исполнителей проектных задач;
- установка приоритетов и сроков выполнения работ по проектированию базы знаний;
- контроль сроков выполнения проектных задач;

**менеджер\*** вносит изменения в часть соответствующего раздела, описывающую проектные задачи, используя соответствующие команды редактирования (при этом изменения автоматически оформляются как предложения и заносятся в раздел *проект ostis-системы. История, текущие процессы и план развития ostis-системы*). Таким образом, **менеджер\*** является **администратором\*** раздела *история, текущие процессы и план развития компьютерной системы*.

**эксперт\*** - это бинарное отношение, связывающее какой-либо раздел базы знаний *ostis-системы* (в общем случае – всю базу знаний) и *sc-элемент*, обозначающий *персону*, которая является экспертом данного раздела базы знаний.

Задачами **эксперта\*** являются:

- верификация и тестирование результатов выполнения проектных задач;
- при необходимости эксперт может оставлять комментарии к любому фрагменту базы знаний относительно его корректности. Все комментарии попадают в *план развития компьютерной системы*.

### 3. Типология действий разработчиков баз знаний

В процессе разработки *sc-модели базы знаний ostis-системы* каждый из пользователей, участвующий в разработке использует определенный набор команд. Каждой такой команде соответствует некоторый класс *действий в sc-памяти* [IMS, 2016]. Все такие действия объединены в общий класс **действие разработчика унифицированных семантических моделей баз знаний**.

Рассмотрим типологию таких действий:

#### **действие разработчика унифицированных семантических моделей баз знаний**

$\leq$  включение\*:

*действие в sc-памяти*

$\Rightarrow$  включение\*:

- действие эксперта базы знаний
- действие администратора базы знаний
- действие менеджера базы знаний
- действие разработчика базы знаний

### 3.1. Действия разработчика базы знаний

#### **действие разработчика базы знаний**

$\Rightarrow$  включение\*:

- действие. доработать предложение по редактированию базы знаний
- действие. построить новый фрагмент для включения в базу знаний

**действие разработчика базы знаний\*** может выполняться любым зарегистрированным пользователем\* *ostis-системы*, в том числе, при необходимости, ее администратором или менеджером.

#### **действие. сформировать предложение по редактированию базы знаний**

$\Rightarrow$  включение\*:

- действие. сформировать предложение проектного задания
- действие. сформировать предложение исполнителя проектного задания

$\leq$  включение\*:

*действие разработчика базы знаний*

Единственным аргументом **действия. сформировать предложение по редактированию базы знаний** является знак *структуры*, описывающей предлагаемое изменение в базе знаний (чаще всего – дополнение к имеющейся базе). Предполагается, что в случае утверждения данного предложения, данная структура будет интегрирована в соответствующий раздел базы знаний, и, при необходимости, будут инициированы *действия*, знаки которых входят в ее состав. Процесс формирования данной *структуры* может быть автоматизирован при помощи соответствующих команд.

Выполнение **действия. сформировать предложение по редактированию базы знаний** предполагает следующие изменения в базе знаний:

- генерируется связка отношения *автор\**, связывающая знак *структуры*, обозначающей предложение и знак пользователя, делающего предложение;
- генерируется и инициируется знак *действия. рассмотреть предложение по редактированию базы знаний, первым аргументом' (предложением\*)* которого становится знак *структуры*, описывающей предложение по редактированию базы знаний;
- если в базе знаний существует знак какого-либо *действия разработчика базы знаний* (или

более частного), *аргументом* которого является структура, описывающая вносимое предложение, то генерируется *связка отношения последовательность действий\**, связывающая знак этого действия разработчика базы знаний и знак действия. *рассмотреть предложение по редактированию базы знаний*. Факт наличия такой связи может облегчить задачу администратора по рассмотрению предложения. В противном случае считается, что задача на формирование указанной структуры поставлена не была, и разработчик принял решение о необходимости ее разработки самостоятельно. При этом указанное действие разработчика базы знаний становится *прошлой сущностью*.

Среди действий рядовых разработчиков выделен ряд классов действий, которые выполняются только разработчиками, имеющими дополнительные уровни ответственности, т.е. *администратор\**, *менеджер\** и *эксперт базы знаний\**.

**действие. утвердить предложение по редактированию базы знаний**  
= действие. одобрить предложение по редактированию базы знаний  
<= включение\*:

- действие эксперта базы знаний
- действие менеджера базы знаний
- действие администратора базы знаний

Единственным аргументом **действия. утвердить предложение по редактированию базы знаний** является знак действия. рассмотреть предложение по редактированию базы знаний.

Изменения в базе знаний могут быть различными, в зависимости от более частного типа аргумента **действия. утвердить предложение по редактированию базы знаний** и типа пользователя, выполняющего это действие.

Если пользователь является *администратором\** проекта соответствующего раздела базы знаний, то:

- генерируется *связка отношения утверждено\**, связывающая указанный аргумент и знак пользователя, утвердившего предложение;
- *структура*, описывающая предложение по редактированию базы знаний, интегрируется в базу знаний, при этом иницируются знаки *действий*, входящих в состав этой *структуры*;
- знак этой *структуры* с необходимой спецификацией заносится в раздел, описывающий историю эволюции базы знаний;
- аргумент **действия. утвердить предложение по редактированию базы знаний** становится *прошлой сущностью*.

Если пользователь является менеджером проекта соответствующего раздела базы знаний, и аргумент **действия. утвердить предложение по редактированию базы знаний** является действием. рассмотреть предлагаемого исполнителя проектной задачи то:

- генерируется *связка отношения утверждено\**, связывающая указанный аргумент и знак пользователя, утвердившего предложение;
- структура, описывающая предложение по редактированию базы знаний, интегрируется в базу знаний, при этом иницируются знаки действий, входящих в состав этой структуры;
- аргумент **действия. утвердить предложение по редактированию базы знаний** становится *прошлой сущностью*.

Во всех остальных случаях выполнение **действия. утвердить предложение по редактированию базы знаний** считается неправомерным (например, когда пользователь является рядовым разработчиком базы знаний).

**действие. отклонить предложение по редактированию базы знаний**  
<= включение\*:

- действие эксперта базы знаний
- действие менеджера базы знаний
- действие администратора базы знаний

Единственным аргументом **действия. отклонить предложение по редактированию базы знаний** является знак действия. рассмотреть предложение по редактированию базы знаний.

В зависимости от более частного типа аргумента **действия. отклонить предложение по редактированию базы знаний** и типа пользователя, выполняющего это действие, изменения в базе знаний могут быть различными.

Если пользователь является администратором проекта соответствующего раздела базы знаний, то:

- генерируется *связка отношения отклонено\**, связывающая указанный аргумент и знак пользователя, отклонившего предложение;
- знак этой *структуры* с необходимой спецификацией заносится в раздел, описывающий историю эволюции базы знаний;
- аргумент **действия. отклонить предложение по редактированию базы знаний** становится *прошлой сущностью*.

Если пользователь является менеджером проекта соответствующего раздела базы знаний, и аргумент **действия. отклонить предложение по редактированию базы знаний** является действием. рассмотреть предлагаемого исполнителя проектной задачи то:

- генерируется *связка отношения отклонено\**, связывающая указанный аргумент и знак пользователя, отклонившего предложение;
- аргумент **действия. отклонить предложение по редактированию базы знаний** становится *прошлой сущностью*.

Во всех остальных случаях выполнение **действия. отклонить предложение по редактированию базы знаний** считается неправомерным (например, когда пользователь является рядовым разработчиком базы знаний).

### 3.2. Действия администратора базы знаний

#### *действие администратора базы знаний*

=> включение\*:

- *действие. рассмотреть предложение по редактированию базы знаний*  
=> включение\*:
  - *действие. рассмотреть новое проектное задание*
  - *действие. рассмотреть результат верификации предложения*
- *действие. сформировать задание на верификацию предложения*
- *действие. утвердить результат верификации предложения*
- *действие. отклонить результат верификации предложения*

Первым аргументом *действия. сформировать задание на верификацию предложения* является знак *действия. рассмотреть предложение по редактированию базы знаний*. При необходимости может быть указан второй аргумент, являющийся знаком пользователя, который является экспертом, который должен, по мнению администратора, произвести верификацию заданного предложения.

Выполнение *действия. сформировать задание на верификацию предложения* предполагает следующие изменения в базе знаний:

- генерируется и иницируется знак *действия. верифицировать заданную структуру*, в качестве объекта\* для данного *действия* указывается знак того предложения, которое должно быть верифицировано;
- если был явно указан эксперт, который должен произвести верификацию, генерируется связка отношения *исполнитель\**, связывающая указанный знак *действия. верифицировать заданную структуру* и знак этого эксперта;
- генерируется связка отношения *поддействие\**, связывающая знак *действия. рассмотреть предложение по редактированию базы знаний* и указанный знак *действия. верифицировать заданную структуру*.

Единственным аргументом *действия. утвердить результат верификации предложения* является знак *действия. рассмотреть результат верификации предложения*.

Выполнение *действия. утвердить результат верификации предложения* предполагает следующие изменения в базе знаний:

- генерируется и иницируется знак *действия. доработать предложение по редактированию базы знаний, аргументом' (целью\*)* которого становится структура, описывающая новую версию предложения, которые было верифицировано экспертом. В качестве инициатора\* данного *действия* указывается администратор, от имени которого осуществляется выполнение *действия. утвердить результат верификации предложения*, в качестве исполнителя\* -

разработчик предыдущей версии этого предложения по редактированию базы знаний;

- генерируется связка отношения *утверждено\**, связывающая аргумент *действия. утвердить результат верификации предложения* и знак, обозначающий администратора, утвердившего результат верификации предложения;
- генерируется связка отношения *последовательность действий\**, связывающая аргумент *действия. утвердить результат верификации предложения* и упомянутый знак *действия. доработать предложение по редактированию базы знаний*;

- аргумент *действия. утвердить результат верификации предложения* становится *прошлой сущностью*;

- исходное *действие. рассмотреть предложение по редактированию базы знаний, аргументом' (целью\*)* которого была предыдущая версия предложения по редактированию базы знаний также становится *прошлой сущностью*.

Единственным аргументом *действия. отклонить результат верификации предложения* является знак *действия. рассмотреть результат верификации предложения*.

Выполнение *действия. отклонить результат верификации предложения* предполагает следующие изменения в базе знаний:

- генерируется связка отношения *отклонено\**, связывающая аргумент *действия. отклонить результат верификации предложения* и знак, обозначающий администратора, отклонившего результат верификации предложения;
- аргумент *действия. отклонить результат верификации предложения* становится *прошлой сущностью*.

### 3.3. Действия менеджера базы знаний

#### *действие менеджера базы знаний*

=> включение\*:

- *действие. сформировать предложение проектного задания*
- *действие. рассмотреть предлагаемого исполнителя проектной задачи*

<= включение\*:

*действие. рассмотреть предложение по редактированию базы знаний*

Единственным аргументом *действия. сформировать предложение проектного задания* является знак *структуры*, которую необходимо разработать в процессе выполнения данного проектного задания.

Выполнение *действия. сформировать предложение проектного задания* предполагает следующие изменения в базе знаний:

- генерируется знак *действия. построить новый фрагмент для включения в базу знаний, первым аргументом' (целью\*)* которого становится знак *структуры, структуры*, которую необходимо

разработать в процессе выполнения данного проектного задания;

- генерируется *структура*, содержащая весь фрагмент базы знаний, сформированный в предыдущем пункте, далее выполняются те же шаги, как если бы указанная структура была аргументом *действия*. *сформировать предложение по редактированию базы знаний*, с учетом того, что вместо класса *действие*. *рассмотреть предложение по редактированию базы знаний* используется более частный класс *действие*. *рассмотреть новое проектное задание*.

Каждое *действие*. *сформировать предложение исполнителя проектного задания* имеет два аргумента. Первый аргумент' является знаком действия, соответствующего проектному заданию, второй аргумент' – знаком зарегистрированного пользователя\*, который предлагается в качестве исполнителя.

Выполнение *действия*. *сформировать предложение исполнителя проектного задания* предполагает следующие изменения в базе знаний:

- генерируется *структура*, описывающая связь *действия*, соответствующего проектному заданию и исполнителя посредством отношения *исполнитель* \*;

- далее выполняются те же шаги, как если бы данная структура была аргументом *действия*. *сформировать предложение по редактированию базы знаний*, с учетом того, что вместо класса *действие*. *рассмотреть предложение по редактированию базы знаний* используется более частный класс *действие*. *рассмотреть предлагаемого исполнителя проектной задачи*.

### 3.4. Действия эксперта базы знаний

*действие эксперта базы знаний*

=> включение\*:

- *действие*. *верифицировать заданную структуру*
- *действие*. *отклонить верифицируемое предложение*
- *действие*. *утвердить верифицируемое предложение*
- *действие*. *сформировать задание на рассмотрение результата верификации предложения*

Единственным аргументом *действия*. *утвердить верифицируемое предложение* является знак действия. верифицировать заданную структуру.

Если пользователь является экспертом проекта соответствующего раздела базы знаний, то генерируется связка отношения *утверждено*\*, связывающая указанный аргумент и знак пользователя, утвердившего предложение, что говорит о том, что указанный эксперт не возражает против включения данного предложения в базу знаний.

Единственным аргументом *действия*.

*отклонить верифицируемое предложение* является знак действия. верифицировать заданную структуру.

Если пользователь является экспертом проекта соответствующего раздела базы знаний, то генерируется связка отношения *отклонено*\*, связывающая указанный аргумент и знак пользователя, утвердившего предложение, что говорит о том, что указанный эксперт возражает против включения данного предложения в базу знаний (без дополнительных комментариев).

Единственным аргументом *действия*. *сформировать задание на рассмотрение результата верификации* является знак структуры, описывающей результат верификации экспертом некоторого предложения по редактированию базы знаний. Такая структура обязательно включается в себя:

- знак самого этого предложения, а также знак новой версии данного предложения связанный со знаком исходного предложения связкой отношения *новая версия* \*;

- комментарий, *ключевым sc-элементом*' которого является знак этой связки, поясняющий, в чем собственно состоит разница между версиями предложения по редактированию базы знаний (что конкретно необходимо доработать).

Выполнение *действия*. *сформировать задание на рассмотрение результата верификации предложения* предполагает следующие изменения в базе знаний:

- генерируется и иницируется знак *действия*. *рассмотреть результат верификации предложения*, *объектом*\* которого становится *структура*, описывающей результат верификации экспертом соответствующего предложения;

- генерируется связка отношения *последовательность действий*\*, связывающая знак *действия*. *верифицировать заданную структуру* и упомянутый знак *действия*. *рассмотреть результат верификации предложения*.

## 4. Средства спецификации предложений по редактированию базы знаний

Рассмотрим ряд отношений, используемых для спецификации *действий разработчиков sc-моделей баз знаний* и *структур*, описывающих предложение по редактированию базы знаний.

*предложение*\* - это *бинарное отношение*, связывающее знак *действия*. *рассмотреть предложение по редактированию базы знаний* и знак той *структуры*, которая описывает данное предложение по редактированию, например, содержит фрагмент для включения в текущее состояние базы знаний.

*утверждено*\* - это *бинарное отношение*, связывающее знак *действия* по рассмотрению какого-либо предложения по редактированию базы

знаний и знак того пользователя *ostis-системы*, который утвердил (одобрил) данное *действие*. Как правило, это *администратор\** или *эксперт\** соответствующего проекта.

*отклонено\** - это *бинарное отношение*, связывающее знак *действия* по рассмотрению какого-либо предложения по редактированию базы знаний и знак того пользователя *ostis-системы*, который отклонил данное *действие*. Как правило, это *администратор\** или *эксперт\** соответствующего проекта.

*новая версия\** - это *бинарное отношение*, связывающее знак *структуры*, описывающей предложение по редактированию базы знаний и знак *структуры*, обозначающей версию этого предложения после доработки его исполнителем. Наличие такой связи упрощает и ускоряет повторную проверку предложения по редактированию базы знаний.

## Заключение

Рассмотренная в работе модель коллективной разработки баз знаний систем, управляемых знаниями, может быть применена при разработке такого рода систем любого назначения и сложности, поскольку рассчитана на иерархию разработчиков, соответствующую иерархии разделов базы знаний. Представленные в данной работе алгоритмы выполнения действий разработчиков базы знаний рассчитаны на одноуровневую иерархию разработчиков, но могут легко быть обобщены для случая, когда кроме администратора и менеджера всей базы знаний вводятся также роли администраторов и менеджеров ее разделов. В этом случае утверждение какого-либо предложения администратором раздела не приводит к интеграции предложения в соответствующий раздел, и, по сути, является аналогом утверждения предложения экспертом базы знаний или раздела.

Подробный демонстрационный диалог *ostis-системы* с разработчиками ее базы знаний можно найти на сайте системы IMS [IMS, 2016].

Работа выполнена при поддержке гранта БРФФИ-РФФИ-М «Методы и средства онтологического моделирования для семантических технологий проектирования интеллектуальных систем» (Ф15РМ-074, 15-57-04047).

## Библиографический список

- [Гаврилова и др., 2001] Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. Учебник / Гаврилова Т.А.. [и др.]; - СПб. : Изд-во «Питер», 2001.
- [Гаврилова, 2008] Гаврилова, Т. А. Визуальные методы работы со знаниями: попытка обзора / Т. А. Гаврилова, Н. А. Гулякина // Искусственный интеллект и принятие решений, 2008, № 1, С. 15-21
- [Голенков, 2012] Голенков, В.В., Гулякина Н.А. Графодинамические модели параллельной обработки знаний: принципы построения, реализации и проектирования. - В кн. Междунар. научн.-техн. конф. «Открытые семантические технологии проектирования интеллектуальных систем» (OSTIS-

2012). Материалы конф. [Минск, 16-18 февр. 2012 г.]. - Минск: БГУИР, 2012.

[Голенков, 2015] Голенков, В.В. Семантическая технология компонентного проектирования систем, управляемых знаниями. Открытые семантические технологии проектирования интеллектуальных систем (OSTIS-2015): материалы V Междунар. научн.-техн. конф./ В. В. Голенков, Н.А Гулякина// Мн.: БГУИР, 2015

[Давыденко, 2013] Давыденко, И.Т. Технология компонентного проектирования баз знаний на основе унифицированных семантических сетей. - В кн. Междунар. научн.-техн. конф. «Открытые семантические технологии проектирования интеллектуальных систем» (OSTIS-2013). Материалы конф. [Минск, 2013 г.]. - Минск: БГУИР, 2013.

[Кудрявцев, 2010] Кудрявцев Д.В. Системы управления знаниями и применение онтологий: Учеб. пособие / Д.В. Кудрявцев. - СПб.: Изд-во Политехн. ун-та, 2010.

[Тузовский и др., 2005] Тузовский А. Ф., Чириков С. В., Ямпольский В. З. (2005) Системы управления знаниями. — Томск : Изд-во науч.-техн. литературы.

[Хорошевский, 2008] Хорошевский, В.Ф. Пространства знаний в сети Интернет и Semantic Web (Часть 1) / В. Ф. Хорошевский // Искусственный интеллект и принятие решений. - 2008. - № 1. - С.80-97.

[IMS, 2016] Метасистема IMS.OSTIS [Электронный ресурс]. Минск, 2016. - Режим доступа: <http://ims.ostis.net/>. - Дата доступа: 15.01.2016.

## THE SEMANTIC MODEL OF KNOWLEDGE BASES COLLECTIVE DESIGN

Davydenko I.T.

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

[ir.davydenko@gmail.com](mailto:ir.davydenko@gmail.com)

The paper presents the unified model of the knowledge bases collective design. Describes the unified structure of the system's knowledge base, managed by knowledge. Shows the classification of users systems, managed by knowledge, and typology of the developers actions.

The model of the knowledge base, managed by knowledge, considered in this work, can be applied in the development of such systems for any purpose and complexity, because the hierarchy of developers corresponds to the hierarchy of knowledge base sections.