

УДК 004.822:004.89  
DOI: 10.15827/0236-235X.131.404-412

Дата подачи статьи: 13.04.20  
2020. Т. 33. № 3. С. 404–412

## **Агентно-ориентированные модели, методика и средства разработки совместимых решателей задач интеллектуальных систем**

*В.В. Голенков*<sup>1</sup>, д.т.н., профессор, *golen@bsuir.by*  
*Д.В. Шункевич*<sup>1</sup>, к.т.н., зав. кафедрой, *shunkevichdv@gmail.com*

<sup>1</sup> *Белорусский государственный университет информатики и радиоэлектроники, кафедра интеллектуальных информационных технологий, г. Минск, 220013, Республика Беларусь*

Статья посвящена созданию агентно-ориентированных моделей, методики и средств разработки совместимых решателей задач интеллектуальных систем, способных решать комплексные задачи. Рассматриваются требования, предъявляемые к таким решателям, модель решателя задач, удовлетворяющего предъявленным требованиям, а также методика и средства разработки и модификации таких решателей.

Главной проблемой, рассматриваемой в работе, является низкая согласованность принципов, лежащих в основе реализации различных моделей решения задач. Как следствие, существенно затруднено одновременное использование различных моделей решения задач в единой системе при выполнении одной и той же комплексной задачи, практически невозможно повторно использовать технические решения, реализованные в какой-либо системе, кроме того, фактически нет комплексных методик и инструментальных средств, способных обеспечить процесс разработки решателей задач на всех этапах.

Предлагается в качестве основы для решения проблемы совместимости решателей задач использовать мультиагентный подход, а процесс решения любой задачи разделять на логически атомарные действия, которые позволят обеспечить совместимость и модифицируемость разрабатываемых решателей. Решатель предлагается рассматривать как иерархическую систему, состоящую из нескольких взаимосвязанных уровней, что позволяет обеспечить возможность независимого проектирования, отладки и верификации компонентов на разных уровнях.

Предложенные модели, методика и средства применены при разработке ряда прототипов интеллектуальных систем обучающего назначения, а также прототипа системы автоматизации рецептурного производства.

**Ключевые слова:** *семантические технологии, ostis-система, решатель задач, многоагентная система, интеллектуальный агент, база знаний.*

Одним из ключевых компонентов каждой интеллектуальной системы (ИС) является *решатель задач* (РЗ), обеспечивающий возможность решать различные задачи, связанные как непосредственно с основной функциональностью системы, так и с обеспечением эффективности работы, а также автоматизации развития самой этой системы. РЗ, обеспечивающий выполнение всех перечисленных функций, будем называть интегрированным.

Расширение областей применения ИС требует от таких систем возможности решения комплексных задач, предполагающего применение целого ряда различных моделей представления знаний и различных моделей решения задач.

Возможность использования различных моделей решения задач в рамках одной системы позволит декомпозировать комплексную за-

дачу на подзадачи, каждая из которых, в свою очередь, может быть решена одним из известных системе способов. Благодаря комбинации различных моделей решения задач число классов решаемых такой системой задач будет значительно больше, чем суммарное число классов задач, решаемых несколькими системами, каждая из которых реализует только одну из интегрируемых моделей решения задач.

Современные ИС, ориентированные на одновременное использование различных видов знаний и различных моделей решения задач, строятся по принципу гибридных компьютерных систем [1, 2]. Данный подход позволяет решать комплексные задачи, однако при построении гибридных систем возникает необходимость согласования различных моделей решения задач, что существенно повышает накладные расходы при построении таких си-

стем. В результате системы, как правило, имеют сложную монолитную архитектуру, внесение каких-либо изменений в которую требует значительных трудозатрат. Для решения этой проблемы необходимо обеспечить совместимость различных РЗ.

В то же время актуальной является задача обучения ИС новым знаниям и навыкам и ее адаптации под изменяющиеся требования. При этом, в отличие от большинства современных подходов к обучению компьютерных систем (машинному обучению) [3], где класс решаемых задач фактически фиксируется (не меняется в процессе обучения) и оптимизируется лишь способ решения задач указанного класса, речь в данном случае идет о расширении числа классов решаемых системой задач, причем в общем случае – о неограниченном расширении. Очевидно, что подобная реконфигурация системы должна осуществляться непосредственно в процессе эксплуатации системы, а не требовать каждый раз полной остановки всей системы или ее отдельных частей.

На основании сказанного можно сформулировать некоторые специфические требования к РЗ ИС, способной решать комплексные задачи.

- РЗ должен быть легко модифицируемым, то есть трудоемкость внесения изменений в уже разработанный РЗ должна быть минимальной.

- Для того чтобы ИС имела возможность анализировать и оптимизировать имеющийся РЗ, интегрировать в его состав новые компоненты (в том числе самостоятельно), оценивать важность тех или иных компонентов и применимость их для решения той или иной задачи, спецификация РЗ должна быть описана языком, понятным системе. Возможность ИС анализировать (верифицировать, корректировать, оптимизировать) собственные компоненты будем называть рефлексивностью.

Таким образом, несмотря на то, что в настоящее время существует большое число моделей решения задач, многие из которых реализованы и успешно используются на практике в различных системах [4–7], остается актуальной проблема низкой согласованности принципов, лежащих в основе реализации различных моделей такого рода.

### Предлагаемый подход

Разработку модели РЗ, удовлетворяющую перечисленным требованиям, а также соответствующих методики и средств их построения

и модификации предлагается осуществлять в рамках технологии OSTIS [8]. В качестве формальной основы для представления знаний в рамках данной технологии используется унифицированная семантическая сеть с теоретико-множественной интерпретацией. Такая модель представления названа SC-кодом, а элементы такой семантической сети – sc-узлами и sc-коннекторами (sc-дугами, sc-ребрами). Модель какой-либо сущности, описанную средствами SC-кода, будем называть семантической моделью указанной сущности, или просто sc-моделью. Компьютерные системы, построенные по технологии OSTIS, названы ostis-системами. Каждая ostis-система состоит из платформы интерпретации sc-моделей и собственно sc-модели этой системы, которая не зависит от платформы ее интерпретации.

В работе задействован ряд решений, разработанных в рамках технологии:

- графовый процедурный язык SCP, тексты программ которого также записываются с использованием SC-кода и который будет использоваться в качестве базового языка программирования в рамках предлагаемого подхода;

- модели структуризации базы знаний и модели представления различных видов знаний, построенные на основе SC-кода, а также методика и средства коллективного проектирования баз знаний, представленные в работе [9], и средства автоматического редактирования и верификации различных видов знаний, представленные в той же работе;

- вариант реализации платформы интерпретации sc-моделей компьютерных систем, рассмотренный в работе [10].

В основе предлагаемого подхода к решению сформулированных ранее проблем лежат перечисленные далее принципы.

- В качестве основы для построения РЗ предлагается использовать многоагентный подход, который позволит обеспечить возможность построения параллельных асинхронных систем, имеющих распределенную архитектуру, повысить модифицируемость и производительность разработанных РЗ.

- Процесс решения любой задачи предлагается декомпозировать на логически атомарные действия, что также позволит обеспечить совместимость и модифицируемость РЗ.

- РЗ предлагается рассматривать как иерархическую систему, состоящую из нескольких взаимосвязанных уровней. Такой подход позволяет обеспечивать возможность

проектирования, отладки и верификации компонентов на разных уровнях независимо от других уровней.

- Для обеспечения рефлексивности проектируемых ИС предлагается записывать всю информацию о РЗ и решаемых им задачах при помощи SC-кода в той же базе знаний, что и собственно предметные знания системы.

- При проектировании РЗ как иерархической системы на каждом из уровней предлагается использовать компонентный подход, что позволит существенно снизить сроки разработки и повысить надежность РЗ за счет использования отлаженных компонентов. Для реализации такого подхода предлагается разработать в рамках метасистемы IMS [11] библиотеку компонентов РЗ различного уровня, а также методику построения и модификации РЗ, учитывающую наличие такой библиотеки.

- Предлагается строить средства автоматизации и информационной поддержки разработчиков РЗ с использованием технологии OSTIS, то есть и с использованием моделей, методики и средств, предлагаемых в данной работе. Такой подход обеспечит высокие темпы развития указанных средств, а также существенно повысит эффективность средств информационной поддержки, позволяя строить указанные средства как часть интеллектуальной Метасистемы IMS.

Многоагентный подход как основа для построения модифицируемых РЗ имеет ряд преимуществ [12]. Существующие подходы к построению многоагентных систем подробно рассмотрены в работах [13–16].

Основные недостатки большинства популярных современных средств построения многоагентных систем в контексте решаемых в данной работе задач рассмотрены в [17].

В рамках данной работы перечисленные недостатки предполагается устранять за счет использования следующих принципов.

- Коммуникацию агентов предлагается осуществлять по принципу доски объявлений [18], однако в отличие от классического подхода в роли сообщений выступают спецификации в общей семантической памяти выполняемых агентами действий. Такой подход позволяет исключить необходимость разработки специализированного языка для обмена сообщениями, обеспечить модифицируемость всей системы за счет обезличенности общения, а также возможность формулировать задачи в

декларативном ключе, то есть не указывать явно способ решения задачи.

- В роли внешней среды для агентов выступает та же семантическая память, в которой формулируются задачи и посредством которой осуществляется взаимодействие агентов. Такой подход обеспечивает унификацию среды для различных систем агентов, что, в свою очередь, обеспечивает их совместимость.

- Спецификация каждого агента описывается средствами SC-кода в той же семантической памяти, что позволяет минимизировать число специализированных средств, необходимых для спецификации как языковых, так и инструментальных агентов.

- Синхронизацию деятельности агентов предполагается осуществлять на уровне выполняемых ими процессов, направленных на решение тех или иных задач в семантической памяти.

- Каждый информационный процесс в любой момент времени имеет ассоциативный доступ к необходимым фрагментам базы знаний, хранящейся в семантической памяти, за исключением фрагментов, заблокированных другими процессами в соответствии с рассмотренным далее механизмом синхронизации. Таким образом, исключается необходимость хранения каждым агентом информации о внешней среде.

Рассмотрим модель РЗ в соответствии с перечисленными принципами.

### Понятие sc-агента

Единственным видом субъектов, выполняющих преобразования в sc-памяти, будем считать sc-агенты – некоторые субъекты, способные выполнять действия в sc-памяти, принадлежащие некоторому определенному классу логически атомарных действий.

Будем считать, что каждое действие, принадлежащее некоторому конкретному классу логически атомарных действий, обладает двумя необходимыми свойствами:

- выполнение действия не зависит от того, является ли оно частью декомпозиции более общего действия; при выполнении данного действия также не должно учитываться то, что оно предшествует каким-либо другим действиям или следует за ними;

- указанное действие должно представлять собой логически целостный акт преобразования, например, в семантической памяти; по сути оно является транзакцией, то есть ре-

зультатом такого преобразования становится новое состояние преобразуемой системы, а выполняемое действие должно быть либо выполнено полностью, либо не выполнено совсем, частичное выполнение не допускается.

В то же время логическая атомарность не запрещает декомпозировать выполняемое действие на более частные, каждое из которых, в свою очередь, также будет принадлежать некоторому классу логически атомарных действий.

Логическая атомарность выполняемых sc-агентом действий предполагает, что каждый sc-агент реагирует на соответствующий ему класс событий, происходящих в sc-памяти, и осуществляет определенное преобразование sc-текста (текста SC-кода), находящегося в семантической окрестности обрабатываемого события. При этом каждый sc-агент в общем случае не имеет информации о том, какие еще sc-агенты в данный момент присутствуют в системе, и осуществляет взаимодействие с другими sc-агентами исключительно посредством формирования некоторых конструкций (как правило, спецификаций действий) в общей sc-памяти. Таким образом, каждый sc-агент в каждый момент времени контролирует только фрагмент базы знаний в контексте решаемой данным агентом задачи, состояние всей остальной базы знаний в общем случае для него непредсказуемо. Для обеспечения работоспособности такой многоагентной системы необходимо, чтобы каждый sc-агент в ее составе специфицировал в sc-памяти все результаты своих действий.

Важно отметить, что конечный пользователь *ostis*-системы с точки зрения обработки знаний также выступает как sc-агент, формирующий в sc-памяти сообщения путем выполнения элементарных действий, предусмотренных пользовательским интерфейсом. Аналогичным образом осуществляется взаимодействие *ostis*-системы с другими системами и окружающей средой вообще. Вся информация поступает в *ostis*-систему и из нее исключительно посредством соответствующих sc-агентов интерфейса.

Перечислим достоинства предлагаемого подхода к организации обработки знаний в sc-памяти:

- поскольку обработка осуществляется агентами, которые обмениваются сообщениями только через общую память, добавление нового агента или исключение (деактивация) одного или нескольких существующих агентов, как правило, не приводит к изменениям в

других агентах, так как агенты не обмениваются сообщениями напрямую;

- инициирование агентов осуществляется децентрализованно и чаще всего независимо друг от друга, таким образом, даже существенное расширение числа агентов в рамках одной системы не приводит к ухудшению ее производительности;

- спецификации агентов и их программы могут быть записаны на том же языке, что и обрабатываемые знания, в результате чего существенно сокращается перечень специализированных средств, предназначенных для проектирования таких агентов и их коллективов, и упрощается разработка системы за счет использования более универсальных компонентов.

Поскольку предполагается, что копии одного и того же sc-агента или функционально эквивалентные sc-агенты могут работать в разных *ostis*-системах, будучи при этом физически разными sc-агентами, целесообразно рассматривать свойства и классификацию не sc-агентов, а классов функционально эквивалентных sc-агентов, которые будем называть абстрактными sc-агентами. Таким образом, под абстрактным sc-агентом понимается некоторый класс функционально эквивалентных sc-агентов, разные экземпляры которого могут быть реализованы по-разному.

Каждый абстрактный sc-агент имеет соответствующую ему спецификацию, в рамках которой указываются ключевые sc-элементы конкретного sc-агента, а также описание условия инициирования данного sc-агента, то есть класса тех ситуаций в sc-памяти, которые иницируют деятельность данного sc-агента. Кроме того, для каждого абстрактного sc-агента уточняется способ его реализации. С точки зрения реализации можно выделить два класса абстрактных sc-агентов:

- неатомарный абстрактный sc-агент; под ним понимается абстрактный sc-агент, декомпозлируемый на коллектив более простых абстрактных sc-агентов, каждый из которых, в свою очередь, может быть как атомарным абстрактным sc-агентом, так и неатомарным абстрактным sc-агентом;

- атомарный абстрактный sc-агент; под ним понимается абстрактный sc-агент, для которого уточняется конкретная, соответствующая данному агенту программа.

Таким образом, в общем случае спецификация атомарного абстрактного sc-агента включает описание набора программ, реализующих

логику работы агента для атомарного абстрактного sc-агента и указание декомпозиции на более простые агенты для неатомарного абстрактного sc-агента.

В качестве базового языка для описания программ, описывающих деятельность sc-агентов в рамках sc-памяти, предлагается язык SCP, разрабатываемый в рамках технологии OSTIS. Язык SCP ориентирован на обработку унифицированных семантических сетей, представленных в SC-коде.

### Семантическая модель РЗ

С учетом рассмотренных ранее понятий будем говорить, что sc-модель интегрированного РЗ представляет собой неатомарный абстрактный sc-агент, являющийся результатом объединения всех абстрактных sc-агентов, входящих в состав какой-либо конкретной ostis-системы, в один. Другими словами, под sc-моделью интегрированного РЗ понимается коллектив всех sc-агентов, входящих в состав заданной ostis-системы, воспринимаемый как единое целое.

Формально семантическая модель интегрированного РЗ задается следующим образом:  $M_{IPS} = \{AG_{NA}, AG_A, AG_R\}$ , где  $AG_{NA}$  – множество неатомарных абстрактных sc-агентов, входящих в состав РЗ;  $AG_A$  – множество атомарных абстрактных sc-агентов, входящих в состав РЗ;  $AG_R$  – множество понятий, специфицирующих абстрактных sc-агентов в составе РЗ, в том числе описывающих декомпозицию неатомарных агентов на атомарные.

Выделены несколько основных уровней детализации РЗ: уровень самого РЗ; уровень неатомарных sc-агентов, входящих в состав РЗ, в том числе более частных РЗ; уровень атомарных sc-агентов; уровень scr-программ или программ, реализованных на уровне платформы интерпретации sc-моделей.

Такая иерархия уровней обеспечивает возможность, во-первых, поэтапного проектирования РЗ с постепенным повышением степени детализации от верхнего уровня к нижнему, а во-вторых, проектирования, отладки и верификации компонентов на разных уровнях независимо от других уровней, что существенно упрощает задачу построения и модификации РЗ за счет снижения накладных расходов. Кроме того, предлагаемый подход к построению модели РЗ позволяет обеспечить ее модифицируемость и возможность согласованного использования различных моделей решения задач в рамках одного РЗ.

### Принципы взаимодействия параллельных процессов в общей семантической памяти

Для синхронизации выполнения процессов в sc-памяти используется механизм блокировок. Отношение «блокировка\*» связывает знаки процессов в sc-памяти со знаками ситуативных структур, которые содержат sc-элементы, заблокированные на время выполнения данного процесса или на какую-то часть этого периода. Каждая такая структура принадлежит какому-либо из типов блокировки.

В текущей версии для синхронизации выполнения процессов в sc-памяти выделяются три типа блокировок – полная блокировка, блокировка на любое изменение и блокировка на удаление.

С точки зрения средств синхронизации можно выделить три класса sc-агентов:

- sc-агенты интерпретации scr-программ, реализуемые на уровне платформы интерпретации sc-моделей, одной из задач которых является обеспечение описываемого механизма синхронизации; принципы синхронизации агентов данного класса более тривиальны, чем в случае программных sc-агентов, и описываются отдельно;

- программные sc-агенты, обеспечивающие основную функциональность системы, то есть ее возможность решать те или иные задачи и работающие в соответствии с рассматриваемым механизмом;

- sc-метаагенты, задачей которых является координация деятельности программных sc-агентов, в частности, решение проблемы взаимоблокировок.

Более подробно механизм блокировок в семантической памяти рассмотрен в работе [17].

Таким образом, каждому sc-агенту могут в общем случае соответствовать несколько параллельно выполняющихся процессов в sc-памяти, взаимодействие которых регламентируется при помощи описанного механизма блокировок.

### Методика построения и модификации РЗ

Все платформенно-независимые компоненты РЗ могут быть представлены при помощи SC-кода, таким образом, построение РЗ ostis-системы сводится к разработке особого рода фрагмента базы знаний такой системы. В связи с этим при построении и модификации РЗ могут использоваться все существующие

средства автоматизации процесса построения и модификации баз знаний по технологии OSTIS, рассмотренные, в частности, в [9].

Для разработки РЗ на основе рассмотренной модели РЗ предлагается методика, предполагающая применение формальной онтологии деятельности разработчиков таких РЗ и ориентированная на применение многократно используемых компонентов РЗ на каждом уровне структурной иерархии разрабатываемого РЗ, что позволяет снизить трудоемкость. Предлагаемая методика включает несколько этапов (рис. 1).

Более подробно некоторые особенности разработки РЗ согласно предложенной методике рассмотрены в [19].

Основным достоинством предлагаемой методики является ее ориентация на онтологию деятельности разработчиков РЗ, что позволит, с одной стороны, автоматизировать эту деятельность, а с другой, представить спецификации этой деятельности в рамках базы знаний интеллектуальной Метасистемы IMS и обеспе-

чить таким образом информационную поддержку разработчиков РЗ.

### Инструментальные средства построения и модификации РЗ

Для предложенной методики были разработаны и программно реализованы средства автоматизации процесса построения и модификации РЗ. Схематично архитектура средств показана на рисунке 2.

Библиотека включает собственно множество компонентов РЗ, средства их спецификации и поиска на основе спецификации.

Множество компонентов РЗ включает компоненты разного уровня – от целых РЗ до отдельных атомарных абстрактных sc-агентов и конкретных программ обработки sc-текстов.

Разработанные средства отладки РЗ дают возможность отладки на уровне и sc-агентов (инициирование агентов, снятие/установка блокировок), и scr-программ (по аналогии с современными средами программирования).

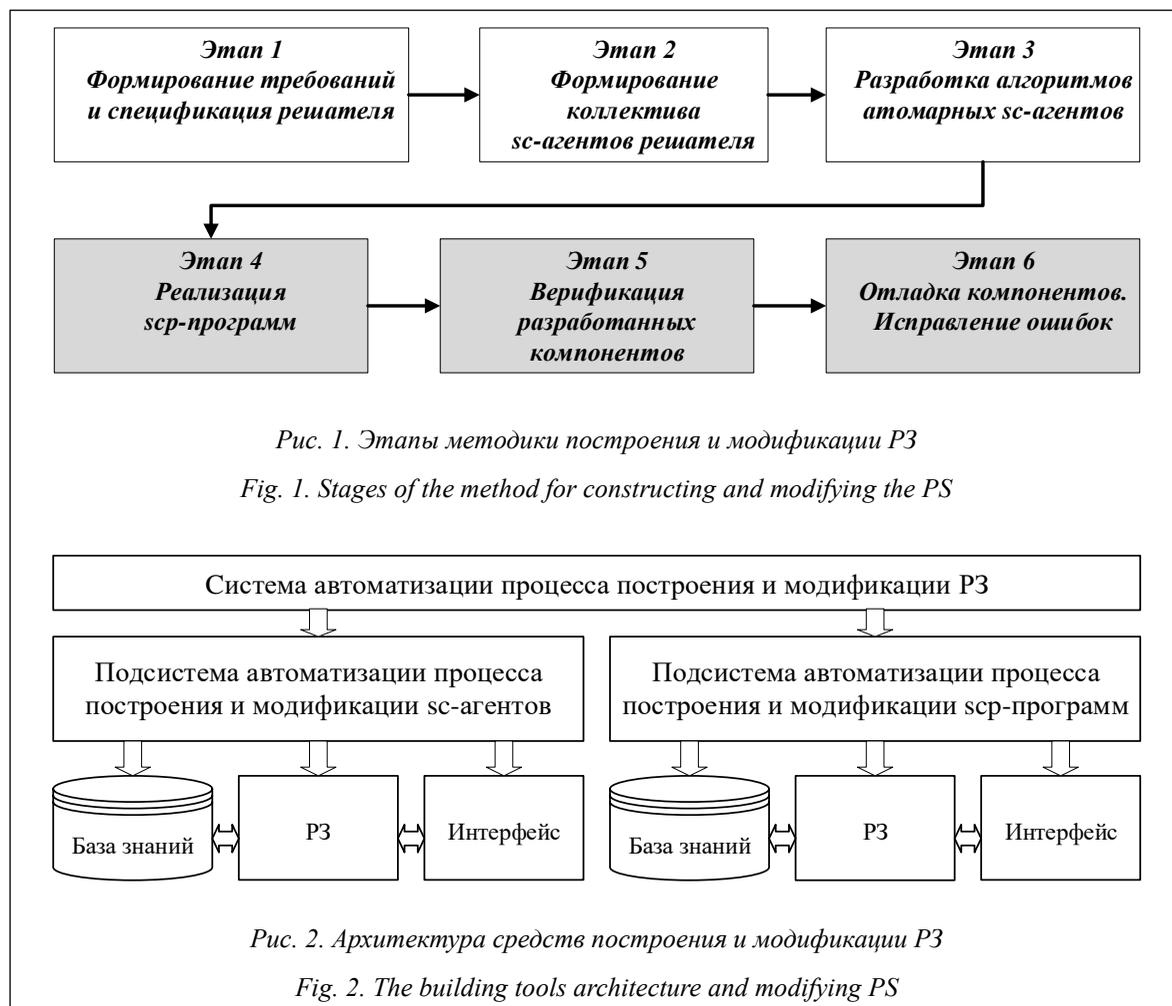


Рис. 1. Этапы методики построения и модификации РЗ

Fig. 1. Stages of the method for constructing and modifying the PS

Рис. 2. Архитектура средств построения и модификации РЗ

Fig. 2. The building tools architecture and modifying PS

Основной особенностью рассмотренных средств построения и модификации РЗ является их реализация на основе технологии OSTIS, в том числе использование для построения средств рассмотренной ранее модели РЗ. Данная особенность позволяет обеспечить модифицируемость самих средств, то есть легкость наращивания их функциональности, в том числе за счет использования компонентов из постоянной пополняемой библиотеки многократно используемых компонентов РЗ.

### Заключение

В работе описана агентно-ориентированная модель гибридного РЗ, рассматривающая каждый такой РЗ как коллектив агентов, работающих над общей семантической памятью, позволяющая реализовать и интегрировать различные модели решения задач в рамках гибридного РЗ, а также обеспечить модифицируемость таких РЗ.

Кроме того, рассмотрена методика построения и модификации гибридных РЗ, основанная на формальной онтологии деятельности их разработчиков и ориентированная на применение библиотеки многократно используемых компонентов РЗ, что позволяет снизить сроки создания РЗ. Также рассмотрены средства автоматизации и информационной поддержки процесса построения и модификации гибридных РЗ, включающие средства автоматизации процесса построения агентов обработки знаний и библиотеку многократно используемых компонентов таких РЗ. Эти средства разработаны с использованием предложенной модели РЗ.

Модели, методика и средства успешно применены при разработке РЗ интеллектуальных справочных систем по различным учебным дисциплинам, таким как геометрия, теория графов, история, химия, а также прототипа системы автоматизации предприятий рецептурного производства [20].

### Литература

1. Колесников А.В., Кириков И.А., Листопад С.В. Гибридные интеллектуальные системы с самоорганизацией: координация, согласованность, спор. М.: Изд-во ИПИ РАН, 2014. 189 с.
2. Castillo O., Melin P., Kasprzyk J. Recent Advances on Hybrid Intelligent Systems. Springer, 2014, 572 p. DOI: 10.1007/978-3-319-05170-3\_1.
3. Николенко С.И., Тулупьев А.Л. Самообучающиеся системы. М., 2009. 288 с.
4. Ефимов Е.И. Решатели интеллектуальных задач. М.: Наука, 1982. 320 с.
5. Подколзин А.С. Компьютерное моделирование логических процессов. Архитектура и языки решателя задач. М.: Физматлит, 2008. 1024 с.
6. Wolfram Alpha. URL: <http://www.wolframalpha.com> (дата обращения: 10.03.2019).
7. Владимиров А.Н., Варламов О.О., Носов А.В., Потапова Т.С. Программный комплекс «УДАВ»: практическая реализация активного обучаемого логического ввода с линейной вычислительной сложностью на основе миварной сети правил // Тр. НИИР. 2010. № 1. С. 108–116.
8. Голенков В.В., Гулякина Н.А. Проект открытой семантической технологии компонентного проектирования интеллектуальных систем. Ч. 1: Принципы создания // Онтология проектирования. 2014. № 1. С. 42–64.
9. Davydenko I. Semantic models, method and tools of knowledge bases coordinated development based on reusable components // Открытые семантические технологии проектирования интеллектуальных систем. 2018. № 8. С. 99–118 (англ.).
10. Корончик Д.Н. Реализация хранилища унифицированных семантических сетей // Открытые семантические технологии проектирования интеллектуальных систем. 2013. № 3. С. 125–128.
11. Метасистема IMS. URL: <http://ims.ostis.net> (дата обращения: 24.05.2019).
12. Wooldridge M. An Introduction to MultiAgent Systems. Wiley, 2009, 484 p.
13. Городецкий В.И., Грушинский М.С., Хабалов А.В. Многоагентные системы (обзор) // Новости искусственного интеллекта. 1998. № 2. С. 64–116.
14. Городецкий В.И., Самойлов В.В., Троцкий Д.В. Базовая онтология коллективного поведения автономных агентов и ее расширения // Изв. РАН: Теория и системы управления. 2015. № 5. С. 102–121.
15. Тарасов В.Б. От многоагентных систем к интеллектуальным организациям. М.: Едиториал УРСС, 2002. 352 с.
16. Florian R.V. Autonomous Artificial Intelligent Agents. 2003, 50 p.
17. Shunkevich D.V. Agent-oriented models, method and tools of compatible problem solvers development for intelligent systems. Proc. OSTIS-2018, Minsk, 2018, pp. 119–132.
18. Jagannathan V., Dodhiawala R., Baum L.S. (Eds.). Blackboard Architectures and Applications. Academic Press, 1989, 560 p.

19. Golenkov V., Shunkevich D., Davydenko I., Grakova N. Principles of organization and automation of the semantic computer systems development. Proc. OSTIS-2019, Minsk, 2019, pp. 53–90.

20. Таберко В.В., Иванюк Д.С., Касьяник В.В., Головкин В.А., Русецкий, К.В., Шункевич Д.В., Гракова Н.В. Принципы построения системы комплексного информационного обслуживания сотрудников предприятия рецептурного производства // Открытые семантические технологии проектирования интеллектуальных систем: сб. тр. Междунар. науч.-технич. конф. Минск, 2019. С. 215–224.

Software & Systems  
DOI: 10.15827/0236-235X.131.404-412

Received 13.04.20  
2020, vol. 33, no. 3, pp. 404–412

### Agent-based models, methods and tools for the development of compatible problem-solvers for intelligent systems

V.V. Golenkov<sup>1</sup>, Dr.Sc. (Engineering), Professor, golen@bsuir.by

D.V. Shunkevich<sup>1</sup>, Ph.D. (Engineering), Head of Department, shunkevichdv@gmail.com

<sup>1</sup> Belarusian State University of Informatics and Radioelectronics,  
Department of Intelligent information technologies, Minsk, 220013, Belarus

**Abstract.** The article is devoted to the development of agent-based models, methods, and tools for the development of compatible problem-solvers for intelligent systems that can solve complex problems. We consider the requirements for such solvers, a model of a problem solver that meets the requirements, as well as the methodology and development tool and modifying such solvers.

The main problem considered in the paper is the problem of low consistency of the principles underlying the implementation of various models for solving problems. As a result, the simultaneous use of different models for solving problems in a single system when solving the same complex problem is substantially difficult, it is practically impossible to reuse technical solutions implemented in any system, in addition, there are practically no complex methods and tools for developing problem solvers capable of providing the solver development process at all stages.

It is proposed to use a multi-agent approach as a basis for solving the problem of compatibility of problem solvers. The process of solving any problem is proposed to be divided into logically atomic actions, which will ensure compatibility and modifiability of the developed solvers. It is proposed to consider the solver as a hierarchical system consisting of several interconnected levels, which provides the ability of the independent design, debugging, and verification of components at different levels.

The proposed models, methods, and tools were applied in the development of a number of prototypes of intelligent educational systems, as well as a prototype of a recipe production automation system.

**Keywords:** semantic technologies, ostis-system, problem solver, multi-agent system, intelligent agent, knowledge base.

### References

1. Kolesnikov A.V., Kirikov I.A., Listopad S.V. *Hybrid Intelligent Systems with Self-Organization: Coordination, Consistency, Dispute*. Moscow, 2014, 189 p. (in Russ.).
2. Castillo O., Melin P., Kacprzyk J. *Recent Advances on Hybrid Intelligent Systems*. Springer, 2014, 572 p. DOI: 10.1007/978-3-319-05170-3\_1.
3. Nikolenko S.I., Tulupev A.L. *Self Learning Systems*. Moscow, 2009, 288 p. (in Russ.).
4. Efimov E.I. *Intellectual Solvers*. Moscow, 1982, 320 p. (in Russ.).
5. Podkolzin A.S. *Computer Simulation of Logical Processes. Solver Architecture and Task Solver Languages*. Moscow, 2008, 1024 p. (in Russ.).
6. *Wolfram Alpha*. Available at: <http://www.wolframalpha.com/> (accessed March 10, 2019).
7. Vladimirov A.N., Varlamov O.O., Nosov A.V., Potapova T.S. UDAV software package: practical implementation of an active learner logical inference with linear computational complexity based on a mivar network of rules. *Tr. NIIR*, 2010, no. 1, pp. 108–116 (in Russ.).

8. Golenkov V.V., Gulyakina N.A. The project of open semantic technology of component design of intelligent systems. Pt. 1: Principles of creation. *Ontology of Designing*, 2014, no. 1, pp. 42–64 (in Russ.).
9. Davydenko I. Semantic models, method and tools of knowledge bases coordinated development based on reusable components. *Open Semantic Technologies for Intelligent Systems*, 2018, no. 8, pp. 99–118.
10. Koronchik D.N. Implementation of storage of unified semantic networks. *Open Semantic Technologies for Intelligent Systems*, 2013, pp. 125–129 (in Russ.).
11. *MetaSystem IMS*. Available at: <http://ims.ostis.net> (accessed May 24, 2019).
12. Wooldridge M. *An Introduction to MultiAgent Systems*. Wiley, 2009, 484 p.
13. Gorodetsky V.I., Grushinsky M.S., Khabalov A.V. Multi-agent systems (Overview). *The Artificial Intelligence News*, 1998, no. 2, pp. 64–116 (in Russ.).
14. Gorodetsky V.I., Samoylov V.V., Trotsky D.V. Basic ontology of collective behavior of autonomous agents and its expansion. *J. of Comp. and Syst. Sci. Int.*, 2015, no. 5, pp. 102–121 (in Russ.).
15. Tarassov V.B. *From Multi-Agent Systems to Intelligent Organizations*. Moscow, 2002, 352 p. (in Russ.).
16. Florian R.V. *Autonomous Artificial Intelligent Agents*. 2003, 50 p.
17. Shunkevich D.V. Agent-oriented models, method and tools of compatible problem solvers development for intelligent systems. *Proc. OSTIS-2018*, Minsk, 2018, pp. 119–132.
18. Jagannathan V., Dodhiawala R., Baum L.S. (Eds.). *Blackboard Architectures and Applications*. Academic Press, 1989, 560 p.
19. Golenkov V., Shunkevich D., Davydenko I., Grakova N. Principles of organization and automation of the semantic computer systems development. *Proc. OSTIS-2019*, Minsk, 2019, pp. 53–90.
20. Taberko V., Ivanyuk D., Kasyanik V., Golovko V., Rusetski K., Shunkevich D., Grakova N. Design principles of integrated information services for batch manufacturing enterprise employees. *Proc. OSTIS-2019*, Minsk, 2019, pp. 215–224 (in Russ.).

#### Для цитирования

Голенков В.В., Шункевич Д.В. Агентно-ориентированные модели, методика и средства разработки совместимых решателей задач интеллектуальных систем // Программные продукты и системы. 2020. Т. 33. № 3. С. 404–412. DOI: 10.15827/0236-235X.131.404-412.

#### For citation

Golenkov V.V., Shunkevich D.V. Agent-based models, methods and tools for the development of compatible problem-solvers for intelligent systems. *Software & Systems*, 2020, vol. 33, no. 3, pp. 404–412 (in Russ.). DOI: 10.15827/0236-235X.131.404-412.