

В.И. ГОРОДЕЦКИЙ, П.О. СКОБЕЛЕВ  
**МНОГОАГЕНТНЫЕ ТЕХНОЛОГИИ ДЛЯ  
ИНДУСТРИАЛЬНЫХ ПРИЛОЖЕНИЙ: РЕАЛЬНОСТЬ И  
ПЕРСПЕКТИВА**

*Городецкий В.И., Скобелев П.О. Многоагентные технологии для промышленных приложений: реальность и перспектива.*

**Аннотация.** Уже в течение более чем четверти века технология многоагентных систем рассматривается как одна из наиболее перспективных технологий концептуализации и программной реализации сложных распределенных систем. Однако на практике происходит совсем иное: индустрия почти не использует эту технологию, и это несмотря на то, что появляются все новые и новые классы приложений, для которых эта технология представляется чуть ли ни единственно возможной технологией разработки. В статье анализируются недавние прогнозы и реальные достижения в части практического применения многоагентных систем на промышленном уровне. Выявляются проблемы, которые в настоящее время препятствуют широкому промышленному внедрению многоагентных систем и технологий, а также пути их преодоления. Анализируются классы приложений, в реализации которых многоагентные технологии имеют неоспоримые преимущества и оцениваются перспективы развития этих технологий до уровня промышленного применения.

**Ключевые слова:** многоагентные системы, промышленные приложения, дорожная карта MAS, взаимодействие агентов, протокол взаимодействия, интернет вещей.

**1. Введение.** Сообщество специалистов в области интеллектуальных информационных технологий (ИТ) уже более десяти последних лет относит многоагентные системы (MAS) и технологии (MAS-технологии) к числу наиболее перспективных средств разработки сложных систем. Но почему-то ИТ-индустрия не спешит их использовать, несмотря на то, что эта технология продолжает развиваться учеными и специалистами многих ведущих университетов мира. С другой стороны, появляются новые задачи и новые классы приложений, которые как будто специально созданы для применения MAS. Прежде всего это относится к широкому классу мобильных приложений, к приложениям в области облачных сервисов и Интернета вещей, коллективной робототехники, человеко-машинных робототехнических систем и другим актуальным приложениям.

Мировое сообщество ученых и специалистов в области MAS предлагает новые модели для современных приложений, активно развивает игровые модели формализации экономических моделей, разрабатывает новые типы аукционов для соперничающих коалиций агентов и роботов, совершенствует логические языки для обеспечения требуемой выразительности BDI-моделей автономных агентов и MAS в различных инновационных приложениях, предлагает адаптивные

механизмы обучения агентов, теоретические методы верификации распределенных МАС, изучает и развивает новые принципы функционирования агентских сообществ и их кооперации и тому подобное. Тем не менее индустрия практически не реагирует на новые модели, методы и архитектуры МАС и на предлагаемые технологии. Естественно, встает вопрос о том, почему это происходит и когда можно ожидать использования многоагентных технологий на индустриальном уровне. Или этого не случится вообще?

Вместе с тем, возможно, что время МАС наступит уже в самом ближайшем будущем, что связано с появлением в последнее время нового понятия «Цифровых экосистем», построенных на конкуренции и кооперации сервисов [1].

Интересно отметить, что это определение существенно сужается Gartner, который говорит лишь об интеграции в «системах систем» [2].

В данной работе проводится сравнительный анализ ожиданий в части индустриального применения МАС-технологий и реального состояния их использования (раздел 2), анализируются причины и проблемы, которые существенно тормозят внедрение технологий МАС в практику (раздел 3), а также предлагаются пути преодоления этих проблем (раздел 4). Основной вывод данной работы состоит в том, что в настоящее время для выхода МАС-технологий на рынок индустриальных применений необходимо пересмотреть некоторые базовые парадигмы в области формализации моделей агентов и МАС, а также технологические аспекты их проектирования и программной реализации. Приводятся примеры успешного использования МАС-технологий для создания интеллектуальных приложений индустриального уровня, однако большинство из них, как показывает анализ, не вполне следуют традиционным моделям и технологии МАС (раздел 5), тем самым подтверждая тезис о необходимости пересмотра базовых технологических аспектов парадигмы МАС.

Представляется, что начавший развиваться в последнее время Интернет агентов, людей и вещей станет новой парадигмой не только для Industry 4.0, направленной на промышленную автоматизацию и интеллектуализацию производств различного типа, но и для Industry 5.0, которая будет в большей степени обращена на развитие человека и поддержку процессов самоорганизации в обществе [3].

## **2. Состояние прикладных разработок в области МАС.**

Концепция МАС была предложена в середине 1980-х годов. Она сразу была высоко оценена как научным, так и индустриальным сообществами. Уже за первые два десятилетия были построены базовые теоретические основы МАС, начались активные разработки в области технологии и инструментальных средств ее поддержки. К

началу 2000-х годов было предложено несколько хорошо продуманных методологий разработки МАС, началась разработка инструментальных программных средств их поддержки. В 1996 г. была создана общественная организация FIPA (от англ. *Foundation for Intelligent Physical Agents*), главной задачей которой было научное обоснование стандартов в области агентов и МАС, а уже в 2005 г. она стала одним из комитетов IEEE по стандартизации. В это время ожидалось, что МАС и соответствующая технология готовы занять место лидирующей принципиально новой парадигмы проектирования и технологии разработки современных распределенных интеллектуальных систем индустриального уровня практически любой сложности, причем для самого широкого спектра приложений.

Тогда казалось, что оснований для такой точки зрения вполне достаточно. Действительно, эта концепция выглядела очень привлекательной и естественной для понимания и применения, поскольку она предлагает строить системы и решать задачи в том же стиле, в каком они решаются в живой природе и человеческом сообществе, в частности путем взаимодействий, лежащих в основе самоорганизации. Основной принцип создания концептуальной модели МАС-приложений использует представление сложной задачи множеством относительно простых замкнутых подзадач. Их решение поручается программными агентам, которые разрабатываются и программируются практически автономно, работают асинхронно и параллельно, и взаимодействуют с помощью простой техники обмена сообщениями на языке, близком к естественному, то есть аналогично тому, как это делается при решении задач в сообществе живых существ, в частности в человеческом сообществе. Это взаимодействие, реализуемое с помощью диалогов и протоколов, может быть достаточно сложным и разнообразным. При этом агенты, используя обмен сообщениями, могут генерировать события и посылать их другим агентам, вырабатывать и согласовывать варианты решений, пересылать данные, оценивать результаты решения своих подзадач, формировать задания для других агентов, поддерживать синхронизацию коллективных действий, передавать сигналы обратной связи и тому подобное. Эта концепция представляется естественной для приложений, в которых участвует много разных участников с собственными интересами или любых других относительно автономных сущностей. Например, это относится к задачам транспортной логистики, где объектами планирования являются отдельные заказы и грузы, а исполнителями плана являются транспортные средства, водители, станции ремонта, заправки и так далее. То же самое относится и к производственной логистике, аналогичными объектами которой являются заказы, отдельные технологические производственные операции, рабочие и станки,

выполняющие эти операции, другие ресурсы и материалы и так далее. Уже в первые годы существования парадигмы автономных агентов и МАС ее идеи были особенно привлекательными для задач индивидуальной и коллективной робототехники, причем настолько привлекательными, что термины *агент* и *робот* обычно просто не различались [4], что сохраняется и по настоящее время. В то же время классы потенциальных приложений для МАС продолжали расширяться. Это прежде всего относится к приложениям, управляющим сложными крупноразмерными объектами сетевой структуры, а приложений подобного рода на практике становится с каждым днем все больше не только на транспорте и в производстве, но и в энергетике, здравоохранении, военном деле и так далее. Наступающая эра Интернета вещей, в приложениях которой центральным аспектом является именно взаимодействие распределенных автономных объектов физического, виртуального и социального миров, порождает новый широкий класс приложений, для которых МАС-технология представляется идеальной. В 2016 году был проведен первый международный семинар «Интернет агентов», на котором основной темой была технология агентов для Интернета вещей и людей [5].

Одной из самых привлекательных сторон МАС-парадигмы является ее способность естественно и эффективно решать самую трудную задачу разработки сложных программ, а именно программирование взаимодействий множества компонент программы. В концепции и технологии МАС, по существу, эта задача отделяется от программирования агентов и реализуется с помощью диалогов и протоколов взаимодействия. Важно отметить, что в своей базовой формулировке парадигма МАС особо акцентирует внимание на этом факте: она формулируется как парадигма *вычислений на основе взаимодействий* (англ. *computation as interactions*) [6]. Концепция обмена сообщениями с использованием диалогов и протоколов для реализации взаимодействий оказалась очень привлекательной и продуктивной на практике. Не случайно она в последующем получила широкое распространение и в других архитектурах и технологиях разработки сложных интеллектуальных систем.

Показателем уровня зрелости разработок в области теории и практики МАС был проект *Agentlink III* Европейской комиссии FP-6 (2004-2005), основным результатом которого стала дорожная карта *Road Map: «Agent Technology: Computing as Interaction»* [6]. Этот документ фактически подвел итоги двадцатилетнего периода развития парадигмы, модели и технологии МАС, дал оценку практических перспектив МАС и, что самое важное, дал предельно оптимистический прогноз перспектив промышленных применений МАС до 2015+ г.

Однако уже в начале 2000-х годов в развитии теории и технологии МАС что-то пошло не так, как ожидалось. И публичное восприятие работ в области МАС стало менее значимым [7]. Удачные разработки этого и более позднего времени в области интеллектуальных приложений, которые были выполнены в этот период ведущими ИТ-компаниями мира, в частности Apple, Facebook, Google, SAP, совсем не были связаны с МАС или с МАС-технологиями, по крайней мере в общественном восприятии. Например, в приглашенном докладе вице-президента и директора автономной лаборатории компании Моторола Дж. Стресснера (John Strassner) на конференции AAMAS 2007 [8] одной из ключевых тем был анализ состояния индустриальных разработок в области МАС-приложений. Автор этого доклада обнаружил не более шести МАС-приложений, разработанных за 20-летнюю историю МАС, которые хоть и приближенно, но могли бы квалифицироваться как индустриальные.

Скотт Делоч (Scott A. DeLoach), профессор Канзасского университета, один из ведущих ученых в области МАС-технологий, под руководством которого разработаны методология *O-MaSE* и инструментальное средство *agentTool*, в работе [9] явно обозначил отсутствие прогресса в широком промышленном применении МАС. По его мнению, несмотря на более чем 20-летние усилия по разработке агентских технологий, сама эта область все еще находится на ранней стадии развития и не достигла достаточной зрелости, например зрелости объектно-ориентированного подхода (ООП). Он отметил также ряд существенных недоработок в области теории и технологии МАС, которые, как он думает, требуют уточнения. Они обсуждаются далее в разделе 3 данной работы. По его мнению, МАС-концепция и МАС-технология должны в первую очередь продемонстрировать свою способность создавать сложные адаптивные и самоорганизующиеся распределенные системы промышленного уровня, но этого пока не случилось и до настоящего времени.

В 2013 году была опубликована заметная работа [7], в которой авторы тщательно проанализировали реальное состояние прикладных разработок в области МАС на тот момент. Общая идея этой весьма своевременной и даже несколько запоздавшей работы сформулирована авторами уже в ее первых строках: *«В то время, как имеются убедительные свидетельства важности МАС и технологий как исследовательской области, остается неясным, какой практический эффект от нее имеется к настоящему времени»*.

Эта работа интересна в нескольких аспектах. Во-первых, она дает детальный обзор прикладных МАС, которые были разработаны к 2012-2013 гг., что позволяет оценить качество прогноза, данного на

этот же срок в дорожной карте [6]. Во-вторых, в ней имеется интересная статистика, которая позволяет более глубоко проанализировать реальное состояние MAC-разработок различного уровня зрелости на 2013 год, то есть за более чем тридцатилетнюю историю их развития, а также выявить некоторые причины неудовлетворительного состояния соответствующих разработок.

Предваряя анализ прикладных разработок в области MAC и технологий, авторы дорожной карты MAC [6] пишут, что за 20 лет исследований и разработок в этой области к 2005 г. технология MAC еще не достигла уровня зрелости, необходимого для их использования в промышленных разработках. В качестве аргумента для оправдания этого факта приводится сравнение возраста MAC-технологий и ООП, замечая при этом, что ООП приобрело практическую значимость за 30 с лишним лет. Например, ООП язык C++ был создан через 32 года, а *JAVA* — через 39 лет после первых работ по ООП. К настоящему времени возраст MAC-технологий приближается уже к 40 годам, но ничего принципиально нового в области MAC-технологий пока не произошло.

Авторы [6] также сетуют на слабость методологий для проектирования MAC, разработанных к тому времени, хотя с этим трудно согласиться. Действительно, к этому времени было создано и уже достаточно длительно тестировалось несколько глубоко проработанных методологий MAC, например *Gaia* [10], *Tropos* [11], *MaSE* [12], *ADELFE* [13], *MESSAGE* [14], *Prometheus* [15] и ряд других. Другое дело, что на то время они еще не были поддержаны адекватными инструментальными средствами. Важно заметить, что активность разработок в области методологий проектирования и инструментальных средств для поддержки программной реализации агентов и MAC была достаточно высокой и после этого (практически до 2010 года), когда было выполнено более десятка новых разработок. Однако большинство из них к настоящему времени постигла одинаковая участь: их разработки и тестирование прекращены, и только небольшое число методологий и инструментов остается предметом второстепенных разработок.

В период после 2005 г. активность исследований и разработок в области MAC-технологий и средств их инструментальной поддержки не снижалась, но тем не менее прогноз развития промышленных MAC-приложений, данный в дорожной карте [6], оказался слишком оптимистичным [7, 16]. Работа [7] анализирует 152 приложения, сведения о которых автору удалось получить либо от авторов разработок, либо из научной литературы. Но важно заметить, что, как позже выяснилось, более половины проанализированных разработок в последующем, к

сожалению, не были подтверждены авторами, так что уровень и само существование этих разработок остается под вопросом [7].

Наиболее заметные и удачные разработки МАС-приложений разного уровня зрелости при различном соотношении автоматической генерации кода и ручного программирования были выполнены в области управления производством, транспортной логистики, в области аэрокосмических приложений и в энергетике. Однако прогноз темпов развития в этих классах приложений, данный в дорожной карте 2005 г. на 2015 г., не оправдался. Но, наверное, самым неожиданным оказалось то, что темпы появления новых разработок МАС-приложений стали постепенно замедляться, и этот факт отражает реальное падение интереса промышленных компаний к МАС-технологии. Практика показала, что многие приложения, для которых МАС-технология рассматривалась как наиболее перспективная технология [6], к 2013 г. были успешно реализованы с помощью других технологий. Среди них наиболее конкурентоспособными оказались сервис-ориентированные технологии, ГРИД-вычисления, автономные, повсеместные, облачные, вычисления и другие, которые появились значительно позже, однако смогли быстро потеснить МАС-технологии.

Работа [7] была одной из первых, в которой явно прозвучали тревожные ноты относительно перспектив МАС-технологий в конкуренции с другими современными интеллектуальными ИТ, которые стали следствием анализа 152 разработок, известных автору. Эти разработки анализировались с различных точек зрения, однако для целей данной работы существенными является оценка их уровня зрелости. С этой точки зрения приложения в [7] разделены на 3 группы:

– *промышленные* системы или близкие к ним; их оказалось 46 из 152;

– *исследовательские программные прототипы промышленного уровня*, которые тестировались на реальных данных, однако не были использованы в реальной работе; таких было 55;

– *лабораторные исследовательские прототипы (пилотные проекты)*, которые использовались в учебных, научных и других подобных целях; их оказалось 46.

Про остальные пять разработок достоверная информация у авторов [7] отсутствовала. Таким образом, менее трети проанализированных приложений оказались достаточно зрелыми, а их общее число в пять раз меньше прогноза дорожной карты [6]. Но только примерно половина их была заказана промышленными компаниями, а остальные были разработаны либо смешанными коллективами, либо университетскими, то есть были выполнены

коллективами, субъективно заинтересованными в исследовательской работе, но не в индустриальном использовании.

Важным показателем тенденций исследований и разработок этого периода в области МАС-технологий является активное использование методологий и инструментальных средств разработки — в 72% случаев авторы разработок использовали ту или иную методологию и/или программный инструментарий, как правило, ту, которая ими же и разрабатывалась. Это косвенно отражает тот факт, что МАС-приложения в этот период разрабатывались главным образом для тестирования методологий и инструментальных программных систем, но не для индустриальных приложений.

Интересно отметить, что прогноз по типам приложений, которые по мнению авторов документа [3], к 2015 г. будут представлять наибольший интерес, в целом оправдался. Как и предсказывалось, лидером в области использования МАС-приложений индустриального уровня были и остаются в настоящее время транспортная и производственная логистика (9 разработанных приложений), телекоммуникации (9 приложений), электронная коммерция (4), аэрокосмические (4) и военные приложения (3), энергетика (2) и управление бизнес-процессами (2). Неожиданными аутсайдерами в этом аспекте оказались системы в области здравоохранения (2), робототехники (2), в финансовой сфере (0), в области административного управления (0), и в ряде других классов приложений, которым в [6] предсказывались гораздо лучшие перспективы.

Из конкретных данных, приведенных в данном разделе по результатам анализа литературы, можно сделать следующие выводы:

1. МАС-технологии развиваются гораздо сложнее, медленнее и труднее, чем это хотелось бы научному сообществу, и тем более, чем это необходимо индустриальному сообществу. Немало способствует этому и сложившаяся система финансирования научных грантов в Европе, которая требует от исследователей не «стоять» долго в области конкретных технологий, но развивать новые направления и решать конкретные прикладные задачи общеевропейского масштаба.

2. Основные исследования и разработки ведутся в основном в научном сообществе. Индустриальные компании, которые на начальном этапе были, по сути, инициаторами МАС-разработок и ранее играли ведущую роль в стимулировании и финансировании разработок в области МАС, в настоящее время фактически полностью ушли от поддержки этих разработок (Motorola, Siemens и др.). Некоторые компании, например, IBM, Daimler, NASA, Google активно используют концепцию агентов как часть своих разработок, но не называют их агентскими и не акцентируют внимание на агентских

компонентах своих разработок. Ряд компаний (British Telecom, например) снизил объемы агентских разработок [7].

3. Интерес индустриального сообщества к использованию МАС и МАС-технологий к настоящему времени просто упал. При этом наблюдается высокая скрытая турбулентность, перегруппировка и перестройка зарождающегося рынка интеллектуальных информационных систем и технологий. На этом рынке МАС и МАС-технологии реально имеют много конкурентов, и эти конкуренты активно предлагают свои решения для приложений, которые изначально относились к компетенции МАС, вытесняя тем самым последних с этого рынка.

Очевидно, что в настоящее время МАС и технологии во многом проигрывают своим конкурентам на рынке интеллектуальных информационных технологий индустриального уровня. Но, с другой стороны, очевидно также и то, что в последние годы постоянно расширяется *число новых ниш и новых классов приложений*, в которых имеются большие перспективы для технологии агентов и МАС. Поэтому очень важно выяснить причины, которые в настоящее время тормозят практическое использование огромного потенциала, которыми МАС и МАС-технологии, бесспорно, обладают. Это вопрос рассматривается в следующем разделе.

**3. Что мешает практическому использованию огромного потенциала МАС?** Наверное, самое ценное, что в настоящее время реально предложила теория и практика МАС, это концептуализация сложных систем и задач. Именно естественная и понятная концептуализация модели и архитектуры программной реализации систем практически любой сложности привлекает внимание и исследователей, и разработчиков приложений. Если проанализировать литературу по МАС-приложениям, а также тематику докладов на ведущих конференциях по этой тематике, то можно видеть, что подавляющая часть их посвящена концептуальным моделям приложений и архитектур их программной реализации. Следует заметить, что эта тенденция наблюдается и в настоящее время, хотя и не так очевидно, как это было примерно до 2010 г. И это естественно, поскольку на этом этапе разработки приложений парадигма МАС предлагает понятный и привлекательный вариант проектирования.

Однако уже на следующем этапе, когда на базе концептуальной модели нужно построить формальную модель агента, создать архитектуру инфраструктуры, поддерживающей взаимодействие агентов, и описать язык общения агентов, все оказывается, мягко говоря, не так просто, и причины этого нельзя назвать субъективными. Сложность шага формализации модели и архитектуры МАС многократно возрастает.

тает. Решения, которые были предложены для этого этапа специалистами в области формальных моделей, по большей части отторгаются теми, кто заинтересован в их практическом использовании.

Рассмотрим основные причины такого положения дел с индустриальными приложениями МАС-технологий.

1. Отсутствие общепринятого понимания ключевых понятий МАС. На это, например, указано в работе [9]. Отсутствие четких определений и соглашений по основным понятиям в области МАС сильно затрудняет взаимопонимание между исследователями и разработчиками. Например, большинство профессионалов в области компьютерных технологий согласны с определениями основных концепций ООП, такими как классы, объекты, наследование, инкапсуляция. Они легко оперируют этими понятиями на практике. В то же время специалисты в области МАС имеют различное понимание таких базовых понятий, как агент, роль, переговоры, план, возможность и другие. Реальная проблема в области МАС состоит в том, что необходимо уточнить содержание этих базовых понятий и согласовать их взаимоотношения со сходными концепциями ООП, которые используются для программирования агентов и МАС. Возможно, одним из путей преодоления данной проблемы является разработка базовых онтологий для предметно-независимых понятий, используемых в МАС. Один из примеров онтологии предметно-независимых поведенческих понятий BDI-модели агента и МАС можно найти в работе [17].

2. Отсутствие общепринятой нотации для представления моделей МАС и их компонент. Поскольку еще не существует общепринятых определений базовых агентских понятий и их взаимосвязей [9], то отсутствие общей нотации для их описания и описания отношений на их множестве затрудняет исследование различных моделей МАС на практике. Стандартизация моделей и способов их представления необходимы, иначе трудно сравнивать различные подходы, модели и архитектуры МАС, что может привести к неадекватным оценкам новых подходов в области МАС-технологий.

3. Концептуальная и вычислительная сложность логической формализации BDI-модели агента и МАС на основе этой модели. Почти три десятилетия основные усилия исследователей в области теории МАС направлены на разработку моделей *интеллектуальных агентов*. Уже в самом начале этих исследований был сформулирован ряд свойств, которыми должен обладать агент. Как правило, каждый отдельный агент рассматривался как интеллектуальная сущность с собственной развитой базой знаний или по крайней мере моделью знаний, средствами целеполагания и механизмами планирования целенаправленного поведения в непредсказуемой внешней среде. Эта точка

зрения достаточно активно пропагандируется в течение многих лет. Можно сказать, что специалисты в области описания формальных моделей «соревновались» в том, чтобы обеспечивать агента все новыми и новыми интеллектуальными возможностями в части автономного поведения вплоть до способности определять намерения других агентов. Естественно, что простыми средствами такие интеллектуальные возможности агента описать и реализовать непросто, что привело к постоянному усложнению формальной модели агента и МАС.

С самого начала в качестве базовой формальной модели интеллектуального агента была выбрана *BDI*-модель (*BDI* от англ. *Belief – Desire – Intention, Убеждение – Желание – Намерение*) [18], в которой знания, убеждения, намерения и механизмы рассуждений агента описываются в терминах исчисления предикатов, расширенного модальными и темпоральными операторами.

Большинство исследователей в настоящее придерживаются понятий *BDI*-модели агента и МАС и их логической формализации. Важно отметить, что сама по себе *концептуальная основа BDI-модели* в терминах поведенческих и мотивационных понятий достаточно естественна и убедительна. Однако ее логическая формализация определенно сложна для понимания и является существенным барьером в интерпретации базовых понятий МАС [9], включая понимание самого термина *BDI*.

Если обратиться к истории развития искусственного интеллекта, то можно вспомнить, что в нем в течение достаточно долгого периода господствовали логические языки представления знаний и логический вывод в качестве механизма рассуждений. Можно увидеть подходящую аналогию этой истории с тем, что происходило и, к сожалению, все еще происходит в настоящее время в области теории агентов и МАС. Более того, слабые выразительные возможности исчисления предикатов первого порядка для описания моделей автономных агентов и их распределенного взаимодействия предлагается преодолевать за счет его обогащения модальными и темпоральными операторами. Именно эта модель и получила название *BDI*-модели.

Но, к сожалению, эта модель теоретически намного сложнее исчисления предикатов первого порядка, и надеяться на ее практически приемлемую эффективность в задачах индустриального уровня вряд ли следует. К тому же в модели *BDI*-агента взаимодействие агентов, в основном, имеет характер простых диалогов, а потому в ней нет средств для формализации сложных сценариев поведения агентов МАС. Эта задача возлагается на логические модели, с помощью которых агенты должны сами формировать модели других агентов и модель внешней среды для принятия решений. На практике это оказалось

невозможным. Следует заметить, что такой подход означает отказ от сути многоагентной парадигмы, которая формулируется как вычисления на основе взаимодействий и по сути есть шаг в самоорганизацию.

Итог активного продвижения логической формализация *BDI*-модели получился ожидаемым: как и логическая модель искусственного интеллекта, она породила много новых математических проблем и интересных задач в области неклассических логических исчислений, что сильно способствовало развитию соответствующих разделов математики. Но это совсем не способствовало созданию эффективных моделей и технологий в области МАС. Более того, это затормозило практическое использование МАС-технологий на десятилетия. В настоящее время имеется глубоко разработанная логическая теория *BDI*-модели агента, которая теоретически позволяет строить агентов высокого уровня интеллектуальности, способных к планированию целенаправленного поведения и автономному принятию решений, к распределенной координации поведения в достаточно сложных ситуациях. Но эти возможности отвечают уровню теорем существования в математике, так что при попытках практического использования логической модели *BDI*-агента уже для относительно простых приложений возникают серьезные проблемы вычислительной сложности. Убедительным примером подобной ситуации являются *BDI*-модели коллективного поведения роботов, которые активно финансировались DARPA в течение почти десятилетия. В конце 1990-х годов они рассматривались сообществом специалистов в области МАС как самые значительные достижения [19, 20]. Однако эти результаты могли быть продемонстрированы лишь авторами и лишь на очень простых примерах. С начала 2000-х годов информация о дальнейшем развитии этих проектов отсутствует в литературе, и они оба, возможно, просто закрыты. Можно сказать, что концептуально понятную и естественную модель коллективного поведения агентов в автономной миссии погубила логическая модель *BDI*-агентов и их взаимодействия.

Заметим, что в это время, хотя и на вторых ролях, развивались и другие модели агентов и МАС. И именно эти модели главным образом были использованы в большей части тех приложений, которые рассматриваются как успешные в [7, 16] и других работах.

4. Стандарты FIPA. Вопрос о стандартах в технологии МАС стал темой исследований еще в середине 1990-х годов, когда была создана общественная организация FIPA, предназначенная для научного обоснования стандартов МАС-технологий. Но FIPA была создана тем же научным сообществом, которое продвигало в практику логическую

модель *BDI*-агента и *MAC*, поэтому эта же модель рассматривалась *FIPA* в качестве базовой модели *MAC* и при разработке стандартов.

Например, стандартный язык коммуникации агентов *ACL* (от англ. *Agent Communication Language*) [21] для описания содержания сообщений, которыми обмениваются агенты, использует весьма сложный язык. В основе своей это тоже логический язык, который начали разрабатывать еще в 1970-е годы, когда он позиционировался как язык представления знаний в системах искусственного интеллекта с исчислением предикатов в своей основе. Современная его версия, получившая название *ACL*, представляет собой достаточно мощный и выразительный язык интерпретирующего типа, который манипулирует понятиями онтологии и способен представлять содержание сообщений, которыми обмениваются агенты, на языке, близком к естественному языку. Но он привносит в стандарт все черты логической модели *BDI*-агента со всеми вытекающими отсюда последствиями из-за проблем вычислительной сложности и большой загрузки каналов связи. С другой стороны, он труден для понимания и использования разработчиками приложений, а свобода, которую язык *ACL* вроде бы предоставляет разработчикам в модификации компонент синтаксической оболочки этого языка (так называемых перформативов), используется главным образом в исследовательском сообществе специалистов.

На практике же в большинстве случаев оказывается возможным обойтись значительно более простыми специализированными языками. Примером специализированного языка является язык обмена сообщениями, принятый в сервере *RoboCup* [22]. Он использует только необходимые и достаточные средства общения агентов и поэтому весьма эффективен. Например, для двух команд симуляционного футбола по 11 агентов каждая достаточно обычного персонального компьютера выпуска 2004 года, чтобы реализовать интенсивный обмен сообщениями между агентами команд в режиме реального времени. Совсем другой, более простой и прагматичный язык используется в модели *MAC*, предложенной сначала в инструментах компании *Magenta*, а затем развитый (теми же разработчиками) в инструментах Группы компаний *Генезис знаний* и *НПК Разумные решения* [23], где для представления текущих знаний активно используется понятие *сцены (модели ситуации)* как хранилища сведений о сети потребностей и возможностей (*ПВ-сети*) в окружающем мире.

Иная модель обмена сообщениями по сравнению со стандартом *FIPA* используется также в программном инструментарии *Cougaar* [24], который разработан в США по проекту *DARPA* для его использования в военных приложениях. В нем обмен сообщениями поддерживается стандартной архитектурой доски объявлений, на которой агенты пред-

лагают свои сервисы (доступные другим агентам как подключаемые модули) и ищут необходимые им сервисы по мере необходимости. При этом доска объявлений заметно расширяет множество доступных сервисов за счет веб-сервисов и *UDDI*-протокола [25]. Эта возможность реализуется специальной компонентой доски объявлений, которая называется *сервлет*. Она поддерживает коммуникации с объектами Интернет по *http*-протоколу, причем этот сервис доступен для использования всем подключаемым модулям узла.

Важно подчеркнуть, что модели агентов и МАС, предложенные в инструментах компании *Magenta* и *Cougaar*, оказались наиболее успешными в части индустриальных разработок, хотя они и не используют стандартную платформу FIPA.

К большому недостатку FIPA-стандартов следует отнести также и то, что спецификации FIPA вообще не рассматривают проблемы *параллельного программирования*, хотя МАС — это концепция, изначально ориентированная на параллельные вычисления.

5. Отсутствие гибких промышленных методов и технологий для разработки МАС-приложений. В работе [9] отмечается, что разработчики промышленных многоагентных систем сталкиваются с множеством агентских методологий и в большинстве случаев — с отсутствием программных инструментов их поддержки. Это, возможно, связано с тем, что новые методологии являются недостаточно гибкими для распространения на широкие области применений. В большинстве инструментов агенты служат только для дополнительной переупаковки объектов при помощи ООП, что дает некоторые преимущества, но значительно снижает возможности агентского подхода. Поэтому насущной задачей является интеграция существующих методологий разработки и инструментов их поддержки в единую хорошо определенную технологию [26].

Одной из причин негативных тенденций в области промышленного использования многоагентных технологий является не лучшая стратегия в области методологий и средств разработки МАС. С самого начала исследователи в области МАС придавали большое значение созданию методологий проектирования и программной реализации МАС, причем в период 2005-2010 гг. эти исследования и разработки велись наиболее активно. В этот период общее мнение сводилось к тому, что хорошая методология разработки и мощный инструментарий разработки позволят в значительной степени автоматизировать создание МАС-приложений промышленного уровня, и тогда их создание можно будет поставить на поток.

В период до 2010 г. было разработано более десятка методологий проектирования и разработки МАС, если не считать большого количества других менее значимых методов и методологий. К числу наиболее перспективных и глубоко проработанных методологий следует отнести такие методологии как *Gaia* [10], *Tropos* [11], *MaSE* [12], *ADELFE* [13], *MESSAGE* [14], *Prometheus* [15], *SADDE* [27] и ряд других. Большинство этих методологий сопровождалось также созданием инструментальных программных средств их поддержки (например, [28]). Эти средства использовали либо авторскую методологию разработки, либо одну из списка, приведенного выше. Например, к этому времени были разработаны такие средства как *agentTool* [29], *Zeus* [30], *agentBuilder* [31], *PASSI* [32], *MASDK* [33] и ряд других.

Все эти инструменты для своей разработки потребовали длительных усилий больших коллективов высокопрофессиональных исследователей, разработчиков, программистов и тестеров. Обычно на разработку и тестирование методологии и поддерживающего ее программного инструментария уходило не менее 10 лет.

Как правило, передовые методологии этого периода использовали концепцию разработки, управляемую моделью (англ. *model-driven engineering*). В методологиях такого типа концептуальная модель прикладной МАС, формальные модели ее стандартных компонент (модели агентов, онтологии, протоколов, сообщений и пр.), а также модели взаимодействия компонент и архитектура целевого программного продукта описывались на некотором формальном языке, при этом предпочтение отдавалось графическим языкам.

Такой формальный язык должен был автоматически поддерживать непротиворечивость моделей компонент системы разного уровня абстракции и генерацию протоколов взаимодействия агентов системы. Предполагалось, что формальная модель приложения, построенная таким способом, далее будет компилироваться в код программы на языке высокого уровня (например, в код *C++*, *Java* или в код другого языка аналогичного уровня). После этого система должна быть дополнена компонентами, для которых код может быть написан только вручную (это касалось «не агентских» компонент). Предполагалось, что полученный таким способом высокоуровневый код должен был далее компилироваться в исполняемый код.

Не трудно видеть, что инструментальная поддержка подобной методологии проектирования и программной реализации МАС-приложений является объективно «тяжелой», особенно в тех случаях, когда использовалась логическая модель *BDI*-агента и МАС. Как правило, методологии и инструментальные средства опирались на стандарты *FIPA*. По этой, а также по ряду других причин добиться желаемой эф-

фektivности методологии разработки и программной реализации приложения, так же, как и вычислительной эффективности целевого приложения, не удалось. В итоге такой методологической и инструментальной подход, за некоторыми редкими исключениями, себя не оправдал. Только отдельные программные прототипы удалось с его помощью довести до экспериментальных образцов. Некоторым исключением является инструментальный Living Systems® Technology Suite компании Whitestein Technology [34], который, однако, не использовался для массовой разработки МАС из-за трудоемкости, а что-либо про его коммерческую эффективность неизвестно.

Описанная стратегия разработки прикладных МАС активно пропагандировалась и практически использовалась в период 2005-2015+ гг. В литературе можно найти сведения о нескольких сотнях МАС-приложений, разработанных в этот период в различных прикладных областях. Но среди этих разработок большинство не получило последующего развития и применения, и их разработка была прекращена.

Та же судьба постигла и большинство методологий и поддерживающих их инструментальных программных средств: разработка и поддержка большей части из них в настоящее время прекращена.

6. Позиционирование МАС и МАС-технологий в прикладных областях. Как уже отмечалось, существует много классов приложений, которые ранее позиционировались как приложения, для создания которых архитектура и технология МАС являются наиболее перспективными. Однако практика показала, что многие из них уже давно имеют удачную программную реализацию индустриального уровня с помощью других технологий, в то время как их многоагентные реализации либо явно проигрывают таким реализациям, если они имеются, либо эти приложения к настоящему времени вообще не имеют агентской реализации. Причиной такой ситуации является то, что с самого начала МАС рассматривалась как достаточно универсальная ИТ-парадигма и технология, однако практика показала, что это не так, и было бы весьма полезно все-таки очертить область приложений МАС, а не уповать на универсальность этой парадигмы.

К настоящему времени область применений МАС и технологий, в которых они имеют неоспоримые преимущества, фактически не определена и еще только складывается. Перечисление классов потенциальных приложений в [6] базировалось в основном на слишком широком классе приложений и потому прогноз, данный в этой работе, не подтвердился. В этом документе, скорее, перечисляются те прикладные области, в которых агентские технологии могут быть использованы наряду с другими технологиями. В результате более зрелые технологии оказались в большей части случаев более успешными, что способствовало утверждению

негативных мнений о МАС-технологиях и снижению интереса индустриального сообщества к их использованию.

Можно утверждать, что для МАС-технологий пришла пора извлекать уроки и сузить возможный спектр применений. Практика уже сейчас формулирует много новых классов приложений, в которых для МАС-технологий альтернативы просто не просматривается. Примером являются задачи адаптивного управления ресурсами для приложений класса Интернета вещей и другие задачи современной цифровой экономики реального времени, извлечения знаний из больших неструктурированных данных и так далее.

7. Недостаточная зрелость математических моделей и методов самоорганизации для использования в сложных объектах сетевой структуры. Одной из важных причин затруднений, часто возникающих у разработчиков МАС, является отсутствие достаточно изученных адекватных математических моделей, методов и алгоритмов самоорганизации агентов, пригодных для недетерминированного решения сложных задач. Такие модели самоорганизации начали активно изучаться в научном сообществе лишь в самое последнее время, доказательством чему может служить рост числа докладов, посвященных принятию решений на основе консенсуса, на Международном конгрессе IFAC по автоматическому управлению [36].

На различных конференциях и практических семинарах очень часто разработчикам МАС задают такие вопросы, как:

– Какое качество результата может быть обеспечено при решении задач методами самоорганизации по сравнению с обычным комбинаторным подходом к оптимизации?

– Как в таких системах можно гарантировать эффективность функционирования системы?

– Что можно сказать о скорости сходимости процессов самоорганизации и устойчивости результата?

– Как быстро деградирует скорость решения задачи (вычислительная эффективность) с ростом размерности задачи?

И так далее.

Ответить на эти вопросы пока не всегда возможно, и часто «хороший результат» обосновывается лишь экспериментально. Но уже появились первые работы математиков, начавших активно изучать модели самоорганизации в различных приложениях [37], доказывающие, хотя пока и для относительно простых примеров, квазилинейность скорости поиска в МАС решений, близких к оптимальным. Кроме того, показана связь моделей построения самоорганизующихся расписаний с теорией сложных адаптивных систем и моделями нелинейной термодинамики, когда в МАС формируются «устойчивые неравновесия» и наблюдаются

близкие к «автокаталитическим» цепочки изменений [38]. При этом на практике уже в ряде случаев показано, что самоорганизующиеся МАС-приложения способны решать более сложные задачи по сравнению с другими методами, и, что еще важнее, в реальном времени.

В этой связи, можно привести много примеров различных прикладных задач из разных областей, в которых МАС-технологии являются единственным приемлемым вариантом на множестве всех существующих технологий. К ним относятся, например, крупномасштабные задачи управления объектами сетевой структуры, при решении которых классические модели, методы и средства оптимизации быстро «захлебываются» в переборе вариантов, а МАС-технологии оказываются способными в любом случае строить хотя бы допустимые решения. Имеются убедительные разработки, которые доказывают это [39, 40].

**4. Перспективы развития МАС-технологий в industriальных приложениях.** Несмотря на описанные негативные аспекты истории развития теории и технологии МАС, она вполне укладывается в классическую схему зарождения и выхода новой технологии из лабораторий в область промышленных применений, а этот процесс на практике никогда не бывает линейным (рисунок 1). В начале своего развития в 1990-е годы МАС стали вызовом в программировании и потребовали больших усилий выдающихся ученых и программистов для первых успешных разработок — вспомним синие экраны Norton Commander в DOS и первые Интернет-модемы того времени. Естественно, что не все решения оказались удачными.

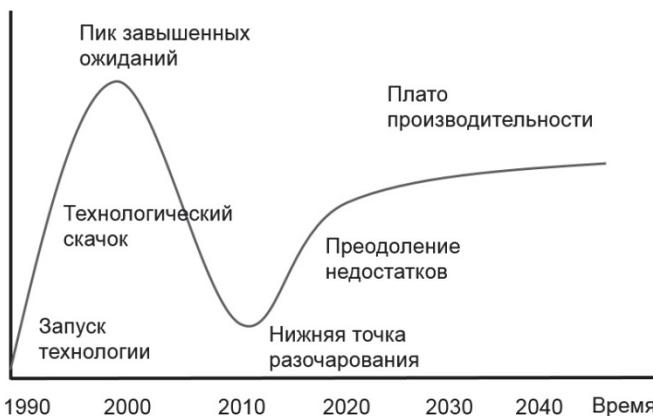


Рис. 1. Перспективы развития МАС и МАС-технологий в контексте стандартного цикла развития новых технологий

Несложно предсказать, что процесс развития такой революционной по своей сути технологии, как МАС-технология, будет и далее весьма нелинейным, что потребуются еще неоднократные попытки его трансформации и не все они будут успешными, прежде чем удастся сделать МАС-технология действительно продуктивной и коммерческой.

Как отмечено в [23, 41, 42], разработчиков МАС подстерегают серьезные проблемы организационного, коммерческого и тому подобного характера из-за конкуренции на рынке ИТ. В качестве примеров можно привести такие проблемы:

- разработки МАС в сфере управления обычно вторгаются в самую «запретную» область принятия решений, критическую для бизнеса, где существует уверенность, что там «все уже занято», хотя часто в наличии имеются только учетные системы, или в нем процессы принятия решений совершенно не формализованы;

- процесс поиска решений в МАС является распределенным и недетерминированным, причем со значительным акцентом на самоорганизацию, что на практике противоречит традициям иерархического управления и, что еще важнее, традициям проектирования систем по принципу «сверху–вниз», поскольку самоорганизующиеся системы разрабатываются только по принципу «снизу–вверх» [43], но далее решение можно гармонизировать сверху, наблюдая формирующееся «целое», что предполагает создание в будущем класса гибридных моделей, в которых начальное решение ищется обычными методами, а далее модифицируется по событиям в МАС;

- продажа инновационных разработок МАС требует глубокого участия предметных специалистов, а не только разработчиков, и занимает много времени (от 3 до 24 месяцев);

- разработка МАС приложений, важных для бизнеса (по опыту крупных проектов), требует затрат сил и времени примерно в 3-5 раза больше, чем ожидается вначале;

- объем разработки многоагентной системы управления («движка») занимает не более 25% общего времени, в то время как все остальное время тратится на другие вопросы, связанные с учетом, базами данных, интерфейсом пользователя, интеграцией и так далее;

- разработка первой версии многоагентной системы наиболее трудоемка и занимает от 3 до 6 месяцев (минимум) даже при наличии опыта использования МАС-технологии;

- внедрение МАС часто занимает больше времени, чем сама разработка, поскольку требует выявления и отработки правил принятия и согласования решений, а также ее интеграции с уже существующими

щими информационными системами заказчика. Примерное соотношение затрат труда (в %) по основным фазам проекта МАС разработки: проектирование — 10, разработка — 20, тестирование — 15, поставка, внедрение и обучение — 35, поддержка — 20;

– разработанная система должна «выживать» в условиях постоянных ошибок пользователей, наличия неполных данных для проектирования, поступления неточных данных и так далее;

– пользователи должны иметь возможность вручную дорабатывать решения, поскольку всегда остаются факторы, которые не представляется возможным учесть при автоматическом принятии решений в системе и тому подобное.

Преодоление негативных тенденций в области практического использования МАС, безусловно, потребует значительных усилий, но в случае успеха результаты окупят все вложения, обеспечивая на практике большие преимущества МАС-технологий. По существу, в настоящее время идет активная работа над ошибками МАС-технологий и преодоление недостатков.

Вместе с тем полноценный выход МАС на плато продуктивности на уровне отраслевых стандартов, по-видимому, можно ожидать не ранее 2020-2025 гг., причем в числе наиболее перспективных областей использования МАС-технологий следует ожидать:

– *Аэрокосмическая отрасль* — коллективное самоорганизующееся поведение беспилотных космических и летательных аппаратов, управление группировками малых спутников, тренажеры для летчиков и авиадиспетчеров, космическая логистика и др.

– *B2B сети производственных и транспортных предприятий*, стратегическое планирование и оперативное управление производством, сетевая логистика (транспортная и др.), где уже используются агентские решения и технологии, но еще остается большое поле для использования сетевых МАС-технологий [39].

– *Военные приложения* — имеется много косвенных признаков и свидетельств активности индустриальных разработок в этой области за рубежом, однако эта информация закрыта.

– *Коллективная робототехника, автономные миссии роботов* — пока находится в стадии изучения возможностей, однако специалистами всего мира рассматривается как одна из самых перспективных областей для многоагентных приложений.

– *Смарт-грид, виртуальные электростанции* и другие приложения в области энергетики. Эти разработки уже начаты 4-5 лет тому назад, и наблюдается рост их активности с использованием МАС-технологий.

– *Здравоохранение* — решения в области окружающей среды для поддержки здоровья населения (англ. *ambient assisted living, personal healthcare* и др.).

– *Задачи в области Интернета вещей*, где нужно разрабатывать распределенные системы обеспечения безопасности, сенсорные сети, интеллектуальные пространства и тому подобное [35, 44, 45].

– *Мобильные приложения* — по оценкам компании Gartner до 40% будущих мобильных приложений в течение ближайших 10 лет будут построены на основе MAC-технологий.

– Виртуальные (оверлейные) сети и *peer-to-peer* (p2p) — приложения (англ. *Software-Defined Networking*) — эта парадигма взаимодействия распределенных приложений сейчас очень востребована.

Но уже и в настоящее время имеются MAC-приложения, которые демонстрируют свои преимущества на практике. Приведем два примера приложений, разработанных с помощью авторских MAC-технологий, которые лишены ряда недостатков, отмеченных в разделе 3 как главные причины неудач MAC-технологий в индустрии ИТ.

1. Самоорганизующиеся В2В-сети предприятий [39]. В2В-сетями называют вид экономического и информационного взаимодействия множества предприятий, имеющий целью координацию планирования и исполнения распределенных бизнес-процессов *в интересах самого бизнеса*. Узлами сети являются предприятия, которые на долгосрочной основе кооперируются в выполнении потока заказов, поступающих в произвольные узлы сети в режиме реального времени. Полагается, что В2В-сеть является открытой, то есть к ней могут присоединяться новые узлы, они также могут выходить из сети и возвращаться в нее снова. Эта сеть является p2p-сетью, в которой как коммуникации, так и содержательные взаимодействия поддерживаются с помощью p2p-протоколов.

В общем случае полагается, что заказы могут поступать в сеть через произвольные ее узлы в режиме реального времени. Заказ может состоять из сложно структурированного множества подзаказов, которые по тем или иным причинам не могут быть выполнены одним узлом сети. Причины могут быть технологическими (предприятие не имеет возможности выполнить все компоненты заказа), ресурсными (предприятие перегружено заказами и может выделить на новый заказ лишь ограниченные ресурсы), или экономическими (распределенный процесс выполняется специализированными предприятиями, что обходится дешевле) и другое [39]. Далее рассматривается MAC-архитектура и технология решения задачи координации расписаний

распределенного выполнения заказов в режиме реального времени, одной из наиболее сложных задач, решаемых в B2B-сетях.

MAC-архитектура узла B2B-сети представлена на рисунке 2. Алгоритм распределенной координации локальных расписаний узлов B2B-сети строится на основе обмена метаданными между ее узлами, представленными в системе составления расписания агентами трех типов:

– *Агент узла Department Agent, DA*, представляющий в сети некоторое предприятие — он ответственен за составления локальных расписаний узла и за взаимодействие с *DA*-агентами других узлов при распределенной координации расписаний на метауровне.

– *Агент ресурса Resource Agent, RA*, — участвует в выполнении *Протокола контрактных сетей* при составлении локального расписания.

– *Локальный агент заказа Local Order Agent, LOA* — участвует в составлении локального расписания узла, отвечая за соблюдение технологии выполнения локальной части заказа.

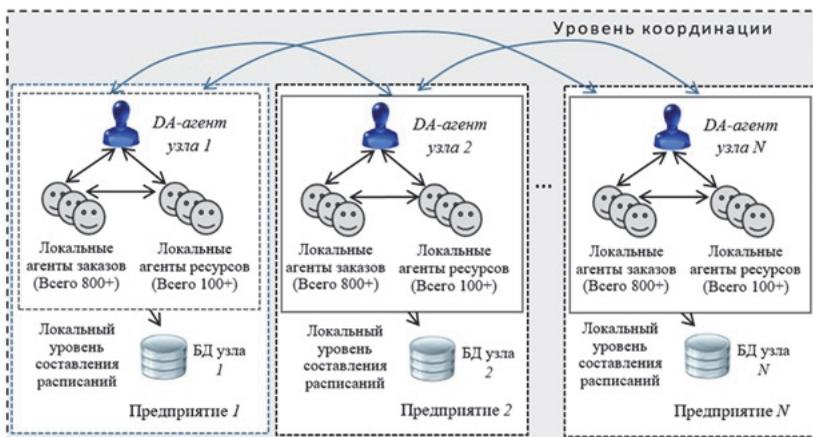


Рис. 2. Архитектура MAC распределенного составления расписаний выполнения взаимосвязанных заказов в производственной B2B-сети

Детали взаимодействия этих агентов при составлении *локального* расписания описаны в [46].

В системе имеется также еще один агент, называемый Агентом заказа *Order Agent, OA*, который является агентом метауровня. Он владеет информацией о заказе в целом, например о самом раннем допустимом времени начала исполнения заказа и о самом позднем допустимом времени его завершения. В этой архитектуре информация о ресурсах и технологиях узлов доступна только агентам соответствующих узлов, что решает задачу конфиденциальности. Важно отметить,

что задача координации локальных расписаний решается на основе взаимодействия *DA*-агентов узлов, а также их взаимодействия с мета-агентами *OA*. Протокол их взаимодействия описан в [46]. Поддержка этого взаимодействия выполняется с помощью распределенной программно-коммуникационной платформы, которая реализует традиционные сервисы платформы, минимальный набор которых включает в себя сервисы белых и желтых страниц, а также сервис адресации сообщений на основе того или иного р2р-протокола [47].

Однако архитектура и реализация этой платформы (среды исполнения агентов и среды их взаимодействия) значительно отличаются от той, которая диктуется стандартами FIPA. В этой архитектуре средой исполнения всех агентов является *Akka System*, которая играет роль *локальной* компоненты р2р-платформы агентов. Она реализует функции управления потоками, диспетчеризацию сообщений, управляет жизненным циклом агентов [48], является каркасом для дополнительных модулей системы и отвечает за ряд других важных задач. Все узлы сети разделяют общую онтологию *Ontology*, то есть одну модель данных.

Эффективная реализация рассматриваемой системы [47] с помощью *MAC*-архитектуры и технологии стала возможной благодаря тому, что в ней используется простая модель агента, в основе которой лежит понятие *актора* [48], которая намного проще *BDI*-модели. В этой реализации взаимодействие распределенных компонент поддерживается нестандартной агентской платформой, которая намного «легче», чем распределенная р2р агентская платформа, предусматриваемая стандартом FIPA. Технология *Akka* предлагает хорошую модель организации параллельных\распределенных вычислений, основанную на модели акторов, которая избавляет разработчиков от ряда острых проблем, не решенных в классических FIPA-платформах, таких как *JADE* [49]. Взаимодействие агентов основано на передаче простых событий, а не сложных *ACL*-сообщений, что также повышает вычислительную эффективность *MAC*-реализации рассматриваемого весьма сложного приложения, сохраняя при этом все концептуальные преимущества парадигмы *MAC*.

2. Адаптивное управление мобильными бригадами. Рассмотрим задачу управления крупной региональной сетью аварийно-ремонтных бригад. Организационно система управления включает в себя центр приема заявок от населения по телефону (далее для краткости «*колл-центр*» от англ. *call center*), центральную диспетчерскую и специально оснащенные мобильные бригады техников по ремонту и обслуживанию объектов в сетях газоснабжения, водоснабжения, водоотведения, телекоммуникации и других в районе с сотнями тысяч жителей.

Заявки от населения («аварийные заявки») характеризуются срочностью устранения аварии (средняя, важная и критическая), местом проведения работ и видом работ. Помимо заявок от населения на устранение аварий, существуют также планово-профилактические работы, которые проводятся этой же службой бригад согласно утвержденному ежемесячному графику. Для более сложных работ требуются специалисты более высокой квалификации. Для некоторых видов работ требуется наличие у бригады дополнительного оборудования, например сварочного аппарата. Каждая бригада имеет в распоряжении специальный автомобиль для выездов по заявкам, оснащенный необходимым оборудованием. В любой момент времени местоположение бригады определяется датчиком GPS/ГЛОНАСС-навигации специального автомобиля бригады. Каждый бригадир имеет мобильный терминал, через который он «видит» план выполнения заявок, назначенных на его бригаду, и может делать фактические отметки о выезде, о начале и об окончании работ по каждой заявке. Аварийные заявки имеют горизонт планирования на текущий день, плановые — в течение месяца. Невыполненные заявки текущего дня переносятся на следующий день.

Содержание задачи управления мобильными бригадами состоит в следующем:

1) Автоматически распределить и выполнить открытый пул заявок текущего дня наискорейшим образом в порядке срочности их выполнения и в соответствии с их приоритетами — *критическая, важная, средняя* или *плановая* заявка. Задача планирования работ имеет достаточно высокую сложность ввиду большого числа ежедневных заявок и ограниченным количеством бригад. Например, в Средневожской газовой компании (г. Самара) обычно бывает порядка 50 заявок в день при наличии 25 бригад, а в Водоканале г. Волгограда — порядка 64 заявок при том, что общее количество бригад равно 15. При этом большую роль играет фактор времени.

2) Дополнительно нужно оперативно, гибко и эффективно реагировать на поток заявок (событий) реального времени, а именно на поступление более срочных заказов или на внезапную поломку одного из специальных автомобилей бригад в пути, что приводит к недоступности бригады. Например, при поступлении более срочной заявки нужно определить бригаду, которая сможет ее выполнить и прибыть на место раньше других с учетом других заявок своего плана, и при этом нужно также оперативно перестроить расписание работ этой бригады.

3) Управлять маршрутом движения бригады к месту работ с учетом текущей дорожной ситуации (дорожные знаки, разметка, пробки) с тем, чтобы сократить время движения.

4) Сократить простой квалифицированных и хорошо оснащенных бригад, которые должны гибко маневрировать расписанием, переходя от срочных заявок к профилактике и наоборот.

5) Обеспечить индивидуальный подход к каждой поступившей заявке и ресурсу, назначаемому для ее выполнения. За выполнение каждой заявки бригаде начисляется бонус в зависимости от сложности выполненных работ согласно регламенту сервисной службы, при этом основным показателем качества работы бригады является фактическое время ее выполнения и отсутствие повторных выездов на следующий день. Задача управления мобильными бригадами решается с помощью адаптивного МАС-планировщика, способного планировать задачи обслуживания в реальном времени с минимизацией пробега автотранспорта бригад, времени выполнения заявок на обслуживание и общее число невыполненных заявок клиентов в текущий день.

Архитектура разработанного МАС-планировщика включает в себя виртуальный мир программных агентов, которые могут обмениваться сообщениями для того, чтобы информировать о своих возможностях и параметрах состояния (о потребном времени на выполнение планируемой заявки, стоимости ее выполнения, о своих координатах и др.).

Сущность работы МАС-планировщика состоит в кролящей адаптивной обработке сообщений о событиях, в ходе которой план гибко корректируется без остановов системы. При отсутствии внешних событий и нагрузки по выполнению аварийных заявок бригады выполняют плановые работы по поддержке/обслуживанию сетей предприятия. Архитектура системы представлена на рисунок 3.



Рис. 3. Логическая архитектура системы адаптивного управления аварийно-ремонтными бригадами

Основной оценкой состояния и возможности выполнения заявки является суммарное время поездки бригады до места выполнения заявки и время ее выполнения.

В отличие от сложных BDI-моделей агентов, в разработанном MAC-планировщике используются простые агенты с интенсивными переговорами. Сам планировщик представляет собой распределенную систему активно взаимодействующих агентов следующих типов:

- *Агент заявки*, ответственный за поиск наиболее подходящей специализированной бригады, способной выполнить работу как можно скорее и с нужным качеством.

- *Агент региона*, который помогает балансировать распределение машин между районами города с учетом прогноза поступления заявок.

- *Агент маршрутов перемещений*, чья цель состоит в том, чтобы предложить бригаде маршрут движения к месту аварии с минимальным временем в пути и таким образом способствовать увеличению числа заявок, выполняемых бригадами за смену.

- *Агент бригады*, целью которого является подбор и согласование заявок и маршрутов с целью повышения продуктивности работы бригады.

- *Агент предприятия*, стремящийся улучшить наиболее проблемные фрагменты расписания мобильных бригад компании.

Все решения по распределению и планированию работы бригад, которые формируются с помощью переговоров агентов по соответствующим протоколам и самоорганизации, предлагаются диспетчеру. Он может при необходимости интерактивно перестроить расписание любой бригады (рисунок 4). Типовой протокол взаимодействия агентов при поступлении заявки на аварийное обслуживание с описанием аварийной ситуации через интерфейс колл-центра описан в работе [26].

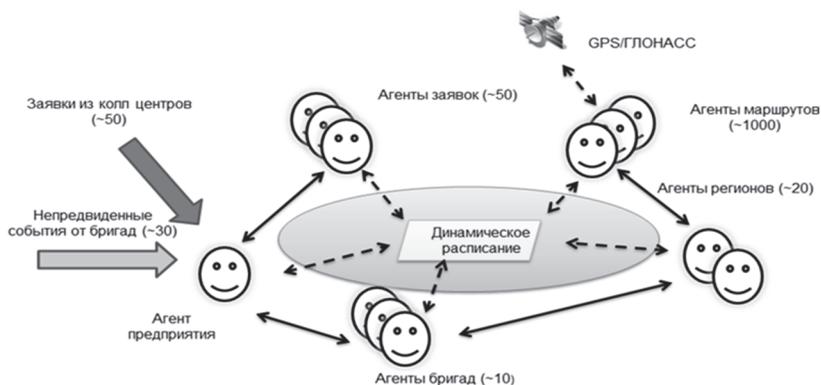


Рис. 4. Сцена мира агентов адаптивного управления аварийно-ремонтными бригадами (штрих линия — чтение из сцены и изменения в сцене, сплошная линия — обмен сообщениями и переговоры агентов)

В результате обмена сообщениями текущее расписание по событиям постоянно адаптивно подстраивается под новые заявки и динамику дорожной ситуации. Время, необходимое для оценки вариантов размещения заявки составляет обычно не больше 4 секунд.

Использование адаптивного планировщика для обработки поступающих заявок в реальном времени позволило достичь повышения продуктивности обслуживающих бригад на 40%, поскольку каждая бригада газовой службы в настоящее время выполняет в среднем 12 задач (ранее эта цифра была равна 7 задачам в день). В результате его внедрения в Средневожской газовой компании, которая уже несколько лет успешно эксплуатирует описанную систему, удалось добиться заметного улучшения различных показателей работы [26].

**5. Заключение.** В научном сообществе МАС-технологии уже долгое время рассматриваются как достаточно мощная интеллектуальная ИТ, способная успешно конкурировать с другими технологиями в широком классе приложений. Однако это мнение до настоящего времени пока не подтверждается реальными разработками. Более того, большое число классов приложений, для которых МАС-технология считалась приоритетной, к настоящему времени успешно реализованы другими более зрелыми технологиями. Это в определенной мере подорвало доверие к МАС-технологиям со стороны индустриального сообщества и, соответственно, интерес к финансированию разработок.

Большие негативные последствия для авторитета МАС-технологий обусловлены также недооценкой сложности разработки МАС-приложений, которая требует от разработчиков не только высокой алгоритмической квалификации и хороших навыков объектного программирования, но также и навыков программирования параллельных и асинхронных вычислений, привлечения моделей искусственного интеллекта и активного включения в процесс использования достижений в области телекоммуникаций и их дальнейшей разработки.

Серьезным внутренним препятствием для развития МАС стала ориентация базовой модели агента МАС на использование логического языка, дополненного модальными и темпоральными операторами для формального описания *BDI*-модели поведения агентов. Эта модель, весьма привлекательная из-за ее богатых выразительных возможностей и математической корректности, на практике оказалось непригодной для создания промышленных систем из-за вычислительной сложности и, соответственно, трудности создания и использования инструментальных программных средств поддержки технологии МАС и, как следствие, высокой стоимости реализации разработок. И хотя постоянно возрастает

множество потенциальных МАС-приложений, исследования в области теории и практики МАС проводятся практически без учета потребностей индустрии.

Тем не менее агентские модели и технологии по-прежнему остаются перспективными для большого количества новых и будущих приложений, в особенности для тех из них, которым свойственны такие черты, как множественность и автономность объектов приложения, сетевая структура их взаимодействия, высокая неопределенность среды функционирования и структурная динамика поведения, большой масштаб. Эффективное управление в таких системах может быть достигнуто только за счет использования принципов самоорганизации и распределенной координации.

Новые поколения таких систем строятся на основе концепций Интернета вещей и Интернета людей, в которых каждая «вещь» рано или поздно должна стать «умной», а пользователь — получить своего ассистента, помогающего решать проблемы реальной жизни в реальном времени. Любая вещь в этом подходе рано или поздно должна будет обладать не только датчиками и устройствами воздействия на объекты внешнего мира и коммуникации, но и принимать решения, причем по возможности согласованно с другими вещами. Эти требования, в свою очередь, порождают концепцию Интернета агентов как новой версии технологии многоагентных систем, способной реализовать на практике принципы самоорганизации и кооперации всех компонент сложной системы [35].

Для того чтобы эта технология стала продуктивной и способной обслуживать потребности новых промышленных приложений, необходима смена базовой парадигмы формализации модели агента и программно-коммуникационной инфраструктуры с тем, чтобы технология МАС стала по-настоящему эффективной и продуктивной, была способна обеспечить поддержание полного жизненного цикла агента, его взаимодействия с внешней средой и облачными сервисами. В новой парадигме формализации МАС-приложение рационально строить как множество простых агентов с богатой компонентой взаимодействия и с широким использованием принципов самоорганизации и эволюции, присущих живым системам — в конечном счете для создания цифровых экосистем (систем систем), построенных как конкурирующие и кооперирующиеся сервисы.

### **Литература**

1. Digital Ecosystems. URL: [https://en.wikipedia.org/wiki/Digital\\_ecosystem](https://en.wikipedia.org/wiki/Digital_ecosystem) (дата обращения: 04.11.2017).
2. Digital Business Ecosystems & The Platform Economy. URL: <http://www.gartner.com/technology/topics/business-ecosystems.jsp> (дата обращения: 04.11.2017).

3. <https://www.linkedin.com/pulse/industrial-upcycling-definition-michael-rada/> (дата обращения: 04.11.2017).
4. *Kaminka G.* Robots Are Agents, Too! // Keynote Lecture. International Conference on Autonomous Agents and Multi-agent Systems (AAMAS 2007). 2007. pp. 4.
5. The First International Workshop on the Internet of Agents (IoA). 2016. URL: <http://www2.cs.siu.edu/~salqithami/ia/> (дата обращения: 04.11.2017).
6. *Luck M., McBurney P., Shehory O., Willmott S.* Agent technology: computing as Interaction (a roadmap for agent based computing) // University of Southampton Department of Electronics & Computer Science. 2005. 108 p. URL: <http://www.agentlink.org/roadmap/> (дата обращения: 04.11.2017).
7. *Müller J., Fisher K.* Application Impact of Multi-agent Systems and Technologies: A Survey // Agent-oriented software engineering. 2014. pp. 27–53.
8. *Strassner J.* Using agents and autonomic computing to build next generation seamless mobility services // Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems. 2007. Article no. 1.
9. *DeLoach S.A.* Moving multi-agent systems from research to practice // International Journal Agent-Oriented Software Engineering. 2009. vol. 3. no. 4. pp. 378–382.
10. *Zambonelli F., Jennings N., Wooldridge M.* Developing Multi-agent systems: The GAIA methodology // ACM Transactions on Software Engineering and Methodology. 2003. vol. 12(3). pp. 417–470.
11. *Brinkkemper J., Solvberg A.* Tropos: A framework for requirements-driven software development // Info. Syst. Engg.: State Art Res. Themes. 2000. LNCS 1. pp. 261–273.
12. *DeLoach S.* Analysis and Design using MaSE and agentTool // Proceedings of the 12<sup>th</sup> Midwest Artificial Intelligence and Cognitive Science Conference (MAICS). Miami University Press. 2001.
13. *Bernon C., Gleizes M.P., Peyruqueou S., Picard G.* ADELFE: a methodology for adaptive multi-agent systems engineering // International Workshop on Engineering Societies in the Agents World. 2002. pp. 156–169.
14. *Caire G. et al.* Agent Oriented Analysis using MESSAGE/UML // International Workshop on Agent-Oriented Software Engineering. 2001. pp. 119–135.
15. *Padgham L., Winikoff M.* Prometheus: A Paradigm Methodology for Engineering Intelligent Agents // Proceedings of the Workshop on Agent-Oriented Methodologies. 2002. pp. 97–108.
16. *Leitao P., Vrba P.* Recent Developments and Future Trends of Industrial Agents // Holonic and Multi-Agent Systems for Manufacturing. 2011. LNCS 6867. pp. 15–28.
17. *Городецкий В.И., Самойлов В.В., Троцкий Д.В.* Базовая онтология коллективного поведения автономных агентов и ее расширения // Известия РАН: Теория и системы управления. 2015. № 5. С. 102–121.
18. *Wooldridge M.* An Introduction to Multi-Agent Systems // John Wiley & Sons. 2009. 368 p.
19. *Tambe M.* Towards Flexible Teamwork // Journal of artificial intelligence research. 1997. vol. 7. pp. 83–124.
20. *Sycara K., Sukthankar G.* Literature review of teamwork models // Robotics Institute, Carnegie Mellon University. 2006. 31 p. URL: [http://www.ri.cmu.edu/pub\\_files/pub4/sycara\\_katia\\_2006\\_1/sycara\\_katia\\_2006\\_1.pdf](http://www.ri.cmu.edu/pub_files/pub4/sycara_katia_2006_1/sycara_katia_2006_1.pdf) (дата обращения: 04.11.2017).
21. ACL – Agent Communication Language. URL: <http://fipa.org/specs/fipa00061/SC00061G.pdf> (дата обращения: 04.11.2017).
22. RoboCup Soccer Server URL: <http://swarm.cs.virginia.edu/robocup/documentation/manual.pdf> (дата обращения: 04.11.2017).

23. *Виттик В.А., Скобелев П.О.* Мультиагентные модели взаимодействия для построения сетей потребностей и возможностей в открытых системах // Автоматика и Телемеханика. 2003. № 1. С. 177–185.
24. Cougaar Agent Architecture. URL: <https://en.wikipedia.org/wiki/Cougaar> (дата обращения: 04.11.2017).
25. Technology Reports, Universal Description, Discovery, and Integration (UDDI). URL: <http://xml.coverpages.org/uddi.html> (дата обращения: 04.11.2017).
26. *Городецкий В.И., Бухвалов О.Л., Скобелев П.О. Майоров И.* Современное состояние и перспективы промышленных применений многоагентных систем // Управление большими системами. 2017. Вып. 66. С. 94–157.
27. *Sierra C., Sabater J., Augusti J., Garcia P.* SADDE: Social Agents Design Driven by Equations // Methodologies and software engineering for agent systems. Kluwer Academic Publishers. 2004. pp. 1–24.
28. *Luck M., Gomez-Sanz J.* Agent-oriented Software Engineering IX // 9th International Workshop (AOSE 2008). 2009. LNCS 5386. 287 p.
29. *Garcia-Ojeda J.C., DeLoach S.A.* AgentTool Process Editor: Supporting the Design of Tailored Agent-based Processes // Proceedings of the 2009 ACM symposium on Applied Computing. 2009. pp. 707–714.
30. *Nwana H.S., Ndumu D.T., Lee L.C., Collis J.C.* ZEUS: A Toolkit for Building Distributed Multi-Agent Systems // Applied Artificial Intelligence. 1999. vol. 13. no. 1-2. pp. 129–185.
31. AgentBuilde – an integrated software toolkit that allows software developers to quickly develop intelligent software agents and agent-based applications. URL: <http://www.agentbuilder.com/Documentation/brochures/ABFamilyBrochure.pdf> (дата обращения: 04.11.2017).
32. *Burrafato P., Cossentino M.* Designing a multi-agent solution for a bookstore with PASSI methodology // Procs. Agent-Oriented Information Systems. 2002. pp. 119–135.
33. *Gorodetsky V., Karsaev O., Samoylov V., Konushy V.* Support for Analysis, Design and Implementation Stages with MASDK // International Workshop on Agent-Oriented Software Engineering. 2008. pp. 272–287.
34. *Rimassa G., Greenwood D., Kernland M.E.* The Living Systems Technology Suite: An Autonomous Middleware for Autonomic Computing // International Conference on Autonomic and Autonomous Systems (ICAS'06). 2006. pp. 33–33.
35. *Gorodetsky V.* Internet of Agents: From Set of Autonomous Agents to Network Object // Second International Workshop on Internet of Agents. 2017. pp. 1–17.
36. International Federation of Automatic Control (IFAC). URL: <https://www.ifac-control.org/> (дата обращения: 04.11.2017).
37. *Granichin O., Volkovich Z., Toledano-Kitai D.* Randomized Algorithms in Automatic Control and Data Mining // Springer Series: Intelligent Systems Reference Library. 251 p. URL: <http://www.twirpx.com/file/1562616/> (дата обращения: 04.11.2017).
38. *Mayorov I., Skobelev P.* Toward Thermodynamics of Real Time Scheduling // International Journal of Design & Nature and Ecodynamics. 2015. vol. 10. no. 3 pp. 21–223.
39. *Городецкий В.И.* Многоагентная самоорганизация в B2B сетях // XII Всероссийское совещание по проблемам управления ВСПУ-2014. С. 8954–8965. URL: <http://vspu2014.ipu.ru/proceedings/vspu2014.zip> (дата обращения: 04.11.2017).
40. *Rzhevski G., Skobelev P.* Managing complexity // WIT Press. 2014. 156 p.
41. *Скобелев П.О.* Мультиагентные технологии в промышленных приложениях: к 20-летию основания Самарской научной школы мультиагентных систем // Мехатроника, автоматизация, управление. 2010. № 12. С. 33–46.
42. *Скобелев П.О.* Интеллектуальные системы управления ресурсами в реальном времени: принципы разработки, опыт промышленных внедрений и перспективы

- развития // Приложение к теоретическому и прикладному научно-техническому журналу «Информационные технологии». 2013. № 1. С. 1–32.
43. *Городецкий В.И.* Самоорганизация и многоагентные системы. II. Приложения и технология разработки // Известия РАН. Теория и системы управления. 2012. № 3. С. 102–123.
  44. *Bonomi F., Milito R., Zhu J., Addepalli S.* Fog computing and its role in the internet of things // Proceedings of the first edition of the MCC workshop on Mobile cloud computing. 2012. pp. 13–16.
  45. *Yi Sh. et al.* Security and Privacy Issues of Fog Computing: A Survey // International Conference on Wireless Algorithms, Systems, and Applications. 2015. LNCS 9204. pp. 685–695.
  46. *Бухвалов О.Л. и др.* Распределенная координация в В2В производственных сетях // Известия Южного федерального университета. Технические науки. 2013. № 3. С. 193–203.
  47. *Городецкий В.И., Карсаев О.В., Самойлов В.В., Серебряков С.В.* Инструментальные средства для открытых сетей агентов // Известия РАН. Теория и Системы Управления. Москва. Наука. 2008. №. 3. С. 106–124.
  48. *Hewitt C., Bishop P., Steiger R.* A Universal Modular Actor Formalism for Artificial Intelligence // Proceedings of International Joint Conference on Artificial Intelligence (IJCAI-1973). pp. 235–245.
  49. *Bellifemine F., Poggi A., Rimassa G.* JADE – A FIPA-compliant agent framework. JADE. URL: <http://jmvidal.cse.sc.edu/library/jade.pdf> (дата обращения: 04.11.2017).

**Городецкий Владимир Иванович** — д-р техн. наук, профессор, главный научный сотрудник лаборатории интеллектуальных систем, Федеральное государственное бюджетное учреждение науки Санкт-Петербургский институт информатики и автоматизации Российской академии наук (СПИИРАН). Область научных интересов: прикладная статистика, планирование и составление расписаний, искусственный интеллект и принятие решений, многоагентные системы, инструментальные системы, интеллектуальная обработка данных и машинное обучение, системы с неопределенностью, рекомендующие системы, транспортная и производственная логистика, модели коллективного поведения роботов. Число научных публикаций — 300. [gor@iias.spb.su](mailto:gor@iias.spb.su), <http://space.iias.spb.su/ai/gorodetsky/>; 39, 14 линия В.О., Санкт-Петербург, 199178; р.т.: +7(812)323-3570, Факс: +7(812)328-0685.

**Скобелев Петр Олегович** — д-р техн. наук, ведущий научный сотрудник лаборатории интеллектуальных систем, Институт проблем управления сложными системами Российской академии наук (ИПУСС РАН), заведующий кафедрой электронных систем Самарский технический университет (СамГТУ), профессор, Самарский аэрокосмический университет, президент, Группа компаний «Генезис знаний», генеральный конструктор, Группа компаний «Генезис знаний». Область научных интересов: базы знаний и мультиагентные технологии, искусственный интеллект и принятие решений, распределение, планирование и оптимизация ресурсов в транспортной и производственной логистике, понимание текстов и извлечение знаний. Число научных публикаций — 250. [skobelev@kg.ru](mailto:skobelev@kg.ru), <http://www.kg.ru>; Московское шоссе, 17, офис 1202, Самара, 443017; р.т.: +7(846)279-37-79, Факс: +7(846)279-37-79.

**Поддержка исследований.** Результаты этой работы были получены в рамках исследований по бюджетной тематике СПИИРАН (проект № 0073-2015-0003) и по проекту Программы Президиума РАН 1.5, проект № 214, а также по гранту РФФИ «Теоретические основы эмерджентного интеллекта для решения сложных задач управления ресурсами» (проект № 16-01-00759)

V.I. GORODETSKY, P.O. SKOBELEV  
**INDUSTRIAL APPLICATIONS OF MULTI-AGENT  
 TECHNOLOGY: REALITY AND PERSPECTIVES**

*V.I. Gorodetsky, P.O. Skobelev. Industrial Applications of Multi-agent Technology: Reality and Perspectives.*

**Abstract.** Since early 1990<sup>th</sup>, multi-agent technology is evaluated as one of the most perspective design and implementation technologies for industrial scale distributed applications. However, the practice has falsified all the prognoses and expectations. The paper examines the current state-of-the-art in industrial use of the multi-agent technology. It analyzes external and internal reasons preventing broad practical use of the technology and formulates the lessons learnt through this examination. Finally, the paper outlines the basic issues to be revised in order to practically realize the great potential of the multi-agent technology. The paper also shows, by example, that multi-agent technology has currently no alternative for many novel most important applications including Internet of Things.

**Keywords:** multi-agent systems, industrial applications, multi-agent system roadmap, agent interactions, interaction protocol, Internet of Things.

**Gorodetsky Vladimir Ivanovich** — Ph.D., Dr. Sci., professor, chief researcher of intelligent systems laboratory, St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS). Research interests: applied statistics, planning, artificial intelligence, multi-agent systems, knowledge discovery and data fusion, P2P agent-based systems and technologies, self-organized systems, distributed data mining and knowledge discovery. The number of publications — 300. gor@iias.spb.su, <http://space.iias.spb.su/ai/gorodetsky/>; 39, 14-th Line V.O., St. Petersburg, 199178, Russia; office phone: +7(812)323-3570, Fax: +7(812)328-0685.

**Skobelev Petr Olegovich** — Ph.D., Dr. Sci., leading researcher of intelligent systems laboratory, Institute of Control of Complex Systems of Russian Academy of Science, head of electronic systems department, Samara State Technical University, professor, Samara State Aerospace University, president, Knowledge Genesis Group, chief developer, Knowledge Genesis Group. Research interests: knowledge bases and multi-agent technology, artificial intelligence and decision making, resource allocation, scheduling, optimization and control, text understanding and data mining. The number of publications — 250. skobelev@kg.ru, <http://www.kg.ru>; 17, Moscovskoe Shosse str., office 1202, Samara, 443013, Russia; office phone: +7(846)279-37-79, Fax: +7(846)279-37-79.

**Acknowledgements.** A part of the results were got as the results of the efforts according to the SPIIRAS Governmental Project № 0073-2015-0003 and the Project # 214 of the Program # 1.5 of the Russian Academy of Sciences, and RFFI grant "Theoretical Foundations of Emergent Intelligence for Solving Complex Problems of Management resources ", project #16-01-00759

## References

1. Digital Ecosystems. Available at: [https://en.wikipedia.org/wiki/Digital\\_ecosystem](https://en.wikipedia.org/wiki/Digital_ecosystem) (accessed: 04.11.2017).
2. Digital Business Ecosystems & The Platform Economy. Available at: <http://www.gartner.com/technology/topics/business-ecosystems.jsp> (accessed: 04.11.2017).
3. <https://www.linkedin.com/pulse/industrial-upcycling-definition-michael-rada/> (accessed: 04.11.2017).

4. Kaminka G. Robots Are Agents, Too! Keynote Lecture. International Conference on Autonomous Agents and Multi-agent Systems (AAMAS 2007). 2007. pp. 4.
5. The First International Workshop on the Internet of Agents (IoA). 2016. URL: <http://www2.cs.siu.edu/~salqithami/iaa/> (accessed: 04.11.2017).
6. Luck M., McBurney P., Shehory O., Willmott S. Agent technology: computing as Interaction (a roadmap for agent based computing). University of Southampton Department of Electronics & Computer Science. 2005. 108 p. Available at: <http://www.agentlink.org/roadmap/> (accessed: 04.11.2017).
7. Müller J., Fisher K. Application Impact of Multi-agent Systems and Technologies: A Survey. Agent-oriented software engineering. 2014. pp. 27–53.
8. Strassner J. Using agents and autonomic computing to build next generation seamless mobility services. Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems. 2007. Article no. 1.
9. DeLoach S.A. Moving multi-agent systems from research to practice. *International Journal Agent-Oriented Software Engineering*. 2009. vol. 3. no. 4. pp. 378–382.
10. Zambonelli F., Jennings N., Wooldridge M. Developing Multi-agent systems: The GAIA methodology. *ACM Transactions on Software Engineering and Methodology*. 2003. vol. 12(3). pp. 417–470.
11. Brinkkemper J., Solvberg A. Tropos: A framework for requirements-driven software development. *Info. Syst. Engg.: State Art Res. Themes*. 2000. LNCS 1. pp. 261–273.
12. DeLoach S. Analysis and Design using MaSE and agentTool. Proceedings of the 12<sup>th</sup> Midwest Artificial Intelligence and Cognitive Science Conference (MAICS). Miami University Press. 2001.
13. Bernon C., Gleizes M.P., Peyruqueou S., Picard G. ADELFE: a methodology for adaptive multi-agent systems engineering. International Workshop on Engineering Societies in the Agents World. 2002. pp. 156–169.
14. Caire G. et al. Agent Oriented Analysis using MESSAGE/UML. International Workshop on Agent-Oriented Software Engineering. 2001. pp. 119–135.
15. Padgham L., Winikoff M. Prometheus: A Paradigm Methodology for Engineering Intelligent Agents. Proceedings of the Workshop on Agent-Oriented Methodologies. 2002. pp. 97–108.
16. Leitao P., Vrba P. Recent Developments and Future Trends of Industrial Agents. Holonic and Multi-Agent Systems for Manufacturing. 2011. LNCS 6867. pp. 15–28.
17. Gorodetsky V.I., Samoilov V.V. Trotskii D.V. [The Reference Ontology of Collective Behavior of Autonomous Agents and Its Extensions]. *Izvestija RAN: Teorija i sistemy upravlenija – Proceedings of the Russian Academy of Sciences: Theory and Control Systems*. 2015. vol. 5. pp. 102–121. (In Russ.).
18. Wooldridge M. An Introduction to Multi-Agent Systems. John Wiley & Sons. 2009. 368 p.
19. Tambe M. Towards Flexible Teamwork. *Journal of artificial intelligence research*. 1997. vol. 7. pp. 83–124.
20. Sycara K., Sukthankar G. Literature review of teamwork models. Robotics Institute, Carnegie Mellon University. 2006. 31 p. Available at: [http://www.ri.cmu.edu/pub\\_files/pub4/sycara\\_katia\\_2006\\_1/sycara\\_katia\\_2006\\_1.pdf](http://www.ri.cmu.edu/pub_files/pub4/sycara_katia_2006_1/sycara_katia_2006_1.pdf) (accessed: 04.11.2017).
21. ACL – Agent Communication Language. Available at: <http://fipa.org/specs/fipa00061/SC00061G.pdf> (accessed: 04.11.2017).
22. RoboCup Soccer Server Available at: <http://swarm.cs.virginia.edu/robocup/documentation/manual.pdf> (accessed: 04.11.2017).
23. Vittikh V.A., Skobelev P.O. [Multi-agent Interaction Models for Constructing the Needs-and-Means Networks in Open Systems]. *Avtomatika i Telemekhanika – Automation and Remote Control*. 2003. vol. 1. pp. 177–185. (In Russ.).

24. Cougaar Agent Architecture. Available at: <https://en.wikipedia.org/wiki/Cougaar> (accessed: 04.11.2017).
25. Technology Reports, Universal Description, Discovery, and Integration (UDDI). Available at: <http://xml.coverpages.org/uddi.html> (accessed: 04.11.2017).
26. Gorodeckij V.I., Buhvalov O.L., Skobelev P.O. Majorov I. Multi-agent Systems Industrial Applications: Current State and Prospects. *Upravlenie Bol'shimi sistemami – Large-Scale Systems Control*. 2017. vol. 66. pp. 94–157. (In Russ.).
27. Sierra C., Sabater J., Augusti J., Garcia P. SADDE: Social Agents Design Driven by Equations. Methodologies and software engineering for agent systems. Kluwer Academic Publishers. 2004. pp. 1–24.
28. Luck M., Gomez-Sanz J. Agent-oriented Software Engineering IX. 9th International Workshop (AOSE 2008). 2009. LNCS 5386. 287 p.
29. Garcia-Ojeda J.C., DeLoach S.A. AgentTool Process Editor: Supporting the Design of Tailored Agent-based Processes. Proceedings of the 2009 ACM symposium on Applied Computing. 2009. pp. 707–714.
30. Nwana H.S., Ndumu D.T., Lee L.C., Collis J.C. ZEUS: A Toolkit for Building Distributed Multi-Agent Systems. *Applied Artificial Intelligence*. 1999. vol. 13. no. 1-2. pp. 129–185.
31. AgentBuilder – an integrated software toolkit that allows software developers to quickly develop intelligent software agents and agent-based applications. Available at: <http://www.agentbuilder.com/Documentation/brochures/ABFamilyBrochure.pdf> (accessed: 04.11.2017).
32. Burrafato P., Cossentino M. Designing a multi-agent solution for a bookstore with PASSI methodology. *Procs. Agent-Oriented Information Systems*. 2002. pp. 119–135.
33. Gorodetsky V., Karsaev O., Samoylov V., Konushy V. Support for Analysis, Design and Implementation Stages with MASDK. International Workshop on Agent-Oriented Software Engineering. 2008. pp. 272–287.
34. Rimassa G., Greenwood D., Kernland M.E. The Living Systems Technology Suite: An Autonomous Middleware for Autonomic Computing. International Conference on Autonomic and Autonomous Systems (ICAS'06). 2006. pp. 33–33.
35. Gorodetsky V. Internet of Agents: From Set of Autonomous Agents to Network Object. Second International Workshop on Internet of Agents. 2017. pp. 1–17.
36. International Federation of Automatic Control (IFAC). Available at: <https://www.ifac-control.org/> (accessed: 04.11.2017).
37. Granichin O., Volkovich Z., Toledano-Kitai D. Randomized Algorithms in Automatic Control and Data Mining. Springer Series: Intelligent Systems Reference Library. 251 p. Available at: <http://www.twirpx.com/file/1562616/> (accessed: 04.11.2017).
38. Mayorov I., Skobelev P. Toward Thermodynamics of Real Time Scheduling. *International Journal of Design & Nature and Ecodynamics*. 2015. vol. 10. no. 3 pp. 21–223.
39. Gorodetsky V.I. [Multi-agent self-organization in B2B Networks]. *XII Vserossiyskoe soveshchanie po problemam upravleniya VSPU-2014* [Proceedings of XII Russian Symposium on Control Problems]. pp. 8954–8965. Available at: <http://vspu2014.ipu.ru/proceedings/vspu2014.zip> (accessed: 04.11.2017). (In Russ.).
40. Rzhnevski G., Skobelev P. Managing complexity. WIT Press. 2014. 156 p.
41. Skobelev P.O. [Industrial applications of Multi-agent technologies: To 20th anniversary of Samara School on Multi-agent Systems]. *Mekhatronika, Avtomatizatsiya, Upravlenie – Mechatronics, Automation, Control*. 2010. vol. 12. pp. 33–46. (In Russ.).
42. Skobelev P.O. [Real-time Intelligent Resource Management System: Principles of Development, Practical Experience and Perspectives] *Prilozheniye k Teoreticheskomu i Prikladnomu Zhurnalu "Informatsionnye Tekhnologii" – Annex to the Theoretical and Applied Journal "Information Technologies"*. 2013. vol. 1. pp. 1–32. (In Russ.).
43. Gorodetskii V.I. [Self-Organization and Multi-agent Systems: II. Applications and the Development Technology]. *Izvestiya RAN: Teoriya i sistemy upravleniya – Proceedings*
44. Труды СПИИРАН. 2017. Вып. 6(55). ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн) [www.proceedings.spiiras.nw.ru](http://www.proceedings.spiiras.nw.ru)

- of the Russian Academy of Sciences: Theory and Control Systems*. 2012. vol. 3. pp. 102–123. (In Russ.).
44. Bonomi F., Milito R., Zhu J., Addepalli S. Fog computing and its role in the internet of things. Proceedings of the first edition of the MCC workshop on Mobile cloud computing. 2012. pp. 13–16.
  45. Yi Sh. et al. Security and Privacy Issues of Fog Computing: A Survey. International Conference on Wireless Algorithms, Systems, and Applications. 2015. LNCS 9204. pp. 685–695.
  46. Bukhvalov O., Gorodetsky V., Karsaev O., Koudriatsev G., Samoylov V. [Distributed Coordination in B2B Production Networks]. *Trudy Yuzhnogo Federalnogo Universiteta (Tekhnicheskie Nauki) – Transactions of South Federal University (Technical Sciences)*. 2013. vol. 3. pp. 193–203. (In Russ.).
  47. Gorodetsky V, Karsaev O., Samoylov V., Serebrykov S. Development tools for open agent networks. *Izvestija RAN: Teorija i sistemy upravlenija – Proceedings of the Russian Academy of Sciences: Theory and Control Systems*. 2008. vol. 3. pp. 106–124. (In Russ.).
  48. Hewitt C., Bishop P., Steiger R. A Universal Modular Actor Formalism for Artificial Intelligence. Proceedings of International Joint Conference on Artificial Intelligence (IJCAI-1973). pp. 235–245.
  49. Bellifemine F., Poggi A., Rimassa G. JADE – A FIPA-compliant agent framework. JADE. Available at: <http://jmvidal.cse.sc.edu/library/jade.pdf> (accessed: 04.11.2017).