

УДК: 168.2

DOI: 10.15372/PS20260206

EDN: ZWHKGB

В.Ш. Гумиров, А.В. Гумиров, Д.И. Свириденко**ЗАДАЧНЫЙ ПОДХОД И СТРАТЕГИЧЕСКОЕ
СЕМАНТИЧЕСКОЕ МОДЕЛИРОВАНИЕ**

Статья продолжает обсуждение концепции семантического моделирования, разрабатываемой авторами. Рассматриваются особенности, ограничения и характеристики информационных систем, создание которых осуществляется либо с помощью технологий ИИ, либо традиционными способами. Анализируется возможность расширения положений и инструментария концепции семантического моделирования применительно к ситуации создания информационных систем, ориентированных на решение *открытого* класса задач. Предлагается строить стратегии поведения таких ИТ-систем, опираясь на возможность использования опыта предыдущих их действий в похожих обстоятельствах, выбирая те действия, которые ранее приводили к наилучшим результатам, а затем, используя задачный подход, подбирать наиболее подходящие при заданных условиях способы решений задач либо путем имитационного моделирования, либо в режиме принятия решений с целью определения способа, наиболее адекватного их постановке, уделяя особое внимание контекстным условиям и критерию решенности. Ставится и обсуждается цель дальнейшего развития методологии семантического моделирования, ориентированной на создание ИТ-систем, следующих в своем поведении указанной выше стратегии, что и объясняет название статьи. В заключение приводится описание одного из возможных вариантов расширенного языка семантического моделирования, предназначенного для спецификации логики поведения подобных систем.

Ключевые слова: информатика; математическая логика; язык; теория конструктивных моделей; семантика; методология; теория программирования; No-code и Low-code; исполнимые спецификации; задача; критерий решенности задачи; контекст решения задачи; семантическое моделирование; онтология; искусственный интеллект

V.Sh. Gumirov, A.V. Gumirov, D.I. Sviridenko

THE TASK APPROACH AND STRATEGIC SEMANTIC MODELING

The article continues the discussion of the concept of semantic modeling developed by the authors. It examines the features, limitations, and characteristics of information systems created either using AI technologies or traditional methods. The possibility of expanding the provisions and tools of the semantic modeling concept is analyzed with regard to the situation of creating information systems focused on solving an *open* class of problems. It is proposed to develop behavioral strategies for such IT systems based on the possibility to leverage the experience of their previous actions in similar circumstances, selecting those actions that have previously yielded the best results, and then, using the task-based approach, select the most appropriate problem-solving methods under given conditions, either through simulation modeling or decision-making in order to determine the method most appropriate to the problem statement, paying particular attention to contextual conditions and the criterion for solution. The article presents and discusses the goal of further developing a semantic modeling methodology aimed at creating IT systems that behave in the manner described above, which explains the title of the article. In conclusion, it describes a possible variant of an extended semantic modeling language designed to specify the behavioral logic of such systems.

Keywords: computer science; mathematical logic; language; constructive model theory; semantics; methodology; programming theory; No-code and Low-code; executable specifications; problem; problem-solving criterion; problem-solving context; semantic modeling; ontology; artificial intelligence

Введение

Одно из многочисленных расхожих определений интеллекта – определение его как способности решать нестандартные задачи нестандартными способами [7]. Одним из важных уточнений похожих определений понятия «интеллект» является указание на способность к решению нестандартных задач (проблем) без проб и ошибок «в уме» [1]. Таким образом, понятие естественного интеллекта существенным образом отличается от представления о том наборе технологий искусственного интеллекта, которые базируются в первую очередь на машинном обучении, что требует огромного коли-

чества проб и ошибок на основе анализа исторических данных (в последнее время все чаще и синтетических данных, сгенерированных на основе той же истории). Другими словами, система на основе машинного обучения в принципе не способна решить задачу, которая не встречалась ранее и по которой нет данных. Да и классический подход к разработке информационно-технологических систем (ИТ-систем), базируясь на парадигме управления требованиями, основанной на фиксации того набора задач, которые целевая ИТ-система должна будет решать, ориентирован на создание таких ИТ-систем, которые также в принципе не способны решать задачи, не заложенные на этапе проектирования.

Если попытаться сегодня поставить задачу создания ИТ-систем, которые могли бы обладать интеллектом в вышеуказанном смысле, то становится понятно, что никакие современные технологии искусственного интеллекта (ИИ) пока не позволяют создавать системы, обладающие подобным свойством. Одним из ключевых факторов, который препятствует этому, является прежде всего то обстоятельство, что в основе всех основных современных методик разработки программного обеспечения (ПО) лежит, по сути, одно и то же ограничение – фиксация некоторого *ограниченного набора решаемых задач*¹ еще на этапе выработки и анализа требований. Такой фиксированный набор может быть даже весьма большим, для крупных ИТ-систем речь может идти о тысячах решаемых задач, но практически никогда не рассматриваются ситуации, в которых набор решаемых задач неограничен, открыт. Многими исследователями признается, что это в настоящее время концептуально и методологически невозможно. И даже если мы рассмотрим одни из самых сложных современных ИИ-систем – системы на основе больших языковых или мультимодальных моделей (LLM, LMM), то окажется, что круг решаемых ими задач весьма ограничен: генерация текста, изображения или видео по запросу пользователя с учетом заданного им контекста, роль которого играют промпты, генерация текста на языке разметки или языке программирования. Тем не менее хочется уметь создавать такие системы поддержки принятия решений или же системы автоматического принятия решений, которые позволяли бы задавать мо-

¹ Такие задачи часто именуются «use cases», «user stories», «functions», и т.п.

дели принятия решений на высоком уровне или даже метауровне для *открытых/неограниченных наборов задач* и реализовывать эти модели как в режиме имитационного моделирования, так и в режиме принятия решений. Ниже указаны желаемые исходные требования к ИТ-системе, ориентированной на решение открытого/неограниченного класса задач:

- класс решаемых системой задач открыт/неограничен;
- система функционирует на основе и в рамках высокоуровневых целей и ключевых показателей системы;
- принятие решений системой осуществляется на основе оценки контекста, т.е. текущей ситуации с учетом совокупности рисков и возможностей;
- должна существовать возможность задавать модели принятия решений на метауровне, без необходимости сильно вдаваться в специфику конкретного приложения и прикладной области.

Поскольку мы собираемся в дальнейшем придерживаться базовых положений семантического моделирования, то перечисленные выше требования позволяют сформулировать следующую цель: развить методологию семантического моделирования применительно к ситуации разработки интеллектуальных систем поддержки принятия решений (СППР) и систем автоматического принятия решений (САПР) для открытого/неограниченного круга задач.

Данный расширенный вариант семантического моделирования будем называть *стратегическим*. Причина такого названия в следующем.

В бизнесе под *стратегией* понимается не конкретный план действий (т.е. не просто некий алгоритм), а определение долгосрочных целей и задач, а также направлений деятельности, ресурсов и ограничений, в рамках которых организация функционирует и которые необходимы для достижения этих целей. Аналогичное понимание стратегии как инструмента принятия решений продвигается и в работах Роджера Мартина, профессора из Университета Торонто, одного из ведущих специалистов в области стратегического управления: «Стратегия – это интегрированная совокупность стратегических решений, которая обеспечивает уникальное позиционирование ком-

пании в соответствующей отрасли, направленное на создание устойчивого преимущества и уникальной ценности для потребителей, позволяющее превзойти конкурентов» [8, с. 12].

Другими словами, стратегия – это не план, а некий набор мероприятий, которые позволяют в зависимости от конкретной меняющейся *ситуации* вырабатывать решения и изменять планы действий с учетом *изменений*. Заметим, что нас как раз интересует возможность создания и разработки таких информационных систем, которые должны быть способны генерировать свои программы работы в *изменчивых ситуациях* на основе некоей заложенной в них стратегии в вышеуказанном смысле [11].

Если же теперь мы хотим определить, что должно соответствовать понятию «стратегия» в теории конструктивных моделей [2], составляющей математическую основу семантического моделирования, то наилучшим кандидатом, с нашей точки зрения, является понятие *локальной теории*, позволяющей в конкретной ситуации принимать решения, ведущие к достижению долгосрочных целей (кстати, являющихся частью множества предложений данной теории) в рамках тех ресурсных, регламентных и внешних ограничений, которые также являются частью множества аксиом и предложений данной теории.

Итак, каковы же основные типы аксиом, которые должны быть в локальной теории, способной играть роль подходящей стратегии? Ниже указан желаемый состав этих аксиом:

- совокупность логических формул, интерпретируемых как онтологическая модель предметной области, включающая в себя основные понятия, которыми оперирует организация в своей работе, и их взаимосвязи;
- общие знания предметной области в логическом формульном виде;
- формульное представление долгосрочных целей и задач (goals and objectives), поскольку все решения, которые целевая система должна принимать, должны им не противоречить;
- формульное описание внешних и внутренних ограничений, где под внешними ограничениями мы понимаем регуляторные ограничения и регламенты, задаваемые государством, надзор-

ными органами и индустриальными стандартами, а под внутренними – внутренние стандарты и регламенты организации, в том числе относящиеся к морально-этической области.

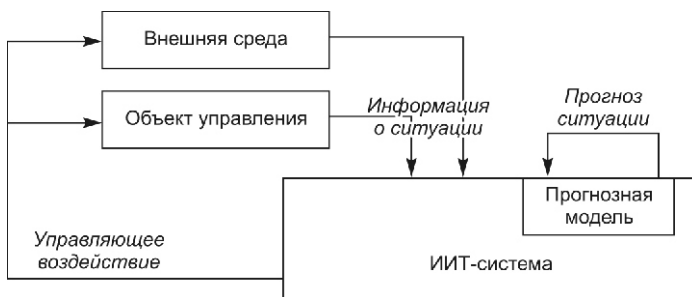
Далее в настоящей статье мы будем обсуждать положения методологии стратегического семантического моделирования, ориентированной на решение проблемы по созданию *интеллектуальных² информационно-технологических систем (ИИТ-систем)*.

1. Исходные требования к ИИТ-системе

В этой статье мы будем обсуждать проблему создания целевой ИИТ-системы, предназначенной для управления неким объектом, который в дальнейшем мы иногда будем называть просто *системой*. Стоит заметить, что объект управления, как правило, представляет собой достаточно сложный предмет – систему, которая в общем случае может состоять из подсистем, которые, в свою очередь, тоже могут состоять из подсистем более низкого уровня, и т.д. Другими словами, речь идет о том, что объект управления может иметь иерархическую структуру, что необходимо учитывать при разработке целевой ИИТ-системы. При этом под управлением далее мы будем понимать выработку решений для заранее открытого/неограниченного круга задач, которые возникают в процессе жизни как объекта управления, так и субъекта управления – управляющей ИИТ-системы.

1.1. Типовая архитектура целевой ИИТ-системы. Прежде всего необходимо определиться с местом и назначением создаваемой ИИТ-системы в реальном мире. Мы говорим о системе, которая предназначена для выработки управляющих решений для некоторого объекта управления (будь то доменная печь, завод, автономный БПЛА или некоторый финтех-сервис). При этом крайне желательно, чтобы ИИТ-система не просто реагировала на изменяющуюся ситуацию во внешней среде и объекте управления, а могла бы *проактивно (упредительно)* управлять на основе *прогноза* изменения ситуации (рис. 1). Понятно, что управляющее воздействие (решение) может

² Интеллектуальных в вышеуказанном смысле, т.е. таких, которые не были ограничены в ассортименте решаемых задач еще на этапе проектирования.

*Рис. 1*

идти не напрямую к объекту управления, а через людей, которые управляют этим объектом (случай СППР).

1.2. Рамочные требования к ИИТ-системам. Сформулируем теперь некие рамочные требования к ИИТ-системам. Мы ожидаем, что создаваемая ИИТ-система должна быть способна:

- 1) вырабатывать решения в режиме, близком к реальному времени;
- 2) представлять свои решения с обоснованием в терминах предметной области;
- 3) вести управление в рамках заданных целевых показателей объекта управления (SLA);
- 4) осуществлять управление, не зависящее от изменяющейся ситуации;
- 5) управлять объектом в проактивном режиме, т.е. уметь использовать прогнозную модель. Заметим, что используемая системой прогнозная модель может изменяться в зависимости от типа решаемых задач, однако для упрощения понимания сути предлагаемой методологии в дальнейшем мы не будем рассматривать проблему модификации прогнозной модели.

1.3. Система аксиом или целевые показатели. Далее мы будем опираться в наших построениях и рассуждениях на работы профессора Д.Е. Пальчунова и его соавторов [10; 15]. Будем считать, что целевая ИИТ-система основывается на четырехуровневой онтологической модели предметной области и что ее первые два уровня (онтология предметной области и общие знания предметной области) имеют статус аксиом (рис. 2). Кроме того, эту систему аксиом мы

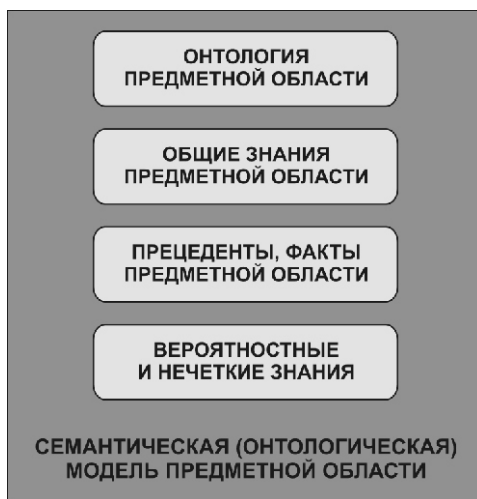


Рис. 2

обогатим также долгосрочными (стратегическими) целями и задачами, к которым добавлены внешние и внутренние ограничения.

1.4. Семантика DSL. Таким образом, в систему аксиом, как было замечено выше, будут входить и определения понятий (язык предметной области), и общие знания предметной области, которые не могут быть нарушены просто по определению. Для краткости будем называть эти компоненты построенной аксиоматики *sDSL* (*semantic Domain Specific Language*).

Из вышесказанного следует, что выстраиваемая система аксиом, по сути, включает в себя набор целевых показателей, которым должен соответствовать объект управления. Однако чтобы быть более точным, следует отметить, что если некоторые из целевых параметров не могут быть нарушены, то другие могут, хотя это нежелательно. Эти тонкости могут быть точно формализованы и заданы в виде соответствующих логических формул, которые также могут входить в систему аксиом. Заметим, что в информатике и ИТ-отрасли такие целевые показатели, в рамках которых целевая система должна функционировать, принято обозначать таким не очень-то говорящим термином, как *SLA* (*Service Level Agreement*). При этом *SLA* обычно

формулируют в терминах, определенных в языке предметной области (sDSL). Внешние и внутренние ограничения могут быть отнесены либо к sDSL, либо к SLA. Выбор подходящего варианта относится скорее к области инженерного искусства, и основным аргументом в пользу того, чтобы регламентные и иные ограничения включать скорее в SLA, нежели в sDSL, является то, что SLA позволяет более гибко задавать эти ограничения не в виде того, что должно быть всегда верно (*must be*), а в виде таких *модальностей*, как *желание* и *намерение* – то, к чему мы стремимся и что хотелось бы, чтобы было выполнено, или должно быть выполнено в определенные моменты времени (например, во время аудита).

1.5. Ситуации и состояния объекта управления. *Текущее состояние* объекта управления в нашей методологии предлагается формально представлять в виде частичной модели (или же как некую логическую формулу без свободных переменных и кванторов³, описывающую текущую ситуацию). Нас же будет интересовать не только текущее состояние объекта управления, но и вся сохраненная *история состояний*. Если быть более точным, то и состояние нас будет интересовать с разбивкой по подсистемам, что приводит нас к некоей иерархии таких состояний, которая соответствует системной иерархии объекта управления.

1.6. Модель управления рисками и возможностями. Важная роль в нашей методологии отводится рискам и возможностям. Набор определений рисков и возможностей нужен для того, чтобы целевая ИИТ-система могла самостоятельно оценивать состояние (историческое, текущее или прогнозируемое) и вырабатывать решения. По сути, речь идет о неких экспертных знаниях, специфичных для данного объекта управления и целевой ИИТ-системы. Эти экспертные знания, если быть строгим с точки зрения четырехуровневой онтологической модели Д.Е. Пальчунова, относятся к третьему уровню (прецеденты и факты предметной области). Мы выделяем именно эту специфическую часть экспертных знаний, поскольку в общем случае все *простые решения*, которые приходится принимать относительно объекта управления, крутятся вокруг именно

³ Такая формула называется атомарной диаграммой.

управления рисками и возможностями. Здесь под *простыми решениями* мы понимаем такие решения, которые, согласно теории адаптации неживых объектов⁴, не требуют адаптации структуры объекта управления, а затрагивают только некоторые параметры его функционирования (т.е. так называемая *стабильная адаптация*). Далее для простоты мы будем рассматривать только такие ситуации, в которых управление требует лишь стабильной адаптации объекта управления в вышеуказанном смысле. Отметим, что задача создания ИИТ-систем для автоматизации задач управления в случаях структурной и катастрофической адаптации объектов управления видится значительно более сложной и потребует, с нашей точки зрения, существенных дополнительных исследований и, возможно, выделения и изучения отдельных подклассов задач и приложений.

Теперь о рисках и возможностях. Далее под *риском* мы будем понимать сущность, удовлетворяющую требованию наличия следующих показателей:

- 1) критерия возникновения, представляющего собой некоторую логическую формулу, принимающую истинное значение тогда и только тогда, когда риск произошел;
- 2) описания влияния (*impact*) в виде формализации класса ситуаций, которые могут возникнуть в случае реализации риска. То есть это то, что может стать следствием реализации риска;
- 3) симптомов или предвестников риска (в форме некоторой дизъюнкции логических условий), при возникновении которых повышается вероятность реализации риска;
- 4) стратегии снижения риска (*risk mitigation strategy*), которая определяет, что нужно делать для того, чтобы понизить вероятность реализации риска;

⁴ В монографии Ш.В. Гумирова «Основы теории адаптации неживых объектов...» определяются основные типы адаптации неживых объектов: стабильная (когда изменяются только параметры функционирования неживого объекта), структурная (когда изменяется внутренняя структура объекта), катастрофическая (когда разрушаются внутренние связи внутри структуры объекта и объект перестает существовать как структура его подсистем) [5].

5) стратегии минимизации последствий риска (risk contingency strategy), которая указывает на то, что нужно делать в случае реализации риска;

6) оценки вероятности возникновения риска в виде некоей опциональной функции, вычисляющей по текущей ситуации и истории ситуаций вероятность того, что риск может быть реализован. Такая оценка может базироваться как на статистических методах, так и на экспертных знаниях;

7) оценки критичности риска, которую можно формализовать как некую аксиому из общей аксиоматики целевой ИИТ-системы. Заметим, что на практике обычно в качестве оценки используют простую шкалу наименований с тремя значениями: *критический риск* – если реализация риска ведет к нарушению SLA; *некритический риск* – если реализация риска не ведет к нарушению SLA, но потребует существенных усилий для минимизации последствий, т.е. риск стоит избегать; *неважный риск* – риск, который желательно избегать, но реализация которого не ведет к существенным последствиям;

8) перечня вариантов действий для минимизации влияния риска в случае его возникновения;

9) перечня вариантов действий для реализации возможности в случае ее возникновения. Заметим, что *возможность*, по сути, мало отличается от *риска*, за исключением того, что *влияние* (impact) имеет позитивное значение с точки зрения достижения целей управления (SLA).

Подытоживая, можно утверждать, что наличие модели управления рисками и возможностями позволяет адекватно оценивать текущую ситуацию и вырабатывать такие варианты дальнейших действий, которые минимизируют вероятность возникновения рисков и повышают вероятность реализации возможностей.

1.7. Оценка ситуации. В целевой системе должна быть определена функция оценки текущей ситуации и истории ситуаций с точки зрения целевых показателей, рисков и возможностей. На выходе эта функция должна выдавать варианты дальнейших действий (другими словами, варианты решений) вместе с их обоснованием и оценками последствий.

2. Требования к компонентам целевой ИИТ-системы

2.1. Подсистема сбора информации о текущей ситуации. Эта подсистема отвечает за то, чтобы максимально точно и оперативно собрать данные об объекте управления, провести предварительную агрегацию этих данных в информационные структуры для последующего анализа, осуществить низкоуровневый анализ информации для последующей оценки ситуации и выработки решения.

В случае, когда объект управления представляет собой сложную иерархическую систему со своими правилами взаимодействия между подсистемами, возможно, стоит реализовывать подсистему сбора информации как мультиагентную систему, где каждый агент соответствует некоторой подсистеме объекта управления. При этом взаимодействие между подсистемами объекта управления формализуется в виде контрактов и представляется в подсистеме сбора информации в виде цифрового регламента. Такой подход позволяет, с одной стороны, контролировать выполнение контрактов между подсистемами, а с другой стороны, принуждать к выполнению этих контрактов для минимизации рисков их нарушения. На выходе эта подсистема выдает изменения в истории ситуаций и изменения текущей ситуации (третий уровень онтологической модели Д.Е. Пальчунова).

2.2. Подсистема мониторинга целевых показателей (SLA). Эта подсистема, базируясь на информации о текущей ситуации, истории и прогнозе, оценивает SLA и их прогноз.

2.3. Подсистема идентификации проблем/возможностей и постановки задач. Данная подсистема выявляет противоречие между желаемым значением SLA и текущим и/или прогнозным. На основе модели управления рисками и возможностями подсистема вырабатывает задачу управления, понимая под ней те локальные цели, которые желательно достичь. Эти локальные цели формализуются в виде некоторой атомарной диаграммы, представляющей собой желаемое состояние объекта управления. Например, если оценка текущей ситуации показывает, что повышается вероятность реализации некоторого риска, то, соответственно, стоит выполнить действия по снижению вероятности (mitigation strategy) наступления этого риска.

Таким образом стратегия снижения данного риска становится одним из конъюнктивных членов локальной цели.

2.4. Подсистема выработки решения. Эта подсистема на основе сформированных локальных целей декомпозирует задачу на подзадачи и путем моделирования (логико-вероятностного или на основе машинного обучения) вырабатывает решения для подзадач. В частности, методы выработки решения могут быть сгенерированы на основе алгоритма Semantic Discovery и обучения с подкреплением (Σgym)⁵ или алгоритмов обучения функциональных систем, разработанных под руководством профессора Е.Е. Витяева и В.В. Мухортова [9].

3. О концепции языка семантического моделирования ИИТ-систем

Предполагается, что читатель знаком с Δ_0 -языком, в терминах которого можно осуществлять декларативную спецификацию различных ИТ-систем в рамках концепции семантического моделирования [3; 4; 12–14]. Однако накопленный опыт использования Δ_0 -языка подсказывает, что при проектировании и последующей декларативной спецификации таких систем хорошо было бы в языке иметь возможность указывать при спецификации некоторые действия, модальности и временные ограничения, которые не только увеличивают выразительную силу языка, позволяя специфицировать, скажем, динамические системы, но и существенно повышают способность создаваемой системы к эффективному поиску решения задач. Однако, с другой стороны, надо понимать, что включение в язык тех же действий, способных динамически менять значения индивидуальных переменных, входящих в спецификации, влечет за собой изменение истинности формульных Δ_0 -спецификаций.

⁵ Σgym – применение Semantic Discovery в режиме обучения с подкреплением для генерации семантических моделей управления рисками для автономных агентов в мультиагентной среде, внутренняя НИОКР (совместно ООО «АиЛайн» и НГУ), выполненная группой студентов под руководством А.В. Гумирова.

Тем не менее, несмотря на эту проблему, возможность перейти от статичной версии языка к его динамичному варианту выглядит крайне привлекательно. Еще более заманчиво выглядит желание пойти дальше и расширить круг специфицируемых в языке ИТ-систем за счет не только динамических систем, но и ИИТ-систем, ориентированных на решение открытого класса задач, что, кстати, означает, что среди поступающих на вход ИИТ-системы могут встречаться задачи самой различной природы. Например, это могут быть задачи обучения или задачи рассуждений. Возникает вопрос: можно ли указать корректный язык семантического моделирования с достаточно простым синтаксисом, денотационной и эквивалентной ей операционной семантиками, в котором присутствовали бы конструкции, представляющие собой и формулы, и действия, а если необходимо, то и нужные средства спецификации и поддержки решения задач обучения и рассуждения, а также модальности и темпоральные операторы. О концепции такого языка (назовем его для краткости *DSM*) и пойдет речь ниже. Отметим, что данная концепция во многом следует идеям монографии Ю.Л. Ершова [6].

Пусть σ^* – расширенная предикатными переменными сигнатура конструктивной модели $HW(\mathcal{M})$ сигнатуры σ и пусть L – это язык исчисления предикатов первого порядка сигнатуры σ^* , ограниченный классом Δ_0 -формул с положительным вхождением в них предикатных переменных из набора $\sigma^* \setminus \sigma$. Таким образом, в языке присутствуют переменные двух типов: *индивидные* ($x, y, z, \dots$) и *предикатные* ($P, Q, H, \dots$). Множество переменных V – это объединение X множества индивидных переменных X и множества предикатных переменных P . Отображение $v: X \cup P \rightarrow HW(M) \cup 2^{HW(M)}$ будем называть *состоянием*. Через $FV(\varphi)$ будем обозначать множество *свободных* индивидных и предикатных переменных формулы φ . Определим индуктивно взаимно определяемые классы *DSM-действий* и *DSM-формул* таким образом, чтобы *DSM-действия* (далее – просто *действия*) были способны изменять состояния, что влечет за собой изменение истинности *DSM-формул* (далее – просто *формулы*). Будем пользоваться символами $\alpha, \beta, \gamma, \dots$ для обозначения

действий и символами $\varphi, \Psi, \dots$ – для обозначения формул. Через $FV(E)$ будем обозначать множество свободных индивидных и предикатных переменных синтаксической конструкции E – действия или формулы. Определим индуктивно синтаксис фрагмента языка DSM.

Синтаксис фрагмента языка DSM

1. Каждая Δ_0 -формула φ языка L сигнатуры σ^* есть *формула DSM* (далее – просто формула).

2. Класс формул замкнут относительно операций $\vee, \wedge, \exists, \forall, \rightarrow, \neg$, $x \in t$, $x \neq t$, $x \leq t$, $x \geq t$, где x – индивидная переменная; t – терм, не содержащий вхождений индивидной переменной x ; $\in$ означает «быть элементом списка», а $\leq$ – «быть начальным подписанием».

3. Пусть $P \in \sigma^* \setminus \sigma$ – некий предикатный символ. Тогда P есть *действие* и $FV(P) = \{P\}$.

4. Если φ -формула и x – индивидная переменная, то выражение $\alpha \triangleq [\lambda x. \varphi]$ есть действие, где $\triangleq$ читается «по определению».

5. Если α – действие, а t – терм сигнатуры σ , то $\varphi \triangleq \alpha(t)$ – формула, у которой $FV(\varphi) = FV(\alpha) \cup FV(t)$.

6. Если α и β – действия, φ – формула, x – переменная, а t – терм, то $[\alpha \wedge \beta]$, $[\alpha \vee \beta]$, $[x \in t. \alpha]$, $[x \neq t. \alpha]$, $[x \leq t. \alpha]$, $[x \geq t. \alpha]$, $[\alpha \rightarrow \varphi]$, $[\varphi \rightarrow \alpha]$, $[\varphi \wedge \alpha]$, $[\varphi \vee \alpha]$ – действия.

7. Если α – действие, x – индивидная переменная, φ – формула, то $\Psi \triangleq \langle x := \alpha \rangle \varphi$ – формула, у которой $FV(\Psi) = FV(\alpha) \cup FV(\varphi) \setminus \{x\}$.

8. Если α и β – действия, x – индивидная переменная, то $\gamma \triangleq \langle x := \alpha \rangle \beta$ – действие, у которого $FV(\gamma) = FV(\alpha) \cup FV(\beta) \setminus \{x\}$.

9. Если α – действие, $P \in \sigma^* \setminus \sigma$ – предикатная переменная, φ – формула, то $\Psi \triangleq \langle P := \alpha \rangle \varphi$ – формула, у которой $FV(\Psi) = FV(\alpha) \cup FV(\varphi) \setminus \{P\}$.

10. Если α и β – действия, P – предикатная переменная, то $\gamma \triangleq \langle P := \alpha \rangle \beta$ – действие, у которого $FV(\gamma) = FV(\alpha) \cup FV(\beta) \setminus \{P\}$.

11. Если α и β – действия, φ – формула, x – переменная, а t – терм, то $[\alpha; \beta]$ – действие (*композиция действий*), $\langle x := t \rangle \varphi$ – формула, $[x := t. \alpha]$ – действие, **[if φ then α else β]** – *if-действие* и **[while φ do α]** – *while-действие*.

12. Если φ – формула, то $[\varphi?]$ – действие (*охранник/фильтр*), выполнение которого заключается в проверке условия φ и в случае его истинности выполнение общего действия продолжается, иначе вся программа закичивается.

13. Обогащаем наш формализм *недетерминированными действиями*, разрешая создание действий вида $[<x := ?> \alpha]$ и $[<P := ? > \alpha]$, где запись $E := ?$ означает «*присвоить E произвольное значение*»; этим значением может быть некая информация из внешнего мира, поставляемая тем или иным *оракулом*.

14. Для решения проблемы планирования обогатим наш язык конструкцией ρ , задаваемой следующей грамматикой: $\rho ::= \mathbf{O}\varphi \mid \varphi \mathbf{U} \mathbf{t} \varphi$. Здесь φ – формула, а $\mathbf{O}$ и $\mathbf{U} \mathbf{t}$ темпоральные операторы, интуитивно понимаемые следующим образом: $\mathbf{O}\varphi$ – «*в следующий момент истинна формула φ* »; $\varphi_1 \mathbf{U} \mathbf{t} \varphi_2$ – «*не более чем через t шагов (когда-нибудь, если $t = \infty$) истинна φ_2 , а до этого истинна формула φ_1* ».

Аналогично работе [6] можно определить математическую семантику фрагмента языка DSM и доказать ряд интересных и полезных математических фактов. Так, например, в рамках построенного языка мы можем анализировать различные взаимоотношения действий, скажем отношения *аппроксимации* и *эквивалентности* между действиями: если α и β – действия, то эти отношения мы будем записывать в виде выражений $(\alpha \dot{\sim} \beta)$ и $(\alpha \sim \beta)$, где $(\alpha \dot{\sim} \beta)$ означает, что $x(\alpha(x) \beta(x))$, а $(\alpha \sim \beta)$ означает, что имеет место $(\alpha \dot{\sim} \beta) \ \& \ (\beta \dot{\sim} \alpha)$. Имеют место равенства: $[\alpha \ \beta](t) = \alpha(t) \ \beta(t)$, $[\alpha \ \beta](t) = \alpha(t) \ \beta(t)$, $[\alpha \ \varphi](t) = [\varphi \ \alpha](t) = \alpha(t) \ \varphi$, $[\alpha \ \varphi](t) = [\varphi \ \alpha](t) = \alpha(t) \ \varphi$, $<x:=\alpha>\beta = y(\alpha(y) \ \varphi(y \ x))$ при условии, что переменная y не входит свободно в α и φ , $<x:=\alpha>\varphi = [y(\alpha(y) \ \beta(y \ x))]$ при условии, что переменная y не входит свободно в α и β .

Пусть $\mathbf{V}$ – это множество всех состояний модели $HW(\mathcal{M})$. Очевидно, что каждое действие α определяет бинарное отношение $\nu_1 \alpha \nu_2$ на множестве $\mathbf{V}$, интуитивно понимаемое как «*действие α , исполняясь и перевычисляя изначально заданные состоянием ν_1 значения*

своих свободных переменных в новые значения, тем самым переводит состояние v_1 в новое состояние v_2 ». Если для состояния v_1 существует состояние v_2 такое, что имеет место $v_1 \alpha v_2$, то будем говорить, что действие α *останавливается* на состоянии v_1 , и это факт обозначать как $v_1 \models \alpha$. Можно ввести понятие *пути* как последовательности $\langle v_1, v_2 \rangle$, $\langle v_2, v_3 \rangle$, ..., $\langle v_{n-1}, v_n \rangle$ и ставить вопрос *планирования* построения такой последовательности программ $\alpha_1, \alpha_2, \dots, \alpha_{n-1}$, что имеет место $v_1 \alpha_1 v_2, v_2 \alpha_2 v_3, \dots, v_{n-1} \alpha_{n-1} v_n$, где v_1 – *исходное*, а v_n – *целевое состояние*. Здесь особо интересной является проблема *декомпозиции* данной задачи путем определения такого набора *интервалов* типа $(v_i: \Psi_i; \alpha_i; v_{i+1}: \Psi_{i+1})$ (с указанием состояний и соответствующих им формул и действий), что весь искомый интервал $(v_{\text{начало}}: \Psi_{\text{начало}}; \alpha; v_{\text{цель}}: \Psi_{\text{цель}})$ предстает как последовательность интервалов: $\langle (v_{\text{начало}}: \Psi_{\text{начало}}; \alpha_{\text{начало}}; v_1: \Psi_1), (v_1: \Psi_1; \alpha_1; v_2: \Psi_2), \dots, (v_n: \Psi_n; \alpha_n; v_{\text{цель}}: \Psi_{\text{цель}}) \rangle$. Нетрудно понять, что в более общем случае мы будем иметь дело не с линейной последовательностью интервалов, а с древовидной структурой, состоящей из интервалов, для которой необходимо будет решать сложную задачу нахождения подходящей ветви этого дерева и при этом желательно находить оптимальную ветку согласно некоему критерию. Заметим также, что ставя проблему планирования, целесообразно определиться с такими понятиями, как *представление* («*считается, что верно ф*»), *желание* («*хочется, чтобы было верно ф*») и *намерение* («*планируется вести себя так, что, согласно представлениям, будет верно ф*»), вводя в язык соответствующие операторы. С этими модальностями мы уже сталкивались ранее, обсуждая требования и архитектуру ИИТ-системы.

При использовании всех этих конструкций представляется целесообразным ввести в язык также и кванторы пути $\mathbf{A}$ и $\mathbf{E}$, которые интерпретируются аналогично кванторам пути логики ветвящегося времени как «*любой возможный путь удовлетворяет свойству р*» и «*существует такой возможный путь, который удовлетворяет свойству р*».

И последнее замечание. Представляется, что при спецификации ИИТ-систем во многих случаях целесообразно предполагать, что время *дискретно* и что ИИТ-система «воспринимает» окружающий мир исключительно только через значения показателей, характеризующих текущее состояние объекта управления. Это позволяет нам считать, что состояние объекта управления в каждый конкретный момент времени t может быть записано в виде вектора значений переменных $V(t) = (v_1, \dots, v_n)$, полностью характеризующего состояние объекта. Именно начиная с этих достаточно жестких предположений можно уже сейчас реально ставить и решать ***задачи обучения и прогнозирования***, о чем шла речь ранее. И здесь, как уже говорилось, можно либо воспользоваться положениями и результатами ***семантического вероятностного вывода***, развиваемого проф. Е.Е. Витяевым, либо проанализировать возможность использования концепции ***конечных автоматов Цетлина***, способных к обучению, либо использовать ***механизм оракульной вычислимости для интеграции ИИТ-системы с LLM и LMM*** как с оракулами, либо применить ***гибридный вариант***. В дальнейшем, после соответствующих научных исследований, естественно, многие указанные выше ограничения могут быть ослаблены, сняты или изменены и можно будет перейти к построению полноценного языка семантического моделирования ИИТ-систем.

Заключение

Проблема создания ИИТ-систем, ориентированных на решение открытого класса задач, является одной из центральных тем, обсуждаемых сейчас на различных конференциях, семинарах, форумах, в чатах и журналах. Особенно жаркие споры по этой тематике ведутся среди исследователей, интересующихся такими направлениями, как общий искусственный интеллект (AGI) и агентный ИИ, в частности физический ИИ (Physical AI) и его разновидность – воплощенный искусственный интеллект (Embodied AI), предназначенный для рациональных киберфизических агентов, действующих в реальной, физической среде. Анализ этих обсуждений показывает, что к на-

стоящему времени удовлетворительное решение проблемы создания ИИ-агентов, способных решать открытый класс задач, пока не предложено. Но это не означает, что данная проблема принципиально не может быть решена. Мнение авторов этой статьи заключается в том, что к ее решению нужно двигаться постепенно, шаг за шагом. И один из таких шагов представлен в настоящей статье.

Литература

1. *Большая психологическая энциклопедия*. URL: <https://psychology.academic.ru/814/интеллект>.
2. *Гончаров С.С., Еришов Ю.Л.* Конструктивные модели. Новосибирск: Научная книга, 1999.
3. *Гончаров С.С., Свириденко Д.И.* Логический язык описания полиномиальной вычислимости // Доклады РАН. 2018. Т. 485, № 1. С. 11–14.
4. *Гончаров С.С., Свириденко Д.И.* Рекурсивные термы в семантическом программировании // Сибирский математический журнал. 2018. Т. 59, № 6. С. 1279–1290.
5. *Гумиров Ш.В.* Основы теории адаптации неживых объектов и адаптивный анализ в геологии. Новокузнецк: Интеллект, 1993.
6. *Еришов Ю.Л.* Определимость и вычислимость. Новосибирск: Научная книга, 2000.
7. *Краткий толковый психолого-психиатрический словарь*. URL: <https://med.niv.ru/doc/dictionary/psycho-psychiatric/fc/slovar-200.htm#zag-244>.
8. *Мартин Р.* Игра на победу: Как стратегия работает на самом деле. М.: ООО «Манн, Иванов и Фербер», 2014.
9. *Мухомтов В.В., Хлебников С.В., Витяев Е.Е.* Улучшенный алгоритм семантического вероятностного вывода в задаче 2-мерного анимата // Нейроинформатика. 2012. Т.6, № 1. С. 50–62.
10. *Пальчунов Д.Е., Яхьяева Г.Е.* Нечеткие алгебраические системы // Вестник НГУ. Сер.: Математика, механика, информатика. 2010. Т.10, № 3. С. 75–92.
11. *Chandler A.D., Jr.* Strategy and Structure: Chapters in the History of the Industrial Enterprise. Cambridge: MIT Press, 1962.
12. *Goncharov S.S.* Conditional terms in semantic programming // Siberian Mathematical Journal. 2017. Т. 58. № 5. С. 794–800.
13. *Goncharov S., Nechesov A.* Solution of the problem $P=L$ // Mathematics. 2022. Vol. 10, No. 1. URL: <https://www.mdpi.com/2227-7390/10/1/113>.
14. *Goncharov S.S., Sviridenko D.I.* Σ -programming // Transactions of the American Mathematical Society. 1989. Vol. 142. P. 101–121.

15. *Gumirov V., Matyukov P., Palchunov D.* Semantic domain-specific languages // SIBIRCON 2019 – International Multi-Conference on Engineering, Computer and Information Sciences, Proceedings Compilation, 2019. P. 0955–0960.

References

1. *The Great Psychological Encyclopedia.* Available at: <https://psychology.academic.ru/814>. (In Russ.).
2. *Goncharov, S.S. & Yu.L. Ershov.* (1999). Constructive Models. Novosibirsk, Nauchnaya Kniga Publ. (In Russ.).
3. *Goncharov, S.S. & D.I. Sviridenko.* (2018). Logical language of description of polynomial computing. Doklady RAN [Reports of the RAS], Vol. 485, No. 1, 11–14. (In Russ.).
4. *Goncharov, S.S. & D.I. Sviridenko.* (2018). Recursive terms in semantic programming. Sibirskiy matematicheskiy zurnal [Siberian Mathematical Journal], Vol. 59, No. 6, 1279–1290. (In Russ.).
5. *Gumirov, Sh.V.* (1993). Fundamentals of the Theory of Adaptation of Inanimate Objects and Adaptive Analysis in Geology. Novokuznetsk, Intellect Publ. (In Russ.).
6. *Ershov, Yu.L.* (2000). Definability and Computability. Novosibirsk, Nauchnaya Kniga Publ. (In Russ.).
7. *Brief Explanatory Dictionary of Psychology and Psychiatry.* Available at: <https://med.niv.ru/doc/dictionary/psycho-psychiatric/fc/slovar-200.htm#zag-244>. (In Russ.).
8. *Martin, R.* (2014). Playing to Win: How Strategy Really Works. Moscow, LLC “Mann, Ivanov and Ferber”. (In Russ.).
9. *Mukhortov, V.V., S.V. Khlebnikov & E.E. Vityaev.* (2012). An improved algorithm for semantic probabilistic inference in the 2D animat problem. Neyroinformatika [Neuroinformatics], Vol. 6, No. 1, 50–62. (In Russ.).
10. *Palchunov, D.E. & G.E. Yakhyayeva.* (2010). Fuzzy algebraic systems. Vestnik NGU. Ser.: Matematika, mehanika, informatika [NSU Bulletin. Series: Mathematics, Mechanics, Computer Science], Vol. 10, No. 3, 75–92. (In Russ.).
11. *Chandler, A.D., Jr.* (1962). Strategy and Structure: Chapters in the History of the Industrial Enterprise. Cambridge, MIT Press.
12. *Goncharov, S.S.* (2017). Conditional terms in semantic programming. Siberian Mathematical Journal, Vol. 58, No. 5, 794–800.
13. *Goncharov, S. & A. Nechesov.* (2022). Solution of the problem $P=L$. Mathematics, Vol. 10, No. 1. Available at: <https://www.mdpi.com/2227-7390/10/1/113>.
14. *Goncharov, S.S. & D.I. Sviridenko.* (1989). Σ -programming. Transactions of the American Mathematical Society, 142, 101–121.
15. *Gumirov, V., P. Matyukov & D. Palchunov.* (2019). Semantic domain-specific languages. In: SIBIRCON 2019 – International Multi-Conference on Engineering, Computer and Information Sciences, Proceedings Compilation, 0955–0960.

Информация об авторах

Гумиров Виталий Шамилович – ООО «Eyeline СНГ» (630090, Новосибирск, ул. Мусы Джалиля, 3/1); независимый исследователь.

Гумиров Андрей Витальевич – Новосибирский национальный исследовательский государственный университет (630090, Новосибирск, ул. Пирогова, 1).

Свириденко Дмитрий Иванович – Институт философии и права СО РАН (630090, Новосибирск, ул. Николаева, 8); Центр искусственного интеллекта Новосибирского национального исследовательского государственного университета (630090, Новосибирск, ул. Пирогова, 1).

dsviridenko47@gmail.com

Information about the authors

Gumirov, Vitaly Shamilovich – LLC Eyeline CIS (3/1, Musa Jalil St., Novosibirsk, 630090, Russia); independent researcher.

Gumirov, Andrey Vitalyevich – Novosibirsk National Research State University (1, Pirogov St., Novosibirsk, 630090, Russia).

Sviridenko, Dmitry Ivanovich – Institute of Philosophy and Law, Siberian Branch of the Russian Academy of Sciences (8, Nikolaev St., Novosibirsk, 630090, Russia); Center for Artificial Intelligence, Novosibirsk National Research State University (1, Pirogov St., Novosibirsk, 630090).

dsviridenko47@gmail.com

Дата поступления 07.04.2026.
Принята к публикации 12.05.2026.