

УДК 004.03:004.9

ПРОЕКТ ОТКРЫТОЙ СЕМАНТИЧЕСКОЙ ТЕХНОЛОГИИ КОМПОНЕНТНОГО ПРОЕКТИРОВАНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ. ЧАСТЬ 2: УНИФИЦИРОВАННЫЕ МОДЕЛИ ПРОЕКТИРОВАНИЯ

В.В. Голенков¹, Н.А. Гулякина²

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

Аннотация

Статья является второй в цикле статей, посвященных рассмотрению открытого проекта, направленного на создание и развитие технологии компонентного проектирования интеллектуальных систем. В работе рассматривается унификация семантических моделей обработки знаний - моделей информационного поиска, моделей интеграции знаний, моделей решения задач, моделей трансляции семантических сетей во внешнее представление и обратно. На основе унифицированных семантических моделей интеллектуальных систем рассмотрена модель их компонентного проектирования, основанная на выделении многократно используемых компонентов интеллектуальных систем и на обеспечении платформенной независимости их проектирования. Рассмотрены также средства обеспечения открытого характера технологии проектирования интеллектуальных систем, методика их эволюционного проектирования и принципы построения метасистемы, предназначенной для комплексной поддержки проектирования интеллектуальных систем.

Ключевые слова: интеллектуальная система, технология проектирования интеллектуальных систем, унифицированная семантическая сеть, язык семантических сетей, компонентное проектирование интеллектуальных систем, графовый язык программирования, интеллектуальная метасистема.

Введение

Данная статья является продолжением работы [1], посвященной рассмотрению открытого проекта, направленного на создание и развитие технологии компонентного проектирования интеллектуальных систем (ИС).

1 Унификация семантических моделей обработки знаний

На основе *унифицированных семантических сетей* (sc-текстов) можно уточнить понятие унифицированной модели обработки информации, а также понятие унифицированной модели решения задач.

Все указанные абстрактные модели будем называть *sc-моделями обработки знаний* или *sc-машинами*, поскольку в основе их лежит использование *SC-кода*. Каждая такая модель (sc-машина) представляет собой многоагентную систему, состоящую:

- 1) из графодинамической памяти, в которой хранятся и обрабатываются тексты SC-кода – такую память будем называть *sc-памятью*;
- 2) из коллектива агентов, работающих над общей для них sc-памятью и взаимодействующих между собой только через эту память – такие агенты будем называть *sc-агентами*.

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

Каждый sc-агент реагирует на соответствующий ему класс ситуаций и/или событий, происходящих в sc-памяти, и осуществляет определенное преобразование sc-текста, находящегося в семантической окрестности обрабатываемой ситуации и/или события. Типология sc-агентов достаточно богата. В частности, можно выделить следующие классы sc-агентов, обеспечивающих:

- интерпретацию программ различных sc-языков программирования высокого уровня;
- информационный поиск;
- реализацию правил логического вывода, соответствующих самым различным логическим исчислениям;
- сведение задач к подзадачам;
- анализ качества хранимой базы знаний, в частности, ее корректности, полноты;
- обнаружение и автоматическое склеивание синонимичных sc-элементов;
- автоматическое устранение ошибок в базе знаний;
- удаление информационного мусора (в частности, удаления фрагментов базы знаний, которые редко востребованы и могут быть достаточно легко восстановлены в случае их отсутствия);
- трансляцию вводимой информации с различных внешних языков в SC-код;
- трансляцию sc-текстов, вводимых пользователю на различные внешние языки, а также классы рецепторных и эффекторных sc-агентов.

В понятии sc-машины набор агентов не фиксируется, т.е. могут существовать разные sc-машины с разным набором sc-агентов. Разные sc-машины можно интегрировать. С формальной точки зрения это сделать не очень сложно:

- 1) интегрировать sc-текст, описывающий текущее состояние взаимодействия sc-агентов одной sc-машины, с аналогичным sc-текстом другой sc-машины;
- 2) полученный интегрированный sc-текст поместить в sc-память интегрированной sc-машины;
- 3) в интегрированную sc-машину включить все sc-агенты первой интегрируемой sc-машины и все sc-агенты второй интегрируемой sc-машины.

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

В абстрактных sc-машинах можно выделить следующие языки программирования.

- Семейство sc-языков программирования высокого и сверхвысокого уровня (как процедурных, так и непроцедурных). Тексты программ этих языков хранятся в базе знаний ИС и описывают способы решения различных классов задач в соответствующих предметных областях.
- Базовый sc-язык программирования (язык SCP), на котором описываются sc-агенты и интерпретации sc-языков программирования высокого и сверхвысокого уровня, а также sc-операции, обеспечивающие интерпретацию различных логических исчислений, различных моделей интеллектуального решателя задач.
- Язык программирования, на котором описывается интерпретатор базового sc-языка программирования. Этот язык, в частности, может использоваться как язык микропрограмм для sc-компьютера, обеспечивающего аппаратную интерпретацию базового sc-языка программирования (языка SCP).

На основе понятия абстрактной *sc*-машины можно уточнить понятие унифицированной логико-семантической модели ИС. Такую унифицированную модель ИС будем называть абстрактной *sc*-моделью интеллектуальной системы, которая включает в себя:

- интегрированную совокупность всех знаний, которые необходимы для функционирования ИС и которые представлены в виде интегрированного *sc*-текста (такую семантическую модель базы знаний будем называть *sc*-моделью базы знаний или *sc*-текстом базы знаний);
- абстрактную *sc*-машину, в памяти которой хранится указанный *sc*-текст базы знаний.

Нетрудно заметить, что *sc*-текст базы знаний ИС является формальным и унифицированным уточнением того, что должна знать ИС, а *sc*-машина ИС и, в первую очередь, набор входящих в её состав *sc*-агентов является формальным и унифицированным уточнением того, что должна ИС уметь делать со своими знаниями.

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

Заметим, что абстрактную логико-семантическую модель в принципе можно построить для любой компьютерной системы (как для интеллектуальной системы, так и для компьютерной системы традиционного вида), обеспечивая их семантическую совместимость на абстрактном логико-семантическом уровне.

1.1 Унификация семантических моделей информационного поиска

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

Ассоциативный доступ – это доступ, основанный не на знании того, где находится искомая (требуемая) информация (в частности, на знании адреса или имени соответствующей области памяти), а на знании того, как искомая информация связана с известной информацией, хранимой в памяти, т.е. на знании некоторой спецификации искомой информации.

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

Унифицированная семантическая модель информационного поиска, которую будем называть *sc*-моделью информационного поиска, включает в себя:

- 1) *SC*-язык вопросов, с помощью которого в виде *sc*-текстов осуществляется описание (спецификация) запрашиваемых фрагментов всего интегрированного *sc*-текста, хранимого в текущий момент в *sc*-памяти (*sc*-текста, который является *sc*-моделью базы знаний). Тексты, принадлежащие *SC*-языку вопросов, будем называть *sc*-вопросами или *sc*-запросами.
- 2) *SC*-язык описания ответов, с помощью которого осуществляется явное выделение *sc*-текстов, являющихся ответами, и явное описание их связи с явно выделенными *sc*-текстами, которые представляют вопросы, соответствующие указанным ответам.
- 3) Семейство информационно-поисковых *sc*-агентов, каждый из которых реагирует на соответствующий ему тип *sc*-вопроса (который при этом должен быть инициирован) и выполняет соответствующую поисковую процедуру в *sc*-памяти.

Семантическая типология вопросов является предметом отдельного рассмотрения. Приведем фрагмент такой типологии, чтобы проиллюстрировать семантическую мощьность SC-языка вопросов. Прежде всего, по аналогии с логическими формулами множество вопросов разбивается на:

- атомарные вопросы;
- неатомарные вопросы, каждый из которых представляет собой конечное множество вопросов.

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

Поскольку в общем случае вопросу может соответствовать несколько правильных ответов, множество вопросов разбивается на вопросы, запрашивающие:

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

Специальным видом неатомарных вопросов являются вопросы, запрашивающие не сами правильные ответы на некоторый вопрос (который может быть как атомарным, так и неатомарным, и который является единственным компонентом сколько-вопроса), а количество таких правильных ответов.

Приведем некоторые типы атомарных вопросов.

- Запрос фрагментов базы знаний, изоморфных заданному образцу. Образец может иметь произвольный размер, произвольную конфигурацию и может быть представлен логической формулой существования, в которой квантор существования действует на логическую формулу произвольного вида. Поиск ответов на запросы рассматриваемого вида заключается в поиске знаков таких объектов, которые заданным образом связаны с другими объектами, которые могут быть и известными, и неизвестными. Другими словами, осуществляется поиск знаков таких объектов, которые удовлетворяют заданным требованиям.
- Запрос всех элементов заданного конечного множества (чаще всего – это множество из элементов некоторой структуры).
- Запрос внешней информационной конструкции, представленной файлом в том или ином формате.
- Запрос полного текста заданного высказывания.
- Запрос фактов истинности или ложности заданного высказывания в рамках заданной формальной теории.
- Запрос выбора альтернатив, запрашивающий одно или несколько истинных высказываний из заданного множества высказываний.
- Запрос обоснования (доказательство) истинности заданного высказывания.
- Запрос основных сведений об указываемом объекте. Фактически, речь идет о выделении из базы знаний семантической окрестности, «центром» которой является знак указываемого объекта. Примерами такого объекта служат понятие, предметная область, формальная теория, высказывание, любая структура, материальный объект.

- Запрос общих свойств объектов, принадлежащих заданному классу.
- Запрос идентифицирующих признаков заданного объекта. Здесь запрашиваются фрагменты базы знаний, каждый из которых однозначно определяет заданный объект. Если заданным объектом является понятие, то таким идентифицирующим признаком является либо определение этого понятия, либо соответствующая теорема о необходимости и достаточности.
- Запрос связей между заданными объектами.
- Запрос сравнительного анализа заданных объектов.
- Запрос сходств заданных объектов (сходства, аналогии – это частный вид связей между объектами).
- Запрос отличий заданных объектов (отличия объектов – это тоже частный вид связей между ними).
- Запрос плана решения заданной конкретной задачи, т.е. плана достижения заданной цели в заданных конкретных условиях.
- Запрос обобщенного способа решения любой задачи из заданного класса задач. Таким обобщенным способом может быть алгоритм, декларативная программа, нестрогое предписание (рекомендация).
- Запрос надцели, которая соответствует заданной цели.

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

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

1.2 Унификация семантических моделей интеграции знаний

На основе унифицированных семантических сетей предлагается построение унифицированных семантических моделей интеграции знаний [2] и использование этих моделей, во-первых, как основы процесса приобретения ИС новых знаний (как со стороны конечных пользователей, так и со стороны разработчиков); во-вторых, как основы интеграции программ и различных семантических моделей решения задач; в-третьих, как основы интеграции абстрактных логико-семантических моделей ИС.

Процесс интеграции двух семантических сетей представляет собой систему следующих взаимодействующих процессов:

- приведения интегрируемых семантических сетей к унифицированному виду, т.е. представление их в SC-коде;
- согласования ключевых узлов и онтологий, используемых в интегрируемых sc-текстах. Очевидно, что полностью автоматизировать такое согласование невозможно. Поэтому разработчикам интегрируемых фрагментов баз знаний и целых баз знаний необходимо уметь договариваться друг с другом;

- выделения в интегрируемых sc-текстах таких sc-элементов, которые имеют глобальные (уникальные) идентификаторы;
- выделения в интегрируемых sc-текстах sc-элементов, имеющих локальные идентификаторы вместе с областью действия каждого такого идентификатора. Область действия локального идентификатора – это такой фрагмент базы знаний, в рамках которого разные sc-элементы, имеющие этот локальный идентификатор, считаются синонимичными;
- склеивания sc-элементов, имеющих одинаковые глобальные идентификаторы;
- склеивания sc-элементов, имеющих одинаковые локальные идентификаторы, если каждый их этих sc-элементов принадлежит области действия своего локального идентификатора и области действия локального идентификатора другого sc-элемента;
- склеивания sc-элементов на основании однозначности используемых алгебраических операций;
- склеивания sc-элементов на основании логических высказываний о существовании единственности;
- склеивания кратных связей, принадлежащих отношениям у которых кратные связи отсутствуют либо всегда, либо при определенных четко оговоренных условиях.

Таким образом, интеграция семантических сетей, т.е. процесс погружения (понимания) одной семантической сети в другую – это нетривиальный процесс рассуждений, направленный на выявление пар синонимичных элементов семантической сети на основе определенных знаний, имеющихся в базе знаний ИС.

1.3 Унификация семантических моделей решения задач

В рамках проектируемой ИС предлагается обеспечить использование не только знаний различных видов, но и различных моделей и стратегий решения задач.

Для этого необходимо акцентировать внимание не только на разработку новых моделей решения задач, но и на унификацию и интеграцию в рамках проектируемых ИС уже разработанных и хорошо зарекомендовавших себя моделей (дедуктивных, индуктивных, абдуктивных, чётких, нечётких, универсальных, специализированных и т.д.) [3]. Подчеркнем то, что в разных проектируемых интеллектуальных системах могут быть востребованы разные сочетания известных моделей и стратегий решения задач. Подавляющее число моделей представления знаний и решения задач не являются альтернативными и дополняют друг друга. Не составляют исключение и такие классы моделей, как фреймовые, логические, продукционные.

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

Запросы могут инициироваться как пользователями, так и самой системой. Это означает, что в процессе обработки информации ИС сама себе может задавать (генерировать, порождать) запросы. Если инициирован некоторый sc-запрос, то сначала активизируются соответствующие агенты информационного поиска в «надежде» на то, что запрашиваемый ответ

(или ответы) на указанный sc-запрос уже присутствует в текущем состоянии базы знаний. И только после того, как информационно-поисковые sc-агенты обнаружат отсутствие ответа в текущем состоянии базы знаний, начинается работа решателя задач, направленная на генерацию требуемого ответа.

Кроме запроса используется также и процедурная формулировка информационной цели – это описание (спецификация) некоторого действия, которое требуется выполнить, и которое направлено на преобразование базы знаний, хранимой в памяти. Указанное действие, выполняется либо одним sc-агентом (в случае, если это элементарное действие над sc-памятью), либо несколькими sc-агентами и порождает определенное событие (изменение состояния sc-памяти).

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

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

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

1.4 Унификация семантических моделей пользовательских интерфейсов

Пользовательский интерфейс интеллектуальной системы, построенной на основе предлагаемой технологии, предлагается рассматривать как специализированную интеллектуальную систему, построенную по той же технологии и предназначенную для трансляции адресуемых пользователю сообщений с внутреннего абстрактного семантического языка представления знаний (SC-кода) на тот или иной внешний язык, тексты которого отображаются пользователю в удобном для него виде, а также для трансляции пользовательских сообщений с внешнего языка на внутренний семантический язык ИС (т.е. в SC-код) [4].

Трактовка пользовательских интерфейсов как ИС и унификация семантических моделей таких систем дает возможность:

- 1) унифицировать проектирование пользовательских интерфейсов;
- 2) легко наращивать возможности пользовательских интерфейсов;

3) неограниченно использовать базу знаний ИС для семантического анализа и понимания вводимой пользователем информации (в частности, естественно-языковых текстов).

В качестве основы организации графического пользовательского интерфейса предлагается использовать способ унифицированного визуального представления абстрактных унифицированных семантических сетей в виде изоморфных им графических конструкций.

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

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

Трактовка элементов управления пользовательским интерфейсом как элементов отображаемого на экране *SCg*-текста позволяет:

- 1) унифицировать представление любой информации, отображаемой на экране;
- 2) унифицировать способы инициирования различных запросов, касающихся любой отображаемой на экране информации (в том числе, и элементов управления);

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

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

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

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

Более подробно различные языки внешнего представления абстрактных *sc*-текстов вместе с большим количеством примеров рассмотрены в работах [5, 6].

2 Компонентное проектирование интеллектуальных систем

В целях ускорения процесса проектирования ИС предлагается создать общую библиотеку многократно используемых семантически совместимых компонентов интеллектуальных

систем, на основе которой разработать методику модульного (компонентного, сборочного) проектирования ИС.

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

- Библиотека многократно используемых компонентов баз знаний. Прежде всего, в эту библиотеку входят различные по содержанию, но семантически совместимые онтологии. Кроме того, сюда относятся различные «джентльменские наборы» знаний, которыми должны владеть «образованные» ИС. К таким знаниям, в частности, относятся базовые знания по арифметике, базовые знания по теории множеств (каждая ИС должна, по крайней мере, отличать элемент заданного множества от его подмножества), базовые знания по теории отношений (каждая ИС должна уметь отличать бинарное отношение от многоместного отношения, должна понимать, что такое соответствие), базовые знания по логике (каждая ИС должна понимать, что такое теория, высказывание, определение, переменная, должна отличать фактографическое высказывание от высказывания, не являющегося фактографическим, должна уметь отличать высказывание от логической формулы, не являющейся высказыванием) и многие другие знания, востребованность которых может быть самой разной.
- Библиотека компонентов семантических моделей информационного поиска. Это, прежде всего, различные информационно-поисковые агенты.
- Библиотека компонентов семантических моделей интеграции знаний и машин обработки знаний.
- Библиотека интерпретаторов программ, соответствующих различным языкам программирования.
- Библиотека различных стратегий решения задач, различных моделей решения задач и агентов, входящих в состав таких моделей.
- Библиотека компонентов пользовательских интерфейсов.

Все компоненты, включаемые в состав общей библиотеки компонентов ИС, оформляются как компоненты интеллектуальной собственности (*intellectual property*).

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

Для обеспечения семантической совместимости таких компонентов ИС, которые являются унифицированными семантическими моделями (sc-моделями знаний, sc-моделями машин обработки знаний, sc-агентов, sc-моделями ИС), необходимо, во-первых, согласовать семантику (смысл) всех используемых ключевых узлов и, во-вторых, согласовать глобальные идентификаторы ключевых узлов, используемых в разных компонентах. После этого интеграция всех компонентов, входящих в состав библиотеки, и в любых комбинациях осуществляется автоматически, без вмешательства разработчика.

2.1 Выделение легко расширяемых минимальных конфигураций для различных компонентов интеллектуальных систем

Минимальные конфигурации (ядра) компонентов ИС обеспечивают семантическую, логическую или функциональную полноту этих компонентов, но пренебрегают производительностью этих компонентов или предоставляемым пользователем комфортом. При этом для каждого такого компонента разрабатывается иерархическая система направлений и уровней его расширения, обеспечивающих повышение производительности и/или пользовательского комфорта.

К числу компонентов, для которых выделяется минимальная конфигурация (ядро) относятся:

- различные варианты (форматы) кодирования текстов SC-кода в памяти современных компьютеров;
- различные языки внешнего представления текстов SC-кода (SCg-код, SCs-код, SCn-код);
- различные языки программирования, построенные на базе SC-кода (в частности, базовый графовый язык программирования SCP) [5, 7];
- пользовательские интерфейсы ИС.

2.2 Платформенная независимость проектирования интеллектуальных систем

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

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

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

Полностью построенная абстрактная логико-семантическая модель проектируемой ИС:

- 1) является открытой, поскольку ее можно легко пополнять новыми знаниями и навыками, интегрируя их в текущую версию модели;
- 2) концентрирует внимание на семантические аспекты функционирования ИС и не содержит никаких лишних деталей, обусловленных тем или иным способом ее технической реализации (интерпретации);
- 3) является абстрактным инвариантом целого множества различных способов ее технологической реализации (в том числе и с помощью принципиально новых компьютеров).

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

Если разработчиков ИС что-то не устраивает в выбранном варианте интерпретации sc-моделей (в частности, производительность), должна существовать достаточно продуманная

методика совершенствования выбранного варианта интерпретатора sc-моделей интеллектуальных систем. Очевидно, что для каждого варианта интерпретации sc-моделей ИС указанная методика будет иметь свои особенности.

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

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

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

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

Вопросы программной реализации и, в частности, web-ориентированной реализации унифицированных логико-семантических моделей ИС рассмотрены в работе [8].

2.3 Ориентация на разработку новых платформ, основанных на аппаратной поддержке логико-семантических моделей интеллектуальных систем

Предлагается обеспечить возможность реализации унифицированных логико-семантических моделей ИС на семантических ассоциативных параллельных компьютерах, специально ориентированных на аппаратную реализацию таких моделей.

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

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

- 1) в таких компьютерах принципиально важна поддержка именно параллельной обработки знаний.
- 2) опыт использования параллельных компьютеров показывает, что эффективное их использование предполагающее разработку качественных параллельных программ требует особой профессиональной подготовки и высокой квалификации. Мир параллельного программирования требует особой культуры, особого стиля мышления. Еще более серьезная профессиональная подготовка требуется для разработки параллельных программ, ориентированных на обработку знаний и использующих ассоциативный доступ к обрабатываемой информации.
- 3) уровень развития микроэлектронных технологий сейчас позволяет достаточно быстро реализовывать самые смелые компьютерные архитектуры и модели обработки информации. Необходима только четкая постановка задачи.

4) созданию параллельных компьютеров для обработки знаний должно предшествовать создание технологии проектирования ИС, в основе которой лежат те модели параллельной обработки знаний, которые будут аппаратно поддерживаться в указанных компьютерах. Иначе мы получим «грудю» талантливо сделанного «железа», эффективность использования которого будет достаточно низкой. Это главная причина неудач такого рода проектов.

Предлагаемая технология проектирования ИС как раз и предполагает последовательное выполнение следующих этапов:

- разработка технологии проектирования абстрактных унифицированных логико-семантических моделей ИС;
- разработка нескольких вариантов программной реализации абстрактных унифицированных логико-семантических моделей ИС, выполненных на современных компьютерах;
- эксплуатация достаточно большого количества прикладных ИС и совершенствование технологии проектирования ИС на основе приобретенного опыта.

И только после этого предлагается начать разработку семантического ассоциативного компьютера, появление которого не отменит абсолютно ничего, сделанного ранее. Просто появится еще один, но уже аппаратный вариант реализации абстрактных унифицированных логико-семантических моделей интеллектуальных систем, применение которого для уже разработанных абстрактных унифицированных логико-семантических моделей различных прикладных систем для конечных пользователей этих ИС абсолютно ничего не изменит, кроме существенного повышения быстродействия.

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

Архитектуру аппаратной реализации семантических моделей обработки знаний можно рассматривать как иерархию абстрактных машин, описывающих переход от агентов, имеющих доступ ко всей семантической памяти, к агентам, имеющим доступ только к своей семантической окрестности и, в конечном счете, взаимодействующим только со своими семантическими соседями.

Аппаратная интерпретация абстрактных sc-машин предполагает создание реконфигурируемой памяти с распределенными в ней процессорными элементами. Такую интеграцию памяти и процессора будем называть процессоро-памятью. Реконфигурируемость (структурная перестраиваемость) памяти может быть обеспечена коммутационной средой для процессорных элементов. Можно рассматривать целый ряд подходов к реализации реконфигури-

руемой семантической ассоциативной процессоро-памяти. В частности, процессорным элементам можно ставить в соответствие узлы обрабатываемых унифицированных семантических сетей, а коммутируемым каналам связи между процессорными элементами – коннекторы этой семантической сети. В этом случае текущее состояние конфигурации коммутируемых каналов связи будет полностью соответствовать текущему состоянию конфигурации обрабатываемой семантической сети. Следовательно, память «превращается» из пассивного хранилища байтов в коммутационную среду между процессорными элементами.

2.4 Унификация семантических моделей встроенных интеллектуальных систем

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

- предметной (основной) ИС;
- интеллектуального пользовательского интерфейса;
- интеллектуальной подсистемы адаптивного управления диалогом с конечным пользователем;
- интеллектуальной help-системы для информационного обслуживания и обучения конечных пользователей предметной ИС, которые, начиная работать с системой, не обязаны иметь сразу высокую квалификацию;
- ИС управления проектированием ИС, которая координирует деятельность разработчиков предметной ИС [9];
- ИС управления информационной безопасностью предметной ИС.

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

Таким образом, проектируя каждую ИС, необходимо одновременно проектировать:

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

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

3 Доступность и открытость технологии проектирования интеллектуальных систем

Предлагается обеспечить максимально возможное расширение контингента разработчиков интеллектуальных систем, использующих предлагаемую технологию проектирования ИС, за счет максимальной доступности этой технологии и открытого характера её развития.

Доступность технологии проектирования ИС прежде всего означает то, что в её основе должна лежать такая модель представления и обработки знаний, которая была бы комфортна не только для средств технической реализации этой модели, но и комфортна для человека (для конечного пользователя и разработчика).

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

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

Завершая рассмотрение открытого характера предлагаемой технологии проектирования ИС, сделаем следующие замечания.

- Открытый характер технологии не является препятствием для реализации коммерческих интересов, связанных с этой технологией. Например, на коммерческой основе могут создаваться и предоставляться:
 - различные прикладные ИС;
 - некоторые варианты реализации различных многократно используемых (типовых) компонентов ИС;
 - некоторые варианты реализации интерпретатора абстрактных семантических логико-семантических моделей ИС, в частности, различные варианты построения семантических ассоциативных параллельных компьютеров.
- Открытый характер технологии, при грамотном использовании фактора её открытости, способствует обеспечению информационной безопасности, как самой технологии, так и прикладных интеллектуальных систем, созданных на её основе.
- Открытый характер предлагаемой технологии проектирования ИС может быть эффективно реализован только на базе технологии облачных вычислений, в рамках которой вся предлагаемая технология проектирования ИС рассматривается как некий интернет-сервис [10].

Рефлексивность интеллектуальных систем, разрабатываемых по предлагаемой технологии, означает то, что в базу знаний каждой ИС, разрабатываемой по предлагаемой технологии OSTIS (*Open Semantic Technology for Intelligent Systems*) [11], включается полная документация этой ИС (полное описание себя самой).

Наличие такой документации обеспечивает:

- качественное информационное обслуживание конечных пользователей по вопросам эксплуатации ИС;
- качественное информационное обслуживание разработчиков, которые осуществляют перманентное совершенствование ИС в ходе её эксплуатации;
- возможность адекватной самооценки (оценки своих возможностей при планировании собственных действий);

- возможность конструктивного анализа собственной деятельности и выявления своих недостатков.

4 Методика эволюционного проектирования интеллектуальных систем

В рамках технологии OSTIS предлагается использовать методику поэтапного эволюционного проектирования ИС.

Указанная методика предполагает:

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

С формальной точки зрения проектирование унифицированной логико-семантической модели (sc-модели) ИС в конечном счете сводится к проектированию sc-модели базы знаний этой интеллектуальной системы, поскольку scr-программы, описывающие поведение sc-агентов, можно рассматривать как часть базы знаний. Таким образом, проектируемая база знаний включает в себя:

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

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

В целом начальную стадию проектирования всей ИС на основе предлагаемой технологии можно условно разбить на следующие четыре этапа.

1) Разработка 1-й версии ИС, которая включает в себя:

- 1-ю версию её базы знаний;
- типовое ядро интеллектуальной информационно-поисковой машины, которое входит в состав библиотеки многократно используемых компонентов ИС;
- типовое ядро интеллектуального решателя, которое входит в состав библиотеки многократно используемых компонентов ИС;
- типовое ядро пользовательского интерфейса, которое входит в состав библиотеки многократно используемых компонентов ИС. Разработанная 1-я версия ИС уже обладает определенной целостностью, её можно тестировать и запускать в предварительную опытную эксплуатацию.

2) Разработка 2-й версии ИС, которая включает в себя:

- 2-ю версию её базы знаний;
- 1-ю версию её информационно-поисковой машины;
- типовое ядро её интеллектуального решателя;
- типовое ядро её пользовательского интерфейса;

- 3) Разработка 3-й версии ИС, включающей в себя:
- 3-ю версию её базы знаний;
 - 2-ю версию её информационно-поисковой машины;
 - 1-ю версию её интеллектуального решателя;
 - типовое ядро её пользовательского интерфейса;
- 4) Разработка 4-й версии ИС, включающей в себя:
- 4-ю версию её базы знаний;
 - 3-ю версию её информационно-поисковой машины;
 - 2-ю версию её интеллектуального решателя;
 - 1-ю версию её пользовательского интерфейса.

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

Более подробно методика эволюционного коллективного проектирования унифицированных семантических моделей интеллектуальных систем, на основе содержательной структуризации знаний, описана в работе [12].

5 Реализация технологии проектирования интеллектуальных систем в виде интеллектуальной метасистемы

Предлагается реализовать рассматриваемую технологию как *интеллектуальную метасистему* [13, 14], ориентированную на поддержку проектирования ИС, построенную по тем же самым принципам (т. е. по той же технологии), что и интеллектуальные системы, разрабатываемые на её основе. Эту метасистему назовем IMS (*Intelligent MetaSystem*).

Метасистема IMS должна включать в себя:

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

В метасистеме IMS можно выделить целый ряд подсистем, ориентированных на поддержку проектирования различных компонентов ИС, таких, как:

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

В интеллектуальной метасистеме IMS можно также выделить семейство интеллектуальных подсистем, ориентированных на поддержку проектирования различных классов ИС, таких, как:

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

Создание и развитие технологии OSTIS осуществляется в рамках открытого проекта OSTIS. Проект IMS, направленный на создание и развитие метасистемы IMS является ключевым подпроектом проекта OSTIS. Другими его подпроектами являются проекты создания различного вида приложений на основе технологий OSTIS. При этом технические решения, полученные при разработке приложений, должны пополнять интегрированную библиотеку многократно используемых компонентов интеллектуальных систем.

Заключение

К числу основных современных тенденций развития искусственного интеллекта следует отнести:

- 1) переход от частных теорий (моделей) различных компонентов ИС к общей (единой, комплексной, интегрированной) теории ИС;
- 2) переход от теории ИС к практике создания широко используемых интеллектуальных систем, к существенному расширению сфер реального практического использования ИС;
- 3) переход от создания прикладных ИС к созданию технологий, обеспечивающих быструю их разработку (как минимум, быстрое прототипирование) широким контингентом разработчиков и реализованных в виде интеллектуальных метасистем, осуществляющих поддержку такой разработки;
- 4) переход от разработки ИС к постоянному их совершенствованию (в том числе, в ходе эксплуатации) в целях существенного продления их жизненного цикла;
- 5) переход от технологий разработки ИС к технологиям их постоянного совершенствования (эволюции) в ходе эксплуатации;
- 6) переход от технологий разработки и совершенствования ИС к технологиям совершенствования (эволюции) самих этих технологий (в том числе и соответствующих им интеллектуальных метасистем);
- 7) переход от автономных ИС к коллективным ИС и к коллективам, состоящим из интеллектуальных и традиционных компьютерных систем, а также к технологиям, обеспечивающим разработку таких коллективов взаимодействующих систем.

Для обеспечения перечисленных тенденций необходимо создание не только общей теории ИС, но и общей семантической теории любых компьютерных систем, обеспечивающей их унификацию и совместимость и четко разделяющей многообразие форм реализации систем, от многообразия их смысловых (семантических) моделей.

Такую общую теорию можно строить только на основе формализации смысла обрабатываемой информации. Проблема формализации смысла в современной информатике является ключевой, поскольку без её решения невозможно решить целый ряд проблем, таких, как проблема семантической совместимости компьютерных систем, проблема дублирования технических решений при разработке компьютерных систем из-за многообразия форм их реализации.

Для обеспечения семантической совместимости компьютерных систем необходимо четко отличать:

- многообразие языков (форм) представления информации от многообразия смысла представляемой информации;
- многообразие форм организации обработки информации (машин обработки информации) от многообразия сути обработки информации на смысловом уровне.

Список источников

- [1] **Голенков, В.В.** Проект открытой семантической технологии компонентного проектирования интеллектуальных систем. Часть 1: Принципы создания / В.В. Голенков, Н.А. Гулякина // *Онтология проектирования*. – 2014. - №1(11). – С. 42-64.
- [2] **Шункевич, Д.В.** Семантические модели и средства компонентного проектирования машин обработки знаний / Д.В. Шункевич // *Электроника инфо*. – 2013. - №11. – С. 27-30.
- [3] **Корончик, Д.Н.** Реализация хранилища унифицированных семантических сетей / Д.Н. Корончик // *Материалы международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем»*. – Мн.: БГУИР, 2013. – С. 125-128.
- [4] **Ивашенко, В.П.** Интеграция на основе унифицированного представления знаний / В.П. Ивашенко // *Электроника инфо*. – 2013. - №10. – С. 36-46.
- [5] **Голенков, В.В.** Графодинамические ассоциативные модели и средства параллельной обработки информации в системах искусственного интеллекта / В.В. Голенков, Н.А. Гулякина // *Доклады БГУИР*. – 2004. - №1(5). – С. 92-101.
- [6] **Голенков, В.В.** Представление и обработка знаний в графодинамических ассоциативных машинах / В.В. Голенков, О.Е. Елисеева, В.П. Ивашенко и др. – Мн.: БГУИР, 2001. – 412 с.
- [7] **Касьянов, В.Н.** Графы в программировании: обработка, визуализация и применение / В.Н. Касьянов, В.А. Евстигнеев. – СПб.: BHV–Петербург, 2003. – 1104 с.
- [8] **Колб, Д.Г.** Web-ориентированная реализация семантических моделей интеллектуальных систем / Д.Г. Колб // *Материалы международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем»*. – Мн.: БГУИР, 2012. – С. 111-122.
- [9] **Грибова, В.В.** Управление интеллектуальными системами / В.В. Грибова, А.С. Клещев, Е.А. Шалфеева // *Известия РАН. Теория и системы управления*. - 2010. - №6. - С. 122-137.
- [10] **Грибова, В.В.** Облачная платформа для разработки и управления интеллектуальными системами / В.В. Грибова, А.С. Клещев, Д.А. Крылов и др. // *Материалы международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем»* – Мн.: БГУИР, 2012. – С. 130-145.
- [11] Открытая семантическая технология проектирования интеллектуальных систем [Электронный ресурс]. – 2014. - Режим доступа: <http://ostis.net>. – Дата доступа: 02.10.2014.
- [12] **Давыденко, И.Т.** Модель и средства компонентного проектирования баз знаний на основе унифицированных семантических сетей / И.Т. Давыденко // *Электроника инфо*. – 2013.- №11. – с. 23-25.
- [13] **Стефанюк, В.Л.** Сотрудничающий компьютер / В.Л. Стефанюк, А.В. Жожикашвили. – М.: Наука, 2007. - 274 с.
- [14] Интеллектуальная метасистема поддержки проектирования интеллектуальных систем [Электронный ресурс]. – 2014. - Режим доступа: <http://ims.ostis.net>. – Дата доступа: 02.10.2014.

PROJECT OF OPEN SEMANTIC TECHNOLOGY OF THE COMPONENTAL DESIGN OF INTELLIGENT SYSTEMS. PART 2: UNIFIED MODEL DESIGN

Vladimir Golenkov¹, Natalia Guliakina²

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

¹golen@bsuir.by, ²guliakina@bsuir.by

Abstract

The article is a second one in a cycle of articles, dedicated to an open project aimed towards design and development of a component intellectual systems creation technology. The paper addresses unification of semantic data processing models – models of informational search, models of knowledge integration, models of problem solving, models of semantic networks translation into external presentation and vice versa. The model of component design of intellectual systems based on unified semantic models of intellectual systems is described. The means of ensuring the open nature of technology of intelligent systems, methods of their evolutionary design and principles of meta-system designed for complex design support intelligent systems are described.

Key words: *intelligent system, design technology, unified semantic network, semantic network language, component design of the informational systems, graph-based programming language, intellectual metasystem.*

Reference

- [1] **Golenkov, V.V.** Proekt otkrytoj semanticheskoy tehnologii komponentnogo proektirovaniya intellektual'nyh sistem. Chast' 1: Principy sozdaniya [The project is an open semantic technology component of designing intelligent systems. Part 1: Principles of creation] / V.V. Golenkov, N.A. Guliakina // *Ontology of designing*. – 2014. – No.1(11). – pp. 42-64. (In Russian)
- [2] **Dhunkevich, D.V.** Semanticheskie modeli i sredstva komponentnogo proektirovaniya mashin obrabotki znaniy [Semantic models and tools for component design of knowledge processing machines] / D.V. Shunkevich // *Elektronika info*. – 2013. – No.11. – pp. 27-30. (In Russian)
- [3] **Koronchik, D.N.** Realizacija hranilishha unificirovannyh semanticheskikh setej [Implementation of unified repository of semantic networks] / D.N. Koronchik // proceedings of the «Open semantic technologies of intellectual systems design» conference. – Minsk.: BGUIR, 2013. – pp. 125-128. (In Russian)
- [4] **Ivashenko, V.P.** Integracija na osnove unificirovannogo predstavleniya znaniy [Integration on the basis of a unified knowledge representation] / V.P. Ivashenko // *Elektronika info*. – 2013. – No. 10. – pp. 36-46. (In Russian)
- [5] **Golenkov, V.V.** Grafodinamicheskie assotsiativnye modeli i sredstva parallel'noj obrabotki informatsii v sistemakh iskusstvennogo intelekta [Graphdynamic associative models and means of parallel processing of information in artificial intelligence systems] / V.V. Golenkov, N.A. Gulyakina // *BGYIR proceedings*. - 2004. - Issue 1(5). - pp. 92-101. (In Russian).
- [6] **Golenkov, V.V.** Predstavlenie i obrabotka znaniy v grafodinamicheskikh assotsiativnykh mashinakh [Representation and processing of knowledge in graphdynamic systems] / V.V. Golenkov et al. - Minsk. BGYIR, 2001. – 412 p. (In Russian).
- [7] **Kasianov, V.N.** Grafy v programmirovanii: obrabotka, vizualizacija i primenenie [Graphs in programming: processing, visualization and application] / V.N. Kasianov, V.A. Evstigneev. – St. Ptsb.: BHV–Petersbrg, 2003. – 1104 p. (In Russian)
- [8] **Kolb, D.G.** Web-orientirovannaja realizacija semanticheskikh modelej intellektual'nyh sistem [Web-based implementation of semantic models of intelligent systems] / D.G. Kolb // proceedings of the «Open semantic technologies of intellectual systems design» conference. – Minsk.: BGUIR, 2012. – pp. 111-122. (In Russian)
- [9] **Gribova, V.V.** Upravlenie intellektual'nymi sistemami [Management of intellectual systems] / V.V. Gribova, A.S. Kleshev, E.A. Shalfeva // *Proceedings of the Academy of Sciences. Theory and control systems*. - 2010. – No.6. - pp. 122-137. (In Russian)
- [10] **Gribova, V.V.** Oblachnaja platforma dlja razrabotki i upravleniya intellektual'nymi sistemami [Cloud platform for the development and management of intelligent systems] / V.V. Gribova, A.S. Kleshev, D.A. Krilov and others // proceedings of the «Open semantic technologies of intellectual systems design» conference. – Minsk.: BGUIR, 2012. – pp. 130-145. (In Russian)

- [11] Otkrytaja semanticheskaja tehnologija proektirovanija intellektual'nyh sistem [Open semantic technology of designing intelligent systems]. – 2014. - URL: <http://ostis.net>. – Valid on: 02.10.2014. (In Russian)
- [12] **Davidenko, I.T.** Model' i sredstva komponentnogo proektirovanija baz znanij na osnove unificirovannyh semanticheskikh setej [Model and component design tools of knowledge bases based on unified semantic networks] / I.T. Davidenko // Elektronika info. – 2013.- No.11. – pp. 23-25. (In Russian)
- [13] **Stefanik, V.L.** Sotrudnichajushhij komp'juter [Collaborating computer] / V.L. Stefaniuk, A.V. Jojikashvili. – M.: Science, 2007. - 274 p. (In Russian)
- [14] Intellektual'naja metasisistema podderzhki proektirovanija intellektual'nyh sistem [Intellectual metasystem for intelligent systems design support]. – 2014. - URL: <http://ims.ostis.net>. – Valid on: 02.10.2014. (In Russian)

Сведения об авторах



Голенков Владимир Васильевич, 1949 г. рождения. В 1971 г. с отличием окончил физический факультет Белорусского государственного университета, д.т.н. (1996), профессор, заведующий кафедрой интеллектуальных информационных технологий Белорусского государственного университета информатики и радиоэлектроники, инициатор международной научно-практической конференции «Открытые семантические технологии проектирования интеллектуальных систем», член Российской ассоциации искусственного интеллекта, член редколлегий журналов «Программные продукты и системы», «Онтология проектирования», «Электроника ИНФО», «Речевые технологии». В списке научных трудов более 100 работ в области семантических технологий.

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



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

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