

УДК 519.687.1/4

ББК 32.988-5

## **СОВРЕМЕННОЕ СОСТОЯНИЕ И ПЕРСПЕКТИВЫ ИНДУСТРИАЛЬНЫХ ПРИМЕНЕНИЙ МНОГОАГЕНТНЫХ СИСТЕМ**

**Городецкий В. И.<sup>1</sup>, Бухвалов О. Л.<sup>2</sup>**

*(Санкт-Петербургский институт информатики  
и автоматизации РАН, Санкт-Петербург)*

**Скобелев П. О.<sup>3</sup>,**

*(Институт проблем управления сложными системами  
РАН, Самарский аэрокосмический университет, Самара)*

**Майоров И. В.<sup>4</sup>**

*(Научно-производственная компания  
«Разумные решения», Самарский государственный  
технический университет, Самара)*

*Рассматриваются основные тенденции и перспективы развития индустриальных приложений многоагентной технологии, анализируются недавние прогнозы и реальное состояние ее практического применения. Анализируются внешние и внутренние причины, препятствующие широкому промышленному внедрению многоагентных систем и технологий, а также анализируются уроки, которые следует извлечь из этого анализа. Описываются свойства и типы современных и будущих приложений, в реализации которых многоагентные технологии имеют неоспоримые преимущества. Показывается, что многоагентным системам и технологиям в настоящее время нет альтернативы при управлении крупномасштабными объекта-*

---

<sup>1</sup> Владимир Иванович Городецкий, доктор технических наук, профессор (gor@iias.spb.su).

<sup>2</sup> Олег Леонидович Бухвалов (psyhoveter@gmail.com).

<sup>3</sup> Петр Олегович Скобелев, доктор технических наук, профессор (petr.skobelev@gmail.com).

<sup>4</sup> Игорь Владимирович Майоров (imayorov@smartsolutions-123.ru).

*ми сетевой структуры, построенными на принципах самоорганизации.*

Ключевые слова: многоагентные системы, промышленные приложения, самоорганизация, объекты сетевой структуры.

## **1. Введение**

В обзоре Gartner, вышедшем в свет в октябре 2015 года, многоагентные системы (МАС) и технологии (МАС-технологии) включены в список наиболее перспективных информационных технологий (ИТ) следующего десятилетия [26]. Такой точки зрения придерживаются и многие специалисты в области информационных технологий. Однако несколько неожиданным является тот факт, что ИТ-индустрия не спешит с использованием многоагентной технологии, хотя последняя и в среде специалистов числится в списке перспективных уже давно. Естественно встает вопрос о том, почему многочисленные и уверенные предсказания относительно хороших перспектив технологии МАС пока не оправдываются и когда можно ожидать ее успехов на промышленном уровне.

В настоящей работе дается краткое описание истории развития этого направления, анализируются проблемы, которые существенно тормозят внедрение МАС в практику, и что необходимо сделать, по мнению авторов, для полноценного выхода МАС-технологий на рынок промышленных внедрений.

Концепция МАС была впервые предложена в середине 1980-х годов. Она сразу была высоко оценена как научным, так и промышленным сообществами. В исследования и разработки в области МАС и технологий в 1990-е годы были вовлечены достаточно большие научные силы ведущих университетов и ИТ-компаний мира. В результате 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 разработано несколько хорошо продуманных методологий создания МАС, началась разработка инструментальных программных средств их поддержки. В 1996 г. была создана общественная организа-

ция 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. С самого начала она позиционировалась как парадигма создания сложных систем, построенная на биологических принципах (англ. *bio-inspired* – вдохновляемая биологией, живыми системами), которая 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 разбиение сложной задачи с множеством взаимодействующих сущностей на относительно простые законченные подзадачи, понятные специалисту. Решение этих задач 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, т.е. 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 разнообразным. При этом агенты могут генерировать события и посылать сообщения другим агентам, вырабатывать и согласовывать варианты решений, передавать входные и выходные данные, оценивать результаты решения своих подзадач, формировать задания для других агентов, поддерживать синхронизацию коллективных действий, передавать сигналы обратной связи и т.п. Эта концепция представляется естественной для приложений, в которых участвует много раз-

ных 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 и т.п. То же самое относится и к производственной логистике, аналогичными объектами которой являются (на нижнем уровне) заказы, отдельные технологические производственные операции, рабочие и станки, выполняющие эти операции, материалы и т.д.

Особенно привлекательными для МАС-технологий были и остаются до настоящего времени задачи индивидуальной и коллективной робототехники. Специалистам в области робототехники эта концепция с самого начала представлялась идеальной для моделирования коллективного поведения автономных роботов в различных миссиях [33]. Можно указать и ряд других классов приложений, для которых МАС-парадигма вплоть до настоящего времени воспринимается как наилучший, а иногда и просто единственно возможный вариант концептуализации, моделирования и программной реализации. Это прежде всего касается приложений, управляющих сложными крупноразмерными объектами сетевой структуры, а приложений подобного рода на практике становится с каждым днем все больше не только в транспорте и производстве, но и в энергетике, здравоохранении, военном деле и многих других приложениях. Наступающая эра Интернета вещей, в приложениях которой центральным аспектом является именно взаимодействие распределенных автономных сущностей (как «вещей» и их удаленных пользователей, так и «вещей» между собой для более сложных запросов) порождает новый широкий класс приложений, для которых МАС-технология представляется приоритетной технологией.

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 их взаимодействия. Важно от-

метить, 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*) [35]. 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.

Убедительным 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 *RoadMap: «Agent Technology: Computing as Interaction»* [35]. Этот документ фактически подвел краткие итоги двадцатилетнего развития парадигмы, модели и технологии МАС, дал оценку практических перспектив МАС и, что самое важное, дал предельно оптимистический прогноз перспектив индустриальных 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. Например, в программе Европейской комиссии FP5 исследования, посвященные непосредственно развитию теории и инструментальных средств технологической поддержки процессов разработки МАС почти не финансировались. В [38] по этому поводу отмечается, что с середины 2000-х годов публичное восприятие работ в области МАС стало менее значимым. Финансирование проектов по программам FP5 и FP7 Европейской комиссии по направлению ICT (англ. *Information and Communication Technologies*) было сфокусировано на других направлениях, таких как сервис-ориентированные вычисления, ГРИД-вычисления, автономные вычисления и др. Удачные разработки (англ. *success stories*) этого 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. В программах FP5 и FP7 поддерживались отдельные проекты, в которых присутствовали компоненты, реализованные с использованием МАС-

архитектур и технологий, но проекты в целом не ассоциировались с исследованиями в интересах развития теории и практики агентов или МАС. На конференции ААМАС'2007 в приглашенном докладе вице-президента и директора автономной лаборатории компании Моторола Дж. Стресснера (John Strassner) одной из ключевых тем был анализ состояния индустриальных разработок в области многоагентных приложений. Автор этого доклада с удивлением говорил, что он обнаружил не более шести МАС-приложений, разработанных за 20-летнюю историю МАС, которые, хотя и приближенно, но могли бы квалифицироваться как индустриальные [45].

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 [24] 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) в программировании, который появился в конце 1960-х годов (Simula в 1967) и в начале 90-х годов стал общепринятым. В [24] выделяется также ряд существенных недоработок в области теории и технологии МАС, которые, по его мнению, требуют уточнения. Они обсуждаются далее в разделе 4 данной работы. Автор [24] считает, что разработчики МАС в первую очередь должны продемонстрировать способность МАС-концепции и технологи создавать сложные адаптивные и самоорганизующиеся распределенные системы промышленного уровня, и именно это для МАС остается главным исследовательским вопросом до сих пор.

В 2013 году была опубликована заметная работа [38], в которой авторы тщательно проанализировали реальное состояние прикладных разработок в области МАС на тот момент. Общая идея этой весьма своевременной, а, возможно, даже несколько запоздавшей работы, сформулирована авторами уже в ее первых строках: «*В то время как имеются убедительные свидетельства важности МАС и технологий как исследовательской об-*

*ласти, остается неясным, какой практический эффект от нее имеется к настоящему времени.»<sup>1</sup>*

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 работе [35]. Во-вторых, в ней имеется интересная статистика, которая позволяет более глубоко проанализировать реальное состояние МАС-разработок различного уровня зрелости на это совсем недавнее время, а также выявить некоторые причины неудовлетворительного состояния таких разработок.

Обобщая сказанное выше о современном состоянии и ближайших перспективах в области индустриальных применений МАС и технологий, можно утверждать, что в этой области в настоящее время имеются негативные тенденции, тормозящие использование больших потенциальных возможностей МАС. По этой причине представляется важным выявить и проанализировать эти тенденции, а также оценить реальные перспективы МАС как информационной технологии индустриального уровня в ближайшем будущем.

Этот анализ и составляет главную цель данной работы.

Далее в разделе 1 даются краткие сведения о ключевых свойствах МАС и технологий, а также уточняются задачи данной работы. В разделе 2 анализируется состояние и оцениваются перспективы прикладных разработок в области МАС на 2005 г. [35] с прогнозом их динамики на период (2005–2015+) гг. В разделе 3 на основании фактического материала работ [34, 38] и других источников информации дается характеристика современного состояния разработок в области прикладных МАС различного уровня зрелости и приводится сравнение достигнутых результатов с их прогнозом на 2015+ гг., приведенным в [35]. В разделе 4 анализируются негативные тенденции в области теории и практики МАС, а также причины, которые, по мнению

---

<sup>1</sup> *"While there is ample evidence that Multiagent Systems and Technologies are vigorous as a research area, it is unclear what practical application impact this research area has accomplished to date."*

авторов данной работы, привели к определенному кризису в области индустриальных приложений МАС и технологий. В разделе 5 намечаются возможные варианты преодоления этого кризиса, которые могли бы способствовать скорейшему и более эффективному использованию практически неограниченного потенциала МАС и как парадигмы концептуализации, и как методологии разработки, и как технологии программирования самых сложных распределенных интеллектуальных систем. В этом же разделе приводятся также примеры удачных разработок МАС-приложений, выполненных под руководством и при участии авторов данной работы и формулируются особенности использованных моделей и технологий, которые сделали возможным практическую реализацию потенциала многоагентных систем и технологий. В заключении резюмируются основные результаты работы.

## **2. Многоагентные системы как парадигма вычислений на основе взаимодействий**

*Агентом принято называть автономную компьютерную программу (систему), которая способна к целенаправленному поведению в динамической, непредсказуемо изменяющейся внешней среде. В этом определении агента в качестве его ключевых свойств выделяются автономность и целенаправленность поведения. Автономность агента понимается как его способность функционировать в интересах достижения поставленной цели без вмешательства человека или других систем и при этом осуществлять самоконтроль над своими действиями и внутренним состоянием.*

*Многоагентная система определяется как сеть слабо связанных решателей частных проблем (агентов), которые существуют в общей среде и взаимодействуют между собой для достижения тех или иных целей системы. Взаимодействие может осуществляться агентами либо прямым образом – путем обмена сообщениями, либо некоторым косвенным образом, когда одни агенты воспринимают присутствие других агентов через изменения во внешней среде, с которой они взаимодействуют. МАС может содержать несколько однотипных или разнотипных аген-*



тов, которые могут иметь общие и/или различные цели, могут быть распределенными по компьютерной сети, могут быть написаны на различных языках программирования и работать на различных операционных платформах. Существуют различные взгляды на то, какими свойствами должны обладать агенты и существуют различные их классификации. Однако для целей данной работы вполне достаточно введенных описаний агентов и МАС. Отметим, что именно так сформулированы понятия агентской программы и МАС в базовом документе [35].

Взаимодействие рассматривается в МАС как основной способ вычислений и координации поведения множества автономных программных или физических агентов. В соответствии с существующими стандартами агенты взаимодействуют между собой на языке высокого уровня, используя протоколы. Взаимодействие агентов, а значит, и их результирующее совместное поведение, может иметь различные цели. Агенты МАС могут взаимодействовать с целью *кооперативного решения* некоторой общей сложной или крупномасштабной задачи. В этом случае задача разбивается, например, пользователем на более простые относительно автономные задачи, которые поручаются разным агентам. В таком варианте взаимодействие агентов имеет целью координацию локальных решений для достижения некоторого требуемого качества решения задачи в целом. Эта координация может достигаться либо в полностью распределенном варианте, либо с помощью управления с той или иной степенью централизации, реализуемого агентом, специально выделенным для этих целей. Качество решения исходной большой задачи обычно оценивается с помощью некоторой глобальной функции полезности (англ. *global utility function*), значение которой зависит от локальных решений агентов.

Другой характер взаимодействия агентов реализуется в случае, когда каждый агент имеет свои цели, однако он по каким-либо причинам не в состоянии решить задачу самостоятельно, а потому вынужден прибегать к помощи других агентов. Это взаимодействие агентов тоже имеет целью кооперацию, однако, в отличие от предыдущего случая, в этой ситуации агенту в кооперации может быть *отказано*, что определяется дополнительными соглашениями между агентами, которые принято

называть *взаимными обязательствами* агентов (англ. *commitments*).

Если взаимные обязательства агентов относительно слабы и они «помогают» друг другу не в ущерб собственным интересам, то такое объединение агентов в МАС называется *альянсом*. Если агенты объединяются в группы с достаточно *сильными* взаимными обязательствами, как правило, с тем чтобы помочь друг другу выстоять в конкурентной борьбе с другими агентами и/или их группами, то такое объединение агентов принято называть *коалицией*. Агенты коалиции всегда имеют четко оговоренные условия, определяющие, в каких условиях и каким образом они помогают друг другу. Условия, при которых агенты коалиции прекращают помогать друг другу, также обычно четко оговариваются, и эти условия называются *соглашениями* (англ. *conventions*). Если группа агентов решает общую задачу и при этом она действует как один агент, то такое объединение агентов называется *командой*, и в случае такого объединения локальные цели агентов, формируемые динамически, всегда должны быть направлены на достижение общей цели команды.

Еще один тип взаимодействия агентов имеет место тогда, когда агенты не кооперируются, а, наоборот, *конкурируют* друг с другом. В этом случае каждый агент имеет собственную цель и является, как принято говорить, *самоинтересованным* (от англ. *self-interested*), или эгоистичным. Примеры таких моделей МАС дает электронная коммерция, процессы создания и функционирования виртуальных предприятий и др. При определенных условиях агенты могут как конкурировать, так и кооперироваться друг с другом, при этом они могут легко переходить от конкуренции к кооперации, и наоборот, что называется отношением *коопетиции* (coopetition – от англ. *cooperation* + *competition*).

Например, сотрудники виртуальной организации могут конкурировать за мелкие заказы, но сразу же объединяться, если появляется сложный заказ, который не может быть выполнен ни одним из них поодиночке.

МАС как одно из направлений в современных информационных технологиях формирует *область исследований и разработок*, в которой, с прагматической точки зрения, принято раз-

личать три основных направления с общей теоретической основой (рис. 1) [35]:

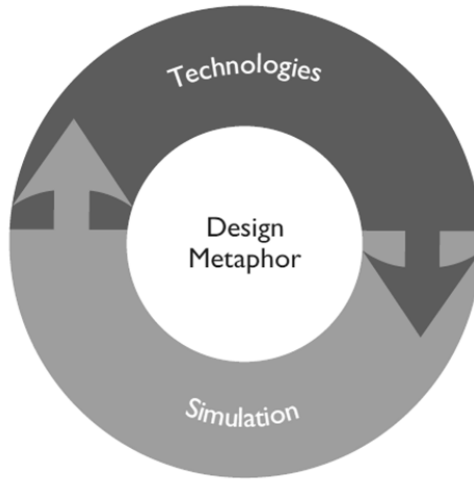


Рис. 1. Прагматика использования понятий агентов и МАС  
(рисунок взят из общедоступного ресурса [35])

- агенты и МАС как *метафора концептуального проектирования* (с акцентом на модульность);
- агенты и МАС как *источник технологий* (с акцентом на взаимодействия как принцип вычислений и принятия согласованных решений);
- агенты и МАС как *средство имитационного моделирования* (с акцентом на автономное поведение компонент, взаимодействующих на основе протоколов).

Что касается теоретической основы этих трех направлений, то к настоящему времени предложены различные теории агентов и МАС, и каждая из них обладает своими выразительными возможностями, определяет сложность концептуальной модели агента и МАС. Обычно каждая теория агентов требует разработки специализированной технологии и инструментов для поддержки процессов проектирования приложений и их программирования. К числу наиболее распространенных моделей аген-

тов и МАС относятся реактивные модели, логические модели, *BDI*-модели (от англ. *Belief-Desire-Intension*), поведенческие, био-инспирированные, самоорганизующиеся и другие модели. Естественно, что каждая из них имеет свои достоинства и недостатки, а самое главное – свою область приложений. Как хорошо известно, адекватный выбор модели по отношению к конкретному классу приложений является основным фактором, определяющим потенциальную успешность конкретной прикладной разработки<sup>1</sup>. В разделе 4, посвященном анализу причин негативных тенденций в области индустриальных применений агентов и МАС, этот аспект обсуждается более предметно и детально.

### **3. 2005 год: состояние прикладных разработок в области МАС и прогноз на 10+ лет**

Предваряя анализ прикладных разработок в области МАС и технологий, авторы [35] пишут, что исследования и разработки в этой области к 2005 г. еще не достигли уровня зрелости, необходимого для их использования в индустриальных разработках (возраст примерно 20 лет на 2005 г.). В качестве аргумента они сравнивают возраст МАС-технологий и ООП, которое приобрело практическую значимость более чем через 30 лет. Например, ООП-язык *C++* был создан через 32 года, а *JAVA* – через 39 лет после первых работ по ООП. Авторы также сетуют на слабость методологий для проектирования МАС, разработанных к тому времени, хотя с этим трудно согласиться.

Действительно, к этому времени было создано и уже достаточно длительно тестировалось несколько глубоко проработанных методологий МАС, например, *Gaia* [52], *Tropos* [37], *MaSE* [22], *ADELFE* [16], *MESSAGE* [19], *Prometheus* [39] и ряд других.

---

<sup>1</sup> Это естественное и как будто совершенно тривиальное утверждение; тем не менее, как показывает многолетний опыт, оно постоянно нуждается в напоминании.

Другое дело, что на то время они еще не были поддержаны адекватными инструментальными средствами. Заметим, что отсутствие таких средств было не случайным: оно было следствием ряда принципиальных ошибок (об этом позже), допущенных в области теории агентов и МАС, о которых документ [35] ничего не говорит. Обратим внимание на то, что перечисленные методологии относятся к числу самых ранних разработок в этой области (все они были созданы до 2003 г.). Следует заметить, что активность разработок в области методологий разработки агентов и МАС, а также инструментальных средств для поддержки их проектирования и программной реализации была достаточно высокой до 2005 года и в последующие несколько лет, 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 демонстрирует свою жизнеспособность и в настоящее время (в списке выше они не указаны), но о них речь пойдет позже и особо.

В таблице 1 приведены данные динамики прогноза разработок промышленных образцов МАС на период до 2015+ г. по различным прикладным областям [35]. Из данных этой таблицы видно, что в качестве наиболее перспективных областей использования МАС в то время рассматривались телекоммуникационные системы и сети, производственные системы, транспорт, здравоохранение, финансовая сфера, программное обеспечение и т.п. Общее число МАС промышленного уровня к концу периода прогнозировалось в количестве около 250 экземпляров.

Однако, хотя в период после 2005 г. активность исследований и разработок в области МАС-технологий и средств их инструментальной поддержки не снижалась, практика показала, что этот прогноз оказался слишком оптимистичным. Реальные достижения в этой области оказались намного скромнее.

Таблица 1. Прогноз количества индустриальных разработок прикладных МАС на период до 2015+ г.

	Наименование области приложения	Прогноз числа разработанных приложений по годам		
		2010	2015	2015+
1	Телекоммуникационные системы и сети	17	28	35
2	Производственные системы	15	27	35
3	Транспортная логистика	15	25	32
4	Здравоохранение	12	25	34
5	Электричество, газ, вода	10	17	25
6	Программное обеспечение	8	17	24
7	Финансы, страхование, недвижимость	10	17	24
8	Административные системы	6	12	21
9	Аппаратное обеспечение	3	9	12
10	Строительство	1	6	11
...	...	...	...	...

#### 4. Прогноз и реальность на 2015+

В литературе можно найти сведения о нескольких сотнях МАС-приложений, разработанных в период 2005-2015 гг. в различных прикладных областях. Анализ состояния исследований и разработок в области индустриальных и других применений МАС и технологий с акцентом на тенденции в этой области проведен в ряде работ, среди которых наиболее заметными являются обзоры [38] и [34]. Далее используются в основном сведения и мнения, приведенные в первом из этих обзоров.

Работа [38] анализирует 152 различных приложения, сведения о которых были либо предоставлены авторами разработок, либо почерпнуты из научной литературы. Важно отметить, что более половины проанализированных разработок в последующем, к сожалению, не были подтверждены авторами, так что их практическое использование или продолжение их разработок остается под вопросом [38].

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. Сведения о некоторых из них приведены в таблицах 2, 3 и 4, соответственно [34, 38].

Анализ уровня зрелости этих и других разработок этого периода показал, что прогноз 2005 г. оказался излишне оптимистичным, причем в нескольких аспектах. В нем, прежде всего, сильно переоценено общее число созданных промышленных образцов МАС, прогнозируемых к 2015 г.

По-иному распределились эти разработки и в зависимости от областей приложений, что, однако, неудивительно, поскольку после 2005 г. появились новые классы актуальных приложений, например, мобильные приложения и мобильные сервисы. Но, наверное, самым неожиданным оказалось то, что темпы появления новых разработок МАС-приложений стали постепенно замедляться. Этот факт отражает реальное падение интереса индустриальных компаний к МАС-технологии. Практика показала, что многие приложения, для которых МАС рассматривались как наиболее перспективная технология программной реализации [35], к 2013 г. были успешно реализованы с помощью других технологий. Среди них наиболее конкурентоспособными оказались сервис-ориентированные технологии, ГРИД-вычисления, автономные, повсеместные, облачные, туманные вычисления и др. Следует заметить, что эти технологии появились значительно позже, однако смогли быстро потеснить МАС-технологии.

Таблица 2. Примеры МАС для управления производством

Назначение/ Заказчик	Разработчик	Предметная область	Уровень зрелости
Production 2000+	Daimler Chrysler, Schneider	Управление производством	Промышленный образец
Car body painting	Daimler-Benz	Управление производством	Программный прототип
ВНР Billiton	Rockwell	Управление технологическим процессом	Промышленный образец
Chilled Water System	Rockwell	Распределенное управление	Программный прототип
Cambridge packing cell	U. Cambridge	Управление производством	Лабораторный стенд
FABMAS	Technical U. of Ilmenau	Управление производством	Промышленный образец
PS-Bikes	Universita de Genova	Управление производством	Программный прототип
Axion-Holding	СПИИРАН-НПК «Разумные решения»	Планирование производства	Промышленный образец
Shop Modelarna Liaz	Certicon, Gerstner Laboratory	Планирование производства	Промышленный образец
SkodaAuto	Gedas, Certicon, Gerstner Lab.	Планирование производства	Промышленный образец
Agent Steel System	Saarstahl AG, DFKI Gmbh	Планирование производства	Промышленный образец
SDM Laboratory	Yokogawa	Управление оборудованием	Лабораторный стенд
NovaFlex	Uninova	Управление производством	Лабораторный стенд
ADACOR	Polytechnic Institute of Braganca	Управление производством	Лабораторный стенд
ABAS	Tampere U. of Technology, Schneider Electric	Управление производством	Лабораторный стенд
OntoReA	TU Wien, Rockwell Automation, COPA-DATA	Управление производством	Лабораторный стенд



Таблица 3. Примеры МАС разного уровня зрелости, созданные для управления логистикой

Назначение / Заказчик	Исполнитель	Предметная область	Применение
Air Liquide America	NuTech	Оптимизация логистики	Промышленный образец
Tankers International	Magenta	Планирование логистики	Промышленный образец
Airport ground service operations	Airbus, EADS, Cologne University of Applied Sciences, Группа компаний «Генезис знаний»	Управление работой наземных служб аэропорта, служб питания на борту, расписанием полетов авиакомпании и др.	Программный прототип и лабораторный стенд
Taxi scheduling / Addison Lee	Magenta	Планирование в реальном времени	Промышленный образец
Rent-a-car scheduling and optimization / Avis	Magenta	Планирование и оптимизация в реальном времени	Промышленный образец
Transportation orders consolidation, routing and scheduling / GIST	Magenta	Консолидация, маршрутизация и планирование паллет в реальном времени	Промышленный образец

*Управление большими системами. Выпуск 66*

*Таблица 3. Примеры МАС разного уровня зрелости, созданные для управления логистикой (продолжение)*

Trucks scheduling/Prologics	НПК «Разумные решения»	Планирование грузоперевозок с (в реальном времени)	Промышленный образец
Southwest Airlines	BiosGroup	Оптимизация работы наземных служб	Промышленный образец
ABX Logistics	Whitestein	Транспортная логистика в реальном времени	Промышленный образец
MAST	Rockwell	Динамическая маршрутизация товаров	Моделирование, Лабораторный стенд
MAS-RFiD	U. of Castilla-La Mancha	Управление логистикой	Моделирование
MASDIMA	TAP, LIACC	Адаптация работы авиалинии	Лабораторный стенд

*Таблица 4. Примеры МАС для аэрокосмической отрасли и энергетики*

<b>Назначение/ клиент</b>	<b>Разработчики</b>	<b>Предметная область</b>	<b>Приме- нение</b>
NASA airspace satellites	NASA	Управление запросами спутников	Промышленный образец
РКК «Энергия»	НПК «Разумные решения»	Динамическое перепланирование полетов и грузопотока МКС, поддержка принятия решений по управлению сменами ГОГУ, нештатным и аварийным ситуациям, программами научных экспериментов	Промышленные образцы
Aerogility	LostWax	Интеллектуальная поддержка принятия решений	Моделирование
Turkey energy forecast	KKK Per. Bsk.	Прогнозирование потребности в энергии	Моделирование
California Energy Commission	AESC, Acronymics	Координирование и планирование	Промышленный образец
Large urban area	Rockwell	Обработка воды	Моделирование

Работа [38], в которой впервые явно прозвучали тревожные ноты относительно перспектив МАС-технологий в конкуренции с другими современными интеллектуальными информационными технологиями, появилась очень своевременно. Она, хотя и в излишне мягкой форме, отражала реальное отношение индустриального сообщества к перспективам МАС-технологий, и это отношение было в 2013 г. и остается сейчас явно не в пользу МАС. Заметим, что работа [38] отражает позицию специалиста в

области МАС, который понимает, что к этому времени потенциал МАС пока не используется в полной мере. Но если бы она отражала позицию индустриальных разработчиков, то критика перспектив МАС-технологий была бы, скорее всего, гораздо более жесткой.

Излишняя мягкость оценок перспектив МАС-технологий в работе [38] обусловлена еще и тем, что основными источниками информации о разработанных и развернутых МАС-приложениях были сами авторы этих разработок. Таких было 103 из 202 приложений, данные о которых рассматривались в качестве исходного материала для анализа в [38], хотя понятно, что в реальности авторы, в силу субъективизма, всегда склонны несколько переоценивать качество и уровень собственных разработок.

Кроме того, нельзя не отметить, что ряд научных разработок в литературе был заявлен авторами как полноценные промышленные системы, хотя на деле они представлял собой всего лишь первые пилоты или исследовательские прототипы.

Обратимся к результатам анализа прикладных разработок в области МАС, приведенным в [38]. После предварительного изучения исходного множества МАС-приложений, информация о которых была предоставлена разработчиками или найдена авторами работы [38] в научной литературе, для анализа было отобрано 152 разработки. Эти разработки анализировались с различных точек зрения. Рассмотрим некоторые из них.

1. *Зрелость разработки.* По этому свойству все разработки 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, но не были использованы в реальной работе (55 из 152), а также
- *лабораторные исследовательские прототипы (пилотные проекты)*, которые использовались в учебных исследовательских, и других аналогичных целях (46 из 152).

Про остальные 5 разработок достоверная информация у авторов [38] отсутствовала.

2. *Тип коллектива разработчиков.* По этому свойству все разработки разделены также на 3 группы:

- *индустриальные* компании (43 из 152, из них 26 разработок отвечают промышленному уровню зрелости);
- *университетское сообщество* (37 из 152, из них только 4 разработанные системы отвечают промышленному уровню зрелости) и
- *смешанные коллективы*, когда в разработках принимали участие одновременно сотрудники обоих сообществ, названных выше (58 из 152, из которых 16 разработок соответствуют промышленному уровню зрелости).

Про остальные 14 разработок достоверная информация у авторов [38] отсутствует.

3. *Тип агентского приложения.* По этому свойству выделено 3 типа разработок:

- *МАС* (таких 125 из 152);
- *автономные агенты* (14 из 152);
- *интерфейсные агенты* (11 из 152).

Для двух разработок тип агентского приложения авторам [38] неизвестен.

Результаты анализа, приведенные в [38], показывают, что активность использования МАС-технологий в реальной жизни оказалась намного ниже прогноза, данного в [35]. Действительно, вместо ожидаемых примерно 200–250 приложений промышленного уровня, к 2015 г. в реальности было разработано всего лишь 46, и при этом по некоторым из них информация о том, что они практически используются после 2013 г., не была подтверждена публикациями. Индустриальное сообщество формирует меньше трети общего интереса к МАС-приложениям (43 из 152 приложений), а основная активность в этой части формируется научным сообществом, т.е. специалистами в области многоагентных систем, *субъективно* заинтересованными в расширении области применимости МАС-технологий.

Важным показателем тенденций этого периода является активное использование в это время методологий и инструментальных средств разработки МАС-приложений. Примерно в 72% случаев авторы разработок использовали ту или иную методологию и/или программный инструментарий. Это косвенно

отражает тот факт, что МАС-приложения в этот период разрабатывались главным образом для тестирования методологий и инструментальных программных систем, и, скорее всего, не были инициированы потребностями индустриальных разработчиков. Это подтверждается тем фактом, что в большинстве случаев разработкой методологии и разработкой приложений с использованием соответствующих методологий занимались одни и те же научные коллективы. Это как раз отражает активность исследований и разработок в области методологий МАС и инструментальных программных систем в период 2005–2015 гг., что уже отмечалось в начале данного раздела.

Что касается предпочтительных языков программирования, то в 82% случаев в качестве базового языка программирования использовались такие стандартные языки как *Java* (55%), *C/C++/C#* (15,6%), *PHP* (7,3%) и *Python* (4,1%).

Среди стран, которые вели и ведут наиболее активные исследования и практические разработки промышленных МАС, лидерами являются США (12 приложений промышленного уровня из 46), Великобритания (6), Испания (5), Германия (4), Италия (4), Чехия (3), Австралия (2), Россия (2), Швеция (2), Швейцария (2), Франция (1), Голландия (1), остальные страны (2). Важно, однако, заметить, что разработчики из США, Австралии и России, на которых приходится треть всех успешных индустриальных разработок МАС (они находятся в активной эксплуатации вплоть до настоящего времени), не используют модели, методологии и программные инструменты, воспринимаемые мировым научным сообществом как наиболее передовые и перспективные. Они используют самые простые модели, например, реактивные и поведенческие, не используют стандартный достаточно сложный *ACL*-язык коммуникаций агентов, принятый FIPA [14], не используют стандартную FIPA-архитектуру агентской платформы [25], а также опираются на собственные механизмы принятия решений агентами.

Интересно отметить, что прогноз по типам приложений, которые, по мнению авторов документа [35], будут представлять наибольший интерес для разработчиков приложений, в целом оправдался. Как и предсказывалось, лидером в области использования МАС-приложений индустриального уровня остаются

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 [35] предсказывались гораздо лучшие перспективы.

Основные выводы, которые можно сделать из материалов данного раздела, состоят в следующем:

1. МАС-технологии развиваются гораздо сложнее, медленнее и труднее, чем это хотелось бы научному сообществу и 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. Поэтому одинаково конкурентными оказываются как технологии широкого применения, например, МАС, нейросети (англ. *neuronets*), повсеместные вычисления (англ. *ubiquitous computing*), туманные вычисления (англ. *fog computing*) интеллектуальное окружение (англ. *ambient intelligence*) и т.п., так и весьма частные технологии типа нечеткой логики (англ. *fuzzy logic*), генетических алгоритмов (англ. *genetic algorithms*), муравьиных алгоритмов (*ant colony*) и т.п.

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. К ним относятся Motorola, Siemens и др. компании. Некоторые компании, например, IBM, Daimler, NASA, Google активно используют 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 [38]. Вместе с тем, на рын-

ке появляются новые промышленные компании (Rockwell, DHL и др.), которые пытаются активно использовать МАС-технологии.

3. Интерес индустриального сообщества к использованию МАС и МАС-технологий к настоящему времени значительно снизился. 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 или не давая им быстро захватить новые сегменты.

Очевидно, что в настоящее время МАС и технологии во многом еще пока 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 агентов и МАС. Поэтому представляется важным выяснить причины, которые в настоящее время тормозят практическое использование огромного потенциала, которыми МАС и МАС-технологии, бесспорно, обладают. Это вопрос рассматривается в следующем разделе.

### **5. 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, которые ими решаются. 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 многоагентным системам, а также тематику докладов на ведущих конференциях по МАС, то можно видеть, что подавляющая часть их посвящена концептуальным моделям приложений и архитектур их про-



граммной реализации. Следует заметить, что эта тенденция наблюдается и в настоящее время, хотя и не так очевидно, как это было примерно до 2010 г. На этом этапе разработки приложений МАС предлагают понятный и привлекательный вариант проектирования и здесь все смотрится весьма перспективно.

Однако уже на следующем этапе, когда на основании концептуальной модели нужно построить формальную модель агента, построить архитектуру программно-коммуникационной среды, с помощью которой взаимодействуют агенты, а также описать язык общения агентов, все оказывается, мягко говоря, несколько сложнее, и причины этого нельзя назвать объективными. Сложность шага формализации модели и архитектуры МАС многократно возрастает. Решения, которые были предложены<sup>1</sup> для этого этапа специалистами в области формальных моделей, практически отторгаются теми, кто заинтересован в их практическом использовании.

Рассмотрим этот и другие аспекты теории МАС аспект несколько подробнее.

### *5.1. ОТСУТСТВИЕ ОБЩЕПРИНЯТОГО ПОНИМАНИЯ КЛЮЧЕВЫХ ПОНЯТИЙ МАС*

На это указано еще в работе [24]. 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 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 сходны-

---

<sup>1</sup> Возможно, правильнее было бы сказать, что эти решения были навязаны прикладникам.

ми концепциями ООП, которые также используются в области программирования агентов и МАС.

### *5.2. ОТСУТСТВИЕ ОБЩЕПРИНЯТОЙ НОТАЦИИ ДЛЯ ПРЕДСТАВЛЕНИЯ МОДЕЛЕЙ И КОМПОНЕНТ МАС*

Поскольку еще не сформулировано даже содержание общепринятых определений агентских понятий и их взаимосвязей [24], то отсутствие общей нотации для их описания и описания отношений на их множестве не позволяет исследовать различные модели МАС на практике. Например, после перевода в общую нотацию, удалось найти много схожих свойств в методологиях *O-MasE* [23] и *Prometheus* [39]. Стандартизация моделей и способов их представления очень желательны, поскольку при отсутствии согласованных концепций существует опасность неадекватных оценок новых многообещающих подходов в области МАС-технологий.

### *5.3. КОНЦЕПТУАЛЬНАЯ И ВЫЧИСЛИТЕЛЬНАЯ СЛОЖНОСТЬ ЛОГИЧЕСКОЙ ФОРМАЛИЗАЦИИ VDI-МОДЕЛИ АГЕНТА И МАС*

Почти два десятилетия основные усилия исследователей в области теории МАС были направлены на разработку моделей *интеллектуальных* агентов. Уже в самом начале этих исследований был сформулирован ряд свойств, которыми должен обладать агент. В этом контексте 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 пропагандировалась в течение многих лет. Можно сказать, что специалисты в области описания формальных моделей «соревновались» в том чтобы обеспечить агента все новыми и новыми интеллектуальными возможностями в части автономного поведения вплоть до способности определять намерения других агентов. Естественно, что простыми средствами столь мощные интеллектуальные возможности агента описать и реализовать невозможно, что приводило к постоянному усложнению формальной модели агента и МАС.

С самого начала развития теории агентов и МАС в качестве базовой формальной модели интеллектуального агента была выбрана *BDI*-модель (*BDI* от англ. *Belief–Desire–Intention*, *Убеждение–Желание–Намерение*) [50], в которой знания, убеждения, намерения и механизмы рассуждений описываются в терминах исчисления предикатов, расширенного модальными и темпоральными операторами.

Большинство специалистов и в настоящее время придерживаются понятий *BDI*-модели агента в МАС и их логической формализации. Однако эта модель, с одной стороны, определенно сложна для понимания и является существенным барьером в интерпретации базовых понятий МАС [24], включая понимание самого термина *BDI*. По мнению автора работы [24], необходимо сосредоточиться на интерпретации тех понятий МАС, которые акцентируют внимание не на способах формального представления модели агента и МАС, а на понятиях, существенных для описания агентов и МАС как средств концептуализации и технологии разработки распределенных систем, использующих принципы самоорганизации. Такими понятиями для *BDI*-архитектуры являются, например, поведенческие и мотивационные свойства агентов.

Важно отметить, что сама по себе *концептуальная основа BDI-модели* достаточно естественна и убедительна. Суть ее состоит в том, что эта концепция оперирует двумя типами понятий: ментальными и поведенческими. К *ментальным* понятиям относятся *события, убеждения, цели и взаимные убеждения*. *События* отражают (представляют формально) то, что происходит в реальном времени с агентами или объектами внешнего мира. Они имеют в качестве атрибутов *тип события* и *имя агента*, генерирующего событие, или агента, с которым событие ассоциировано. *Убеждения* – это знания агента о *внешнем мире*, а именно утверждения, в истинности которых агент убежден в текущий момент времени. Но они, во-первых, могут изменяться во времени и, во-вторых, могут быть неверными с самого начала или стать таковыми в некоторый момент времени. В этом их отличие от знаний. *Цели* – это некоторые состояния агента, которые он стремится достичь. Эти состояния называются *целевыми состояниями агентов*. Поскольку агенты могут обмени-

ваться знаниями и часто должны координировать свое поведение, то важным понятием рассматриваемой модели является понятие взаимных убеждений агентов. *Взаимные убеждения* агентов складываются из убеждений группы агентов. Поскольку разные агенты могут иметь различные убеждения, то взаимные их убеждения – это та часть индивидуальных убеждений, в которых сходятся все агенты группы.

К поведенческим понятиям *BDI*-модели агента относятся *индивидуальные обязательства* и *соглашения* агента, его *индивидуальные намерения*, а также *общие* (коллективные) *обязательства* и *намерения* команды агентов. *Индивидуальные обязательства* агента – это цели (состояния, подцели дерева целей), которые он должен достигнуть в соответствии теми задачами из общего множества задач агентов, которые агент взял на себя. *Индивидуальное намерение* (англ. *intention*) агента – это обычно конкретное *действие*, которое он собирается выполнить для достижения индивидуальной цели. *Соглашениями* называются условия, при выполнении которых агент может отказаться от своих индивидуальных обязательств. В зависимости от типа кооперации (альянс, коалиция или команда), эти соглашения могут быть более жесткими или менее жесткими. Например, если *MAC* является командой агентов, то эти соглашения определяются однозначно и являются достаточно жесткими [20]. Перечисленные понятия могут быть положены в основу верхнего (предметно-независимого) уровня онтологии *MAC*. Один из примеров онтологии поведенческих понятий *BDI*-модели агента и *MAC* можно найти в работе [8].

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<sup>1</sup>. Можно легко увидеть большую аналогию этой истории с тем,

---

<sup>1</sup> Одним из примеров является печально знаменитый японский проект вычислительных машин пятого поколения, который опирался на логический язык *Пролог* и который полностью провалился уже в 1980-е годы.

что происходило и, к сожалению, все еще происходит в настоящее время в области теории агентов и МАС.

Действительно, были попытки использовать логическую формализацию понятий, представляющих *BDI*-концепцию агента и МАС, т.е. описать ментальные понятия агента и модель их распределенного взаимодействия в терминах исчисления предикатов первого порядка (ИП-1). Однако достаточно быстро было обнаружено, что выразительные возможности ИП-1 недостаточны для описания моделей ментальных понятий агентов. Приведем простой пример.

Пусть выражение  $[q/p]\varphi$  означает, что формула  $[q/p]\varphi$  получена из формулы  $\varphi$  путем подстановки в нее подформулы  $q$  вместо подформулы  $p$  (в каком-то числе случаев ее вхождения в формулу  $\varphi$ ), и при этом подформулы  $p$  и  $q$  имеют одинаковые истинностные значения во всех интерпретациях. В классической логике всегда будет справедливо

$$|= (p \leftrightarrow q) \rightarrow \varphi \leftrightarrow [q/p]\varphi,$$

где символ  $|=$  трактуется как отношение, сохраняющее истинностное значение. Это свойство принято называть экстенциональностью<sup>1</sup>. Содержательно оно означает, что истинность формулы  $\varphi$  не изменится при использовании любого количества подстановок подформулы  $q$  вместо подформулы  $p$ , если обе они имеют одно и то же истинностное значение. Однако это свойство делает невозможным адекватное описание свойств ментальных понятий и их взаимосвязей. Покажем это на простом примере из области футбола роботов [49].

Пусть в убеждениях агента истинны такие факты:

1.  $p$ : «мяч находится у своего игрока»,
2.  $s$ : «мячом владеет своя команда» и
3.  $q$ : «у меня мяча нет».

Очевидно, что истинной является также формула «с поскольку  $r$ », где бинарный предикат «поскольку» является одним

---

<sup>1</sup> Термин происходит от английского слова *extension*, который переводится как расширение. Здесь он применяется в смысле расширяемости логики за счет использования подстановок указываемого далее типа.

из конструктивов, который может использоваться в схемах рассуждений с ментальными понятиями. Если в последней формуле использовать подстановку истинной подформулы  $q$  вместо истинной подформулы  $p$ , то в результате получится формула «с поскольку  $q$ » («мячом владеет своя команда» поскольку «у меня мяча нет»), про истинность которой уже ничего определенного утверждать нельзя. Таким образом, предикат «поскольку» не является экстенциональным, а потому не может быть представлен в языке классической логики. Таких примеров можно привести много [50]. Заметим, что описываемая проблема имеет семантический характер.

Слабые выразительные возможности ИП-1 были преодолены за счет его обогащения модальными и темпоральными операторами. А далее все получилось по аналогии с продвижением логической модели в области искусственного интеллекта. Аналогия эта состоит в активном продвижении логической модели, обогащенной модальными и темпоральными операторами в качестве базовой модели агента и МАС. Именно эта модель и получила название *BDI*-модели.

Но эта модель даже теоретически намного сложнее ИП-1, и вряд ли можно надеяться на ее практически приемлемую эффективность. Заметим, что в модели *BDI*-агента взаимодействие агентов предполагается, в основном, на уровне простых диалогов. Это означает, что агенты должны быть настолько «интеллектуальными», чтобы самостоятельно строить модели других агентов и модель внешней среды для принятия решений, что на практике оказалось невозможным. Кроме того, такой подход означает отказ от сути многоагентной парадигмы, которая формулируется как вычисления на основе взаимодействий.

В итоге результат 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 практически реализуемых моделей и технологий в области МАС. Более того, это затормозило практическое использование МАС технологий на десятилетие, как

минимум. Действительно, 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, главным образом проблемы вычислительной сложности. Даже для простых *MAC*-приложений модели знаний агентов получаются чрезвычайно громоздкими, а механизмы рассуждений, использующие вывод в логических исчислениях с модальными и темпоральными операторами, – совершенно неподъемными. Ярким 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 много-агентных систем как самые значительные (англ. *influential*) достижения [46, 47]. Эти работы достаточно детально проанализированы в [10]. Однако на практике эти результаты демонстрировались авторами на очень простых примерах. В начале 2000-х гг. оба эти проекта были закрыты. Можно сказать, что хорошую модель коллективного поведения агентов в автономной миссии погубила логическая модель *BDI*-агента с использованием модальных и темпоральных операторов.

Однако в это время, хотя и на вторых ролях, развивались и другие модели агентов (о них речь пойдет позже). И именно эти модели, главным образом, были использованы в большей части тех приложений, которые рассматриваются как успешные в [34], [38] и в ряде других обзоров по практическому использованию *MAC* и соответствующих технологий. Два таких примера приводятся в конце раздела б.

#### 5.4. СТАНДАРТЫ FIPA (FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS)

Поскольку парадигма и технология МАС выросли из научного направления, которое называлось распределенный искусственный интеллект и которое имело целью решение практических задач, то вопрос о стандартах стал темой исследований еще в середине 1990-х годов, когда была создана общественная организация FIPA. В задачи этой организации входило научное обоснование стандартов в области МАС-технологий. Но поскольку FIPA была создана тем же научным сообществом, которое продвигало в практику логическую модель *BDI*-агента и МАС, то эта же модель рассматривалась FIPA в качестве базовой модели МАС и при разработке стандартов.

Например, язык коммуникации агентов *ACL* (англ. *Agent Communication Language*) [14] в этом стандарте использует весьма сложный язык для описания содержания сообщений, которыми обмениваются агенты. Предшественником этого языка был язык *KQML* (*Knowledge Query and Manipulation Language*). Разработка этого языка началась 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* 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 явля-



ется язык обмена сообщениями, принятый в сервере RoboCup [41]. Он использует только необходимые и достаточные средства общения агентов и поэтому достаточно эффективен. Например, для двух команд симуляционного футбола по 11 агентов каждая оказывается достаточным обычного персонального компьютера выпуска 2004 года для того, 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» [5, 13], 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* [21], который разработан в США по проекту DARPA для его использования в военных приложениях. В нем обмен сообщениями поддерживается архитектурой доски объявлений, на которой агенты предлагают свои сервисы (доступные другим агентам как подключаемые модули) и ищут необходимые им сервисы по мере необходимости. При этом доска объявлений значительно расширяет множество доступных сервисов за счет доступности веб-сервисов в соответствии с *UDDI*-протоколом [48]. Эта возможность реализуется специальной компонентой доски объявлений, которая называется *сервлет*. Она поддерживает коммуникации с объектами Интернет по *http*-протоколу. Этот сервис доступен для использования всеми подключаемыми модулями узла. Отметим, что сервлет может возвращать данные в форматах *HTML*, *XML*, серии *Java*-объектов и бинарных данных.

Важно подчеркнуть, что модели агентов и MAC, предложенные в инструментах компании *Magenta* и *Cougaar*, оказались наиболее успешными в части индустриальных разработок. Можно также заметить, что все перечисленные инструменты 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 MAS – это концепция, изначально ориентированная на параллельные вычисления.

### ***5.5. ОТСУТСТВИЕ ГИБКИХ ПРОМЫШЛЕННЫХ МЕТОДОВ И ТЕХНОЛОГИЙ ДЛЯ РАЗРАБОТКИ MAS-ПРИЛОЖЕНИЙ***

В работе [24] 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 и трудными 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.

Одной из причин современного кризиса в области MAS и технологий является ошибочная стратегия в области методологий и средств разработки MAS. С самого начала исследователи в области MAS придавали большое значение созданию методологий проектирования и программной реализации MAS, причем в период 2005–2010 гг. эти исследования и разработки велись наиболее активно. В этот период общее мнение состояло в том, что хорошая методология разработки и мощное средство инструментальной поддержки ее программной реализации позволят в значительной степени автоматизировать процесс создания прикладных MAS промышленного уровня, так что их создание можно будет поставить на поток.

В период до 2010 г. было разработано более десяти методологий проектирования и разработки MAS, если не считать большого количества других менее значимых разработок в области методологий, которые привнесли мало нового. К числу наиболее перспективных и наиболее обоснованных методологий

следует отнести такие методологии как *Gaia* [52], *Tropos* [37], *MaSE* [22], *ADELFE* [16], *MESSAGE* [19], *Prometheus* [39], *SADDE* [42] и ряд других. Большинство разработок в области методологий MAC сопровождалось также созданием инструментальных программных средств их поддержки (см., например, [36]). Эти средства использовали либо авторскую методологию разработки, либо одну из списка приведенного выше, которые в сообществе специалистов считались передовыми. Например, к этому времени были разработаны такие средства как *agentTool* [27], *Zeus* [31], *agentBuilder* [15], *PASSI* [18], *MASDK* [28] 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 MAC, 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 была дополняться компонентами, для которых код может быть написан только вручную (это касалось «не агентских» компонент). Полученный высокоуровневый код должен был после этого компилироваться в исполняемый код.

Нетрудно видеть, что инструментальная поддержка подобной методологии проектирования и программной реализации МАС-приложений оказывалась достаточно «тяжелой» и сложной, 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 эффективности методологии разработки и программной реализации приложения, так же, как и вычислительной эффективности целевого приложения, не удавалось. В итоге такой методологический и инструментальный подход, за некоторыми редкими исключениями, себя не оправдал. Только отдельные программные прототипы удавалось с его помощью довести до экспериментальных образцов.

Эта 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 (см. также таблицы 2–4 в разделе 3 данной работы). Но среди этих разработок большинство не получило последующего развития и применения на практике, и их разработка была прекращена.

Та же судьба к настоящему времени постигла и большинство методологий и поддерживающих их инструментальных программных средств: разработка и поддержка большей части из них в настоящее время прекращена, поскольку эти разработки не привели к прорыву в области индустриальных применений МАС и технологий.

Повторим, что основной причиной этого, наряду с рядом других причин, является неверный выбор базовой модели агента и МАС, который был поддержан «тяжелыми» стандартами *FIPA*.

### *5.6. ПОЗИЦИОНИРОВАНИЕ МАС И МАС-ТЕХНОЛОГИЙ В ПРИКЛАДНЫХ ОБЛАСТЯХ*

Как уже отмечалось ранее, существует много классов приложений, которые ранее позиционировались как приложения, для которых модель, архитектура и технология МАС являются

наиболее перспективными по сравнению с другими технологиями. 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 явно не так.

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 [35] базировалось в основном на поверхностных ассоциациях и потому было неубедительно. В этом документе, скорее, перечисляются те прикладные области, в которых агентские технологии могут быть использованы, причем наряду с другими областями. В результате более зрелые технологии оказались в ряде случаев более успешными, что способствовало утверждению негативных мнений о МАС-технологиях и снижению интереса индустриального сообщества относительно их возможностей.

Вместе с тем можно привести много различных прикладных задач в самых разных областях, в которых МАС-технологии являются единственным вариантом технологии разработки на множестве всех существующих технологий. К ним относятся задачи, которые решаются в распределенном варианте с применением методов распределенного решения задач на базе принципов самоорганизации и эволюции. В таких задачах классические модели, методы и средства оптимизации быстро «захлебываются» в переборе вариантов, а МАС-технологии оказываются способными строить хотя бы допустимые решения. И эту область приложений следует очертить заранее.

Примером такой задачи в производственной логистике может быть построение согласованных производственных планов и сменно-суточных заданий рабочих в десятках цехов крупного машиностроительного предприятия с миллионами сборочных операций в день. Аналогичной задачей является составление

железнодорожных расписаний на участках интенсивного движения сотен поездов. К такому же классу относятся задачи управления цепочками поставок для тысяч наименований товаров, движущихся с разной скоростью, и т.д. Примеры этих и других подобных промышленных приложений МАС с измеримыми результатами в части повышения эффективности работы предприятий подробно описаны в [1, 28]. Заметим, что формальные модели, используемые в них, свободны от большинства недостатков, перечисленных в данном разделе.

Можно утверждать, что для МАС-технологий пришла пора извлекать уроки и сузить возможный спектр применения от общесистемного применения в распределенных системах – к созданию действительно сложных адаптивных систем, реализуемых на основе принципов самоорганизации, для решения сложных, многокритериальных, плохо формализованных и вычислительно трудоемких задач в реальном времени, как, например, управление ресурсами, распознавание образов, извлечение знаний и т.д. [11, 12, 40, 43].

## **6. Перспективы развития индустриальных МАС-технологий**

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 промышленные применения, а этот процесс на практике никогда не бывает линейным (рис. 2).

В начале своего развития в 1990-е годы на стыке объектного программирования, параллельных вычислений, телекоммуникаций и искусственного интеллекта (вспомним синие экраны Norton Commander в DOS и первые интернет-модемы в то время) МАС стали вызовом в программировании и потребовали больших усилий выдающихся ученых и программистов для первых успешных разработок.

Несложно предсказать, что процесс развития любой технологии, тем более такой революционной по своей сути, как МАС-технология, будет и в дальнейшем весьма нелинейным. В этом процессе потребуется сделать еще много попыток его транс-

формации, успешных и безуспешных, прежде чем удастся сделать МАС-технологию продуктивной и коммерческой.



Рис. 2. Предыстория и перспективы развития МАС и МАС-технологий в контексте стандартного цикла развития новых технологий

Помимо рассмотренных выше концептуальных и алгоритмических проблем, разработчики МАС часто встречаются и с не менее значимыми специфическими технологическими проблемами, которые свойственны системам, использующим *локальные* механизмы самоорганизации и эволюционные вычисления. Заметим, что именно эти механизмы являются базовыми при поиске решений в крупномасштабных задачах с оптимизацией управления объектами сетевой структуры, т.е. в той прикладной области, которая является для МАС-технологий наиболее перспективной. Так, в работах [40, 43] отмечается, что при создании МАС, решающих задачи распределения, планирования, оптимизации и контроля использования ресурсов на множестве объектов, формирующих сеть, зачастую возникают специфические эффекты и проблемы, затрудняющие поиск решения, которые перечисляются ниже:

– В условиях постоянных модификаций управления динамическим объектом, работающим в реальном времени, бывает очень трудно оценить, насколько текущее решение далеко от динамически изменяющегося «оптимального» решения, так что алгоритм управления обычно пытается лишь «догнать» этот оптимум и всегда опаздывает.

– Решение зависит от предыстории, например, от порядка наступления тех или иных событий, что обычно обусловлено динамикой объекта управления и нелинейностью как самого управляемого объекта, так и нелинейностью управления («чувствительностью ко времени»).

– Зачастую появляется «эффект бабочки», при котором малые изменения на входе системы приводят к неожиданным для наблюдателя большим изменениям на выходе, что обычно связано с переходом самоорганизующейся системы из одного локального оптимума (устойчивого состояния) в другой.

– Реакция системы может непредвиденно замедляться для наблюдателя в случае возникновения длинной цепочки изменений, вносимых в управление объектом; это обычно бывает обусловлено большим числом вариантов и экспоненциальным характером роста сложности вычислений на локальном участке или неконтролируемым расширением волны переговоров в зависимости от длины цепочки агентов, затрагиваемых изменениями.

– При повторном запуске системы с теми же самыми входными данными решение может оказаться другим; это обычно бывает обусловлено недетерминизмом параллельных асинхронных процессов и малыми флуктуациями решений или наличием в алгоритмах работы системы случайных механизмов выбора в эвристиках, используемых в процессе поиска «наилучшего» решения, что приводит к тому, что разные компоненты программы в разных реализациях получают иные входные данные.

– В силу динамики объекта управления решение, полученное с помощью эволюционного подхода, невозможно «откатить» назад, поскольку ситуация, как правило, при такой попытке будет уже иной.

– При попытках доработки управления вручную, например, при попытках изменить ручным способом расписание использо-



вания ресурсов, случаются непредвиденные для пользователя существенные изменения расписания как в калейдоскопе и т.д. При возникновении каждой такой проблемы разработчикам на практике приходится искать собственные решения, во многом зависящие от предметной области.

Наконец, как отмечается в [5, 11, 12], разработчиков МАС подстерегают и серьезные проблемы организационного, коммерческого и тому подобного характера, обусловленные конкуренцией на рынке информационных технологий. В качестве примеров можно привести следующие:

- Разработки МАС в сфере управления часто вторгаются в самую «запретную» область принятия решений, критическую для бизнеса, где «все уже занято» (хотя в наличии обычно имеются только учетные системы) или процессы принятия решений совершенно не формализованы.

- Процесс поиска решений в МАС является распределенным, причем с акцентом на самоорганизацию, что на практике противоречит традициям иерархического управления и, что немало важно, традициям проектирования систем по принципу «сверху–вниз», поскольку самоорганизующиеся системы разрабатываются по принципу «снизу–вверх» [9].

- 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 т.д.

- 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 информационными системами заказчика.

– Примерное соотношение затрат труда (%) 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 т.д.

– Пользователи должны быть мотивированы на внедрение системы и в идеале премироваться как по результатам внедрения, так и по величине достигаемого повышения эффективности в работе, чтобы работать на одну цель с разработчиками.

– Пользователи должны иметь возможность вручную дорабатывать расписания, поскольку всегда остаются факторы, которые не представляется возможным учесть при автоматическом принятии решений в системе, и т.п.

Преодоление указанных проблем, безусловно, требует значительных усилий, но в случае успеха результаты окупают все вложения, обеспечивая на практике следующие преимущества МАС-технологий:

– Позволяют создавать интеллектуальные системы нового класса для решения сложных проблем и дают результаты, сопоставимые с результатами работы специалистов.

– Повышают эффективность использования ресурсов, качество обслуживания, снижают затраты денег и времени, риски и штрафы в экономике реального времени.

– Решают сложные задачи и предлагают рациональные, пригодные на практике решения за счет перехода от полного перебора - к поиску конфликтов и их разрешению путем достижения компромиссов.

– Поддерживают работу в реальном времени с быстрой и гибкой реакцией на события.

– Обеспечивают индивидуальный подход для каждого участника процесса использования МАС, включая заказчиков продукции и услуг, исполнителей, субподрядчиков и т.д.;

– Помогают снизить зависимость от персоналий в принятии решений.

- Снижают затраты на разработку за счет повторного использования кода при переходе к новым сферам применений и усложнении решения.

- Дают возможность моделирования типа «Если–то» для оптимизации и прогнозирования развития ситуации.

- Создают надежную и масштабируемую платформу для роста сложности решаемых задач и развития бизнеса без роста численности управленческого персонала.

Можно утверждать, что в настоящее время в области МАС идет активная работа над первыми ошибками и преодоление недостатков, что станет залогом новых достижений.

Вместе с тем, полноценный выход МАС на плато продуктивности на уровне отраслевых стандартов на наш взгляд можно ожидать не ранее 2020–2030 гг.

К числу наиболее перспективных областей использования МАС-технологий относятся такие *новые* прикладные области, как

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

- *В2В-сети производственных и транспортных предприятий*, стратегическое планирование и оперативное управление производством, сетевая логистика (транспортная и др.) – это одна из основных областей, где уже активно используются агентские решения и технологии и где все еще остается большое поле для использования сетевых МАС-технологий [7].

- *Военные приложения* – имеется много косвенных признаков и свидетельств активности индустриальных разработок в этой области за рубежом, однако по большей части эта информация закрытая.

- *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 перспективы МАС в этой области подтверждаются ростом активности соответствующих разработок.

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

– Распределенные системы *наблюдения и обеспечения безопасности*, сенсорные сети, интеллектуальные пространства, и туманные вычисления (англ. *fog computing*), например, в интересах задач Интернета вещей [17, 51].

– *Мобильные приложения* – это совсем новая область МАС-приложений, задачи которой по большей части только формулируются. Важно отметить, что по оценкам компании Gartner до 40% будущих мобильных приложений в течение ближайших 10 лет будут построены на основе многоагентных технологий.

Однако уже в настоящее время имеются МАС-приложения, которые демонстрируют свои преимущества на практике. Приведем два примера таких приложений, которые разработаны под руководством и при участии авторов данной работы. Они реализованы с помощью авторских технологий, которые во многом избегают тех недостатков, которые в разделе 4 отнесены к главным причинам того, что многоагентные технологии к настоящему времени пока еще не заняли среди других интеллектуальных технологий ведущего места.

### 6.1. САМООРГАНИЗУЮЩИЕСЯ В2В-СЕТИ ПРЕДПРИЯТИЙ

В2В-сетями называют вид экономического и информационного взаимодействия множества предприятий, имеющий целью координацию планирования и исполнения распределенных бизнес-процессов *в интересах самого бизнеса*. Это означает, что потребитель конечного продукта, производимого предприятиями сети, отсутствует в ней. 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, т.е. к ней могут присоединяться новые узлы, некоторые узлы могут выходить из сети навсегда или возвращаться в нее снова. Такая сеть является сетью парных взаимодействий (англ. *peer-to-peer*, р2р-сетью), в которой как коммуникации, так

и содержательные взаимодействия поддерживаются с помощью р2р-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 (предприятия перегружено заказами и может выделить на новый заказ лишь ограниченные ресурсы) или экономическими (распределенный процесс выполняется специализированными предприятиями и это обходится дешевле) и др. Более подробное описание существа В2В-сетей, основных особенностей организации их функционирования, а также алгоритмов распределенного планирования и составления расписаний исполнения взаимосвязанных заказов, поступающих в сеть, имеется в [43]. Далее рассматривается только одна из наиболее сложных и трудоемких задач, решаемых в В2В-сетях в режиме реального времени, а именно задача координации расписаний распределенного выполнения заказов.

Многоагентная архитектура узла В2В-сети представлена на рис. 3.

Алгоритм распределенной координации локальных расписаний узлов В2В-сети строится на основе обмена метаданными между ее узлами в процессе составления ими локальных расписаний. В составлении локальных расписаний узла принимают агенты трех типов:

- *Агент узла* или узел некоторого подразделения предприятия (англ. *Department Agent, DA*) ответственен за составления локальных расписаний узла и за взаимодействие с *DA*-агентами других узлов при решении задачи распределенной координации на метауровне.

- *Агент ресурса* участвует в выполнении *Протокола контрактных сетей* [44] при составлении локального расписания.

- *Локальный агент заказа* участвует в составлении локального расписания узла, отвечая за соблюдение технологии выполнения локальной части заказа.

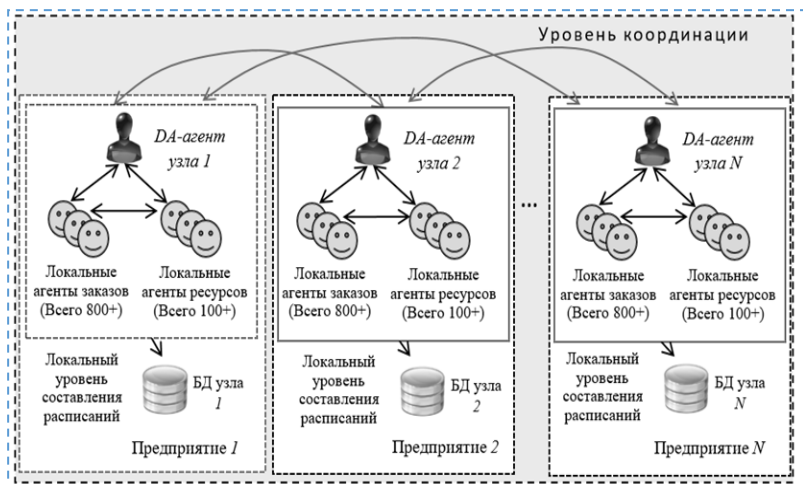


Рис. 3. Архитектура MAS распределенного составления расписаний выполнения взаимосвязанных заказов в производственной B2B-сети

Детали взаимодействия этих агентов при составлении локального расписания описаны в [2]. В системе имеется также Агент заказа (англ. *Order-agent*, *OA*), который является агентом мета-уровня. Он владеет информацией о заказе в целом, например, о самом раннем допустимом времени начала исполнения заказа и о самом позднем допустимом времени его завершения. Важно отметить, что в этой архитектуре информация о ресурсах и технологиях узлов доступна только агентам соответствующих узлов, что решает задачу конфиденциальности

Задача координации локальных расписаний решается на основе взаимодействия *DA*-агентов узлов, а также их взаимодействия с мета-агентами *OA*. Протокол из взаимодействия описан в [3, 6]. 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 *p2p*-протокола [6, 7, 29].

Однако архитектура и реализация этой платформы (среды исполнения агентов и среды их взаимодействия) значительно

отличается от той, которая диктуется стандартами FIPA. В принятой архитектуре (рис. 4) средой исполнения всех агентов является *Akka System*, которая играет роль локальной компоненты р2р-платформы агентов. Эта компонента реализует функции управления потоками, диспетчеризацию сообщений, управляет жизненным циклом акторов (агентов) [30], является каркасом для дополнительных модулей системы и отвечает за ряд других важных сервисных задач. Все узлы В2В-сети разделяют общую онтологию *Ontology*, т.е. одну модель данных.

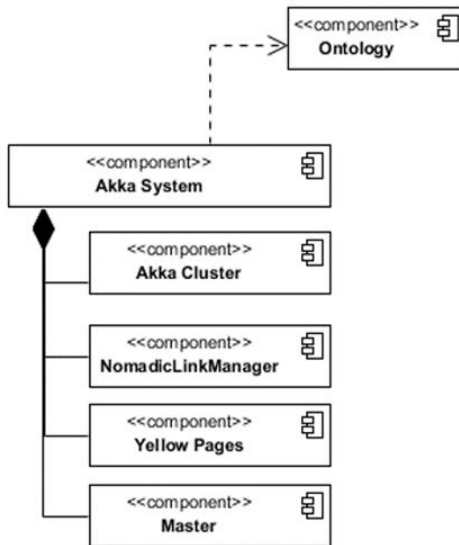


Рис. 4. Архитектура информационно-коммуникационной платформы

*Akka Cluster* – это модуль расширения платформы *Akka*, предоставляющий возможности объединения отдельных *Akka System* в единую р2р-сеть. Специальная компонента *NomadicLinkManager* отвечает за управление ссылочной целостностью распределенных структур, таких как распределенный технологический процесс, где каждая операция должна ‘знать’, какой элемент бизнес-процесса ей предшествует и для каких из его элементов предшественником является она сама. Компонент

*Yellow Pages* – это агент, управляющий поиском и распространением информации о сервисах, предоставляемых отдельными узлами В2В-сети. Под сервисами в данном случае понимаются возможности по исполнению тех или иных типов бизнес-операций, а также атрибуты исполнителей, необходимые для определения возможности назначения бизнес-операции тому или иному исполнителю. Агент *Master* выполняет ряд важных инфраструктурных задач, таких как управление стартом узла, координация компонент одного узла между собой, супервайзинг и контроль других компонент узла, а также предоставление интерфейсов для корпоративной информационной системы и внешних систем, являющихся клиентами В2В-сети. Отметим, что в данной архитектуре отсутствует агент белых страниц, который предусматривается стандартом FIPA. Это связано с тем, что *Akka* по имени актора (агента) всегда может определить его адрес.

Эффективная реализация рассматриваемой системы (см., например, [2, 3]) с помощью многоагентной архитектуры и технологии стала возможной благодаря тому, что в ней используется простая модель агента, в основе которой лежит понятие актора [30], и эта модель намного проще *BDI*-модели. В этой реализации взаимодействие распределенных компонент поддерживается нестандартной агентской платформой, которая намного «легче», чем распределенная *p2p*-агентская платформа, предусматриваемая стандартом FIPA. Технология *Akka* естественно подходит для реализации многопоточных и распределенных систем. Она 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* [32]. Взаимодействие агентов основано на передаче некоторых событий, а не сложных *ACL*-сообщений, что также повышает вычислительную эффективность многоагентной реализации рассматриваемого весьма сложного приложения, сохраняя при этом все концептуальные преимущества парадигмы *MAC*.



## 6.2. 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 др. 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 плану графику. Чем сложнее работа, тем более высокой квалификации требуются специалисты для ее выполнения. Для некоторых видов работ требуется наличие у бригады дополнительного оборудования, например, сварочного. 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 – на месяц. Невыполненные заявки плана текущего дня переносятся на следующий день.

Содержание задачи управления мобильными бригадами состоит в следующем.

1. Автоматически распределить и выполнить наискорейшим образом открытый пул заявок, имеющийся на текущий день, в порядке срочности их выполнения в соответствии с приоритетами заявок – *критическая, важная, средняя* и *плановая* заявка. Сложность задачи планирования работ достаточно высокая вви-

ду большого количества ежедневных заявок и ограниченным количеством аварийных бригад. Например, в Средневожской газовой компании г. Самара обычно бывает порядка 50 заявок в день при наличии 25 бригад, а в Водоканале г. Волгоград – порядка 64 заявок, при том что общее количество бригад – 15. При этом большую роль играет фактор времени.

2. Дополнительно необходимо 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 той или иной бригады. В частности, при поступлении более срочной заявки нужно адаптивно определить бригаду, которая может ее выполнить и сможет прибыть на место быстрее других с учетом своих уже имеющих в плане работ, а 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 должен начисляться бонус в зависимости от сложности выполненных работ согласно регламенту работы сервисной службы. В качестве основного показателя эффективности работы бригады используется фактическое время выполнения работы по заявке и отсутствие повторных выездов на следующий день.

Решение задачи управления мобильными бригадами реализуется с помощью адаптивного МАС-планировщика, который способен планировать выполнение задач обслуживания практически в реальном времени, обеспечивая минимизацию общего пробега автотранспорта обслуживающих бригад, время, необходимое для выполнения заявок на обслуживание и общее число невыполненных заявок клиентов в текущий день.

Рассмотрим архитектуру разработанного МАС-планировщика, которая включает в себя виртуальный мир программных агентов, каждый из которых может обмениваться р2р-сообщениями с другими агентами для того, чтобы сообщить им о своих возможностях и своих предложениях по изменению параметров своего состояния (времени, стоимости, координат, статусов и др.).

Поток внешних событий генерируется заявками, которые поступают из колл-центров и от диспетчеров, сообщениями от ресурсов мобильных бригад о статусах заявок, сообщениями сервисов геопозиционирования GPS/Глонасс и карт о координатах, а также сообщениями диспетчеров о недоступности ресурсов и пробках на дорогах.

Сущность работы МАС-планировщика состоит в скользящей 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.

Логическая архитектура системы представлена на рис. 5.

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

В отличие от сложных формализованных BDI-моделей агентов, в разработанном МАС-планировщике используются простые агенты, которые вовлечены в интенсивные переговоры. Сам МАС-планировщик представляет собой распределенную систему активно взаимодействующих агентов следующих типов:

– *Агент заявки*, целью которого является поиск наиболее подходящей специализированной бригады, способной выполнить работу ее как можно скорее и с нужным качеством.

– *Агент региона* 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 между заявками и таким обра-

зом 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.

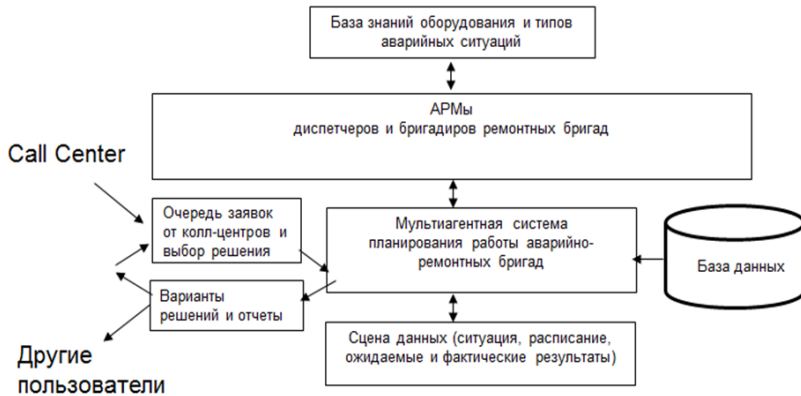


Рис. 5. Логическая архитектура системы адаптивного управления аварийно-ремонтными бригадами

Все решения по распределению и планированию работы бригад, которые формируются с помощью переговоров агентов по соответствующим протоколам и самоорганизации, предлагаются диспетчеру, который может при необходимости интерактивно перестроить расписание любой бригады (рис. 6).

Типичный протокол взаимодействия агентов при поступлении заявки на аварийное обслуживание с описанием аварийной ситуации через интерфейс колл-центра выглядит следующим образом:

– *Агент предприятия* создает *агента заявки* с атрибутами местоположения, статуса срочности, оценки сложности, требуемой специализации и квалификации бригады, оценки времени выполнения и предельного срока, а также стоимости работ.

- *Агент заявки* рассылает сообщение *агентам региона* с указанием места аварии, а также агентами регионов, которые граничат с регионом, где произошла авария.
- *Агенты регионов* запрашивают *агентов бригад*, способных по типу работ и географическому местоположению аварии выполнить заявку с ориентировочным временем начала работ;
- *Агенты бригад* запрашивают *агентов маршрутов* об оценке времени на подъезд к месту аварии с учетом текущей дорожной ситуации.

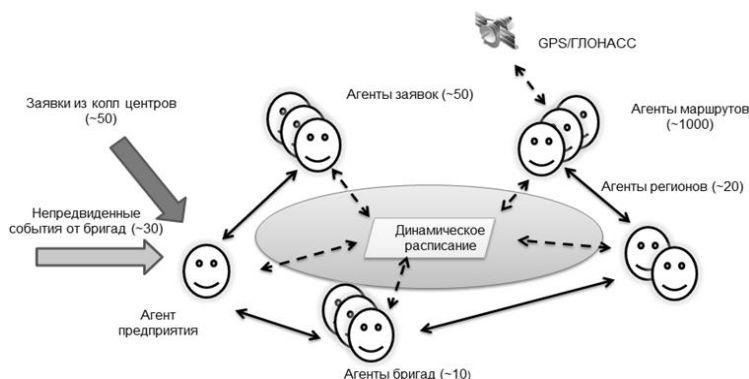


Рис. 6. Сцена мира агентов адаптивного управления аварийно-ремонтными бригадами (штрих линия – чтение из сцены и изменения в сцене, сплошная линия – обмен сообщениями и переговоры агентов)

При получении отклика от агента маршрутов *агент каждой бригады* строит изменения в своем текущем плане с учетом приоритетов запланированных заявок.

- При наличии свободных временных окон в своем расписании работ *агент бригады* отправляет *агенту предприятия* скорректированный план выполнения заявки;

- При наличии конфликтов в расписании *агент бригады* рассылает сообщения *агентам* запланированных заявок с оценкой возможных временных смещений и сброса менее приоритетных заявок.

– *Агенты* сдвинутых заявок в случае согласия уведомляют *агента бригады* и *агента предприятия* об изменении расписания. В случае, если новая заявка получила отказ на обслуживание в нужные сроки, *агент предприятия* обращается к *агентам* соседних *регионов* с запросом на возможность обслуживания. При отсутствии предложений со стороны соседних регионов *агент предприятия* принудительно планирует заявку подходящей бригаде в конце ее текущего расписания с учетом приоритета.

– *Агенты* менее приоритетных заявок, которые оказались вне расписания, повторно обращаются к *агентам регионов* и цикл такой цикл пересчета расписания повторяется;

– При выполнении бригадой очередной заявки *агент предприятия* отмечает фактическое время и корректирует статистику времен исполнения работ данного типа.

– В каждом новом цикле планирования *агент предприятия* оценивает время выполнения каждой заявки, загруженность бригад и прибыльность выполнения, направляя *агенту диспетчера* сообщение с текущими характеристиками расписания.

В результате такого цикла обмена сообщениями текущее расписание по событиям постоянно адаптивно подстраивается под вновь поступающие заявки и текущую дорожную ситуацию. Время, необходимое для оценки вариантов размещения заявки составляет обычно не больше 4 с.

Первое внедрение такого планировщика было выполнено в Средневожской газовой компании, которая уже несколько лет успешно эксплуатирует описанную систему.

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 задач в день). В результате внедрения описанного МАС-планировщика удалось добиться заметного улучшения различных показателей работы:

– сокращение времени реакции на непредвиденные события до 30%;

- повышение заинтересованности бригад в увеличении продуктивности работы;
- поддержка гибкого планирования в реальном времени;
- сокращение трудоемкости и числа ошибок диспетчеров в 2 раза;
- сокращение времени обучения новых диспетчеров в 3–5 раз;
- развития бизнеса компании без роста численности управленческого персонала.

Ведутся работы по другим практическим внедрениям описанного МАС-планировщика, в частности, на предприятии Водоканал в г. Волгограде.

## **7. Заключение**

Чрезмерно высокая оптимистичность прогнозов по широкому промышленному внедрению МАС была обусловлена различными причинами. МАС-технология с самого начала своего становления рассматривалась как достаточно универсальная интеллектуальная информационная технология, способная успешно конкурировать с другими технологиями в широком классе приложений. Однако это позиция оказалась не вполне оправданной. В результате в значительном числе прикладных задач более зрелые технологии оказались более успешными, что подорвало доверие к МАС-технологиям в глазах индустриального сообщества и, соответственно, интерес к их использованию и финансированию.

Большие негативные последствия для авторитета МАС-технологий обусловлены также существенной недооценкой сложности разработки МАС-приложений, которая в ее современной форме 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 разработки достижений в области телекоммуникаций, включая мобильные планшеты и сотовые телефоны.

Серьезным внутренним препятствием для развития МАС стала ориентация базовой модели агента и многоагентной системы на использование логического языка, дополненного модальными и темпоральными операторами, для формального описания *BDI*-модели поведения агентов. Эта модель казалась весьма привлекательной из-за ее богатых выразительных возможностей и математической корректности. Но на деле она оказалось совершенно бесполезной для создания промышленных систем. Особенности грантовой системы финансирования современной науки за рубежом и в России также сказались на результатах, в особенности, для тех исследователей, кто торопился сделать МАС своим новым «маркетинговым флайером».

Не менее важными факторами охлаждения первого «горячего» интереса к МАС в индустриальном сообществе стали значительные трудности промышленных внедрений, связанные со сложностью инструментальных программных средств для поддержки технологии МАС, с высокой стоимостью критических для бизнеса разработок и необходимостью решения множества других системных задач, не связанных напрямую с МАС-технологиями.

Вместе с тем, число успешных промышленных применений МАС продолжает неуклонно расти и захватывает все новые области. И хотя темпы этого роста пока слишком далеки от желаемых, а также от тех прогнозов, которые делались в этом отношении в начале 2000-х годов, агентские модели и технологии по-прежнему остаются перспективными для большого количества новых и будущих приложений, в частности, для тех приложений, которые характеризуются сетевой структурой автономных сущностей, высокой неопределенностью и динамикой поведения, большой размерностью, а также используют принципы самоорганизации в качестве базовых принципов управления и решения сложных задач.

Можно утверждать, что многоагентные технологии – это новый перспективный технологический базис для обеспечения «Grand Future Vision» для многих «сдвигов парадигм» в современных отраслях промышленности, в частности, в таких как

– Управление предприятиями: реформирование больших, централизованных, монолитных организаций в сети бизнес-



центров и центров знаний, работающих на виртуальном рынке, вплоть до «организаций» отдельных сотрудников;

– Авиация и космос: самоорганизующиеся группировки дешевых малых космических аппаратов и беспилотных летательных аппаратов как альтернатива большим и дорогостоящим спутникам, и аналогичным летательным аппаратам;

– Энергетика: переход от больших централизованных электростанций к распределенным сетям малых электростанций и других генераторов электроэнергии на местах, более дешевых и учитывающих индивидуальные нужды потребителей и т.д.

Очевидно, что новое поколение систем такого рода следует изначально ориентировать на Интернет вещей, где каждая вещь рано или поздно должна стать «умной». Она должна обладать не только датчиками и устройствами воздействия на объекты внешнего мира и коммуникации, но и принимать решения, причем, по возможности, согласованно с другими вещами. И тут МАС-технологии, без сомнения, помогут интеллектуализации как самих приложений, так и интеллектуализации их взаимодействия и кооперации.

Вместе с тем, для продуктивного развития МАС становится необходимой смена базовой парадигмы формализации модели агента и информационно-коммуникационной инфраструктуры, обеспечивающей поддержание его жизненного цикла, взаимодействия его с внешней средой и облачными сервисами для того, чтобы добиться вычислительной эффективности этой модели в режиме реального времени. В новой парадигме формализации МАС рационально строить как множество простых агентов с богатой компонентой взаимодействия и широким использованием принципов самоорганизации и эволюции, присущих живым системам.

Потребуется новые усилия ученых и специалистов-практиков для осознания этих новых принципов и для разработки качественно новых инструментов создания МАС в парадигме, основанной на самоорганизующейся поведенческой модели агентов и их взаимодействии.

Результаты этих разработок будут востребованы для решения многих сложных задач и создания нового поколения интеллектуальных систем в парадигме Интернета вещей, более от-

крытых, гибких и эффективных для ежедневного использования потребителями.

**Благодарности.** Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (проекты 16-01-00759 и 14-07-00493). Часть исследований выполнена в рамках проекта № 214 подпрограммы «Интеллектуальные информационные технологии и системы» Программы Президиума РАН I.5П. Частичное финансирование проекта выполнялось за счет средств проекта № 0073-2015-0003, выполняемого в СПИИРАН по бюджетной тематике.

### **Литература**

1. АНДРЕЕВ В.В., БАТИЩЕВ С.В. ВИТТИХ В.А. и др. *Методы и средства создания открытых мультиагентных систем для поддержки процессов принятия решений* // Известия Академии Наук. Теория и системы управления. – 2003. – №1. – С. 126–137.
2. БУХВАЛОВ О.Л., ГОРОДЕЦКИЙ В.И., КАРСАЕВ О.В. и др. *Производственная логистика: Стратегическое планирование, прогнозирование и управление конфликтами* // Известия ЮФУ. – 2012. – №3. – С. 209–218.
3. БУХВАЛОВ О.Л., ГОРОДЕЦКИЙ В.И., КАРСАЕВ О.В. и др. *Распределенная координация в В2В- производственных сетях* // Известия Южного федерального университета. Технические науки. – 2013. – №3. – С. 193–203.
4. ВИТТИХ В.А., СКОБЕЛЕВ П.О. *Метод сопряженных взаимодействий для управления распределением ресурсов в реальном времени* // Автометрия. – 2009.– Т. 45, №2. – С. 78–87.
5. ВИТТИХ 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.

6. ГОРОДЕЦКИЙ В.И., КАРСАЕВ О.В., САМОЙЛОВ В.В. и др. *Инструментальные средства для открытых сетей агентов* // Известия РАН. Теория и системы управления. – 2008. – №3. – С. 106–124.
7. ГОРОДЕЦКИЙ В.И. *Многоагентная самоорганизация в В2В-сетях* // XII Всероссийское совещание по проблемам управления ВСПУ-2014, Москва, 16–19 июня 2014 г.: Труды. – М.: Институт проблем управления им. В.А. Трапезникова. [Электронный ресурс]. – URL: <http://vspu2014.ipu.ru/proceedings/vspu2014.zip> (дата обращения 11.02.2016 г.). – С. 8954 – 8965.
8. 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.
9. ГОРОДЕЦКИЙ В.И. *Самоорганизация и многоагентные системы. II. Приложения и технология разработки* // Известия РАН. Теория и системы управления. – 2012. – №3. – С. 102–123.
10. ГОРОДЕЦКИЙ В.И. *Теория, модели, инфраструктуры и языки спецификации командного поведения автономных агентов. Ч. 1* // Искусственный интеллект и принятие решений. – 2011. – №2. – С. 19–30.
11. 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.
12. СКОБЕЛЕВ П.О. *Мультиагентные технологии в промышленных применениях: к 20-летию основания Самарской научной школы мультиагентных систем* // Мехатроника, автоматизация, управление. – 2010. – №12. – С. 33–46.
13. СКОБЕЛЕВ П.О. *Открытые мультиагентные системы для оперативной обработки информации в процессах принятия решений* // Автометрия. – 2002. – №6. – С.45–61.

14. *ACL – Agent Communication Language.* – URL: <http://fipa.org/specs/fipa00061/SC00061G.pdf> (дата обращения: 19.01.2016).
15. *AgentBuilder – an integrated software toolkit that allows software developers to quickly develop intelligent software agents and agent-based applications.* – URL: <http://www.agentbuilder.com> (дата обращения: 19.01.2016).
16. BERNON C. et al. *ADELFE: A Methodology for Adaptive Multi-Agent Systems Engineering* // Proc. of the 3th International Workshop on Engineering Societies in the Agents World. – 2002. – P. 156–169.
17. BONOMI F. et al. *Fog computing and its role in the internet of things* // Proc. of the First Edition of the MCC Workshop on Mobile Cloud Computing, (MCC 2012), New York, NY, USA, 2012. – P. 13–16.
18. BURRAFATO P., COSENTINO M. *Designing a multi-agent solution for a bookstore with PASSI methodology* // Proc. of International Conference on Agent-Oriented Information Systems. (Eds. Giorgini P., Lesperance Y., Wagner G., Yu E.), 2002. – P. 119–135.
19. CAIRE G. et al. *Agent Oriented Analysis using MESSAGE/UML* // Agent-Oriented Software Engineering II, Springer-Verlag, Lecture Notes in Computer Science. – 2002. – Vol. 2222. – P. 119–125.
20. COHEN P., LEVESQUE H. *Teamwork* // Nous. – 1991. – Vol. 25. – P. 487–515.
21. *Cougaar Agent Architecture.* – URL: <http://cougaar.org/wp/documentation/tutorials/> (дата обращения: 19.01.2016).
22. DELOACH S. *Analysis and Design using MaSE and agentTool* // Proc. of the 12<sup>th</sup> Midwest Artificial Intelligence and Cognitive Science Conference (MAICS), Miami University Press, 2001.
23. DELOACH S., GARCIA-OJEDA J. *The O-MaSE Methodology* // Eds.: M. Cossentino et al. Handbook on Agent-Oriented Design Processes, Springer-Verlag, Berlin–Heidelberg, 2014. – P. 253–285.

24. DELOACH S.A *Moving multi-agent systems from research to practice* // Int. J. Agent-Oriented Software Engineering. – 2009. – Vol. 3, No. 4. – P. 378–382.
25. *FIPA Agent Management System*. – <http://www.fipa.org/specs/fipa00023/XC00023H.html> (дата обращения: 19.01.2016).
26. *Gartner. Top Strategic Predictions for 2016 and Beyond: The Future Is a Digital Thing*. – URL: <https://www.gartner.com/doc/3142020?refval=&pcp=mpe> (дата обращения: 19.01.2016).
27. GARCIA-OJEDA J. et al. *AgentTool Process Editor: Supporting the Design of Tailored Agent-based Processes* // Proc. of the 24th Annual ACM Symposium on Applied Computing, March 8–12, 2009, Honolulu, Hawaii, USA.
28. GORODETSKY V., KARSAEV O., SAMOYLOV V. et al. *Support for Analysis, Design and Implementation Stages with MASDK* // Eds.: Luck M., Gomez-Sanz J.J. – Springer, Heidelberg, LCNS. – 2009. – Vol. 5386. – P. 272–288.
29. GORODETSKY V., KARSAEV O., SAMOYLOV V. et al. *P2P Agent Platform: Implementation and Testing* // Lecture Notes in Artificial Intelligence. – 2009. – Vol. 5319. – P. 41–54.
30. HEWITT C., BISHOP P., STEIGER R.A *Universal Modular Actor Formalism for Artificial Intelligence* // Proc. of International Joint Conference on Artificial Intelligence (IJCAI-1973). – P. 235–245.
31. HYACINTH S. et al. *ZEUS: A Toolkit for Building Distributed Multi-Agent Systems* // Proc. of the 3rd annual conference on Autonomous Agents (AGENTS'99), New York, NY, 1999. – P. 360–361.
32. *JADE*. – URL: <http://sharon.csel.it/projects/jade/> (дата обращения: 19.01.2016).
33. KAMINKA G. *Robots Are Agents, Too!* // Keynote Lecture. International Conference on Autonomous Agents and Multi-agent Systems (AAMAS 2007), Honolulu, Hawaii, May 2007.
34. LEITAO P., VRBA P. *Recent Developments and Future Trends of Industrial Agents* // Proc. of «Holonc and Multi-Agent Systems for Manufacturing» (HoloMAS-2011), Vol. 6867 of the series Lecture Notes in Computer Science. Springer Verlag, 2011–2012. – P. 15–28.

35. LUCK M. et al. *Agent Technology: Computing as Interaction (A Roadmap for Agent Based Computing)* // AgentLink. – 2005. – URL: <http://www.agentlink.org/roadmap/> (дата обращения 19.01.2016).
36. LUCK M., GOMEZ-SANZ J. *Agent-oriented Software Engineering IX* // Lecture Notes in Computer Science. Springer. – 2009. – Vol. 5386.
37. MYLOPOULUS J., CASTRO J. *Tropos: A Framework for Requirements-Driven Software Development* // Proc. of 12th Conference on Advanced Information System Engineering (CAISE), 2000.
38. MÜLLER J., FISHER K. *Application Impact of Multiagent Systems and Technologies: A Survey* // In «Agent-Oriented Software Engineering» book series. – Springer, 2013. – P. 1–26.
39. PADGHAM L., WINIKOFF M. *Prometheus: A Paradigm Methodology for Engineering Intelligent Agents* // Proc. of the Workshop on Agent-Oriented Methodologies, 2002. – P. 97–108.
40. RZEWSKI G., SKOBELEV P. *Managing complexity*. – WIT Press, London–Boston, 2014. – 156 p.
41. RoboCup Soccer Server. <http://dangermouse.brynmawr.edu/pyrobot/tars/rcsoccersim.pdf> (дата обращения: 19.01.2016).
42. SIERRA C. et al. *SADDE: Social Agents Design Driven by Equations* // In: Methodologies and Software Engineering for Agent Systems: The Agent-Oriented Software Engineering Handbook. – Kluwer Academic Publishers, NY, 2004. – P. 1–24.
43. SKOBELEV P. *Multi-Agent Systems for Real Time Adaptive Resource Management* // Industrial Agents: Emerging Applications of Software Agents in Industry / Eds.: P. Leitão, S. Karnouskos. – Elsevier, 2015. – P. 207–230.
44. SMITH G. Contract Net Protocol: *High-level Communication and Control in a Distributed Problem Solver* // *IEEE Transactions on Computers*. – 1980. – C-29(12). – P. 1104–1113.

45. STRASSNER J. *Using Agents and Autonomic Computing to Build Next Generation Seamless Mobility Services* // Keynote Lecture. International Conference on Autonomous Agents and Multi-agent Systems (AAMAS 2007), Honolulu, Hawaii, May 2007.
46. SYCARA K., SUKTHANKAR G. *Literature Review of Teamwork Models* // CMU-RI-TR-06-50, November 2006, Robotics Institute Carnegie Mellon University. [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: 19.01.2016).
47. TAMBE M. *Towards Flexible Teamwork* // Journal of Artificial Intelligence Research. – 1997. – No. 7. – P. 83–124.
48. *Technology Reports, Universal Description, Discovery, and Integration (UDDI)*. – URL: <http://xml.coverpages.org/uddi.html> (дата обращения: 19.01.2016).
49. VAN DER HOEK W. *Logical Foundations of Agent-based Computing*. // In: Multi-agent systems and applications. Lecture Notes in Artificial Intelligence / Eds.: M. Luck, V. Marik, O. Stepankova, R. Trappl. – Vol. 2086. – Springer Verlag, 2001.
50. WOOLDRIDGE M. *An Introduction to MultiAgent Systems*. – John Wiley & Sons, 2009. – 368 p.
51. YI SH. et al. *Security and Privacy Issues of Fog Computing: A Survey* // Wireless Algorithms, Systems and Applications. – Vol. 204 of the series Lecture Notes in Computer Science. – Springer International Publishing, Switzerland, 2015. – P. 685–695.
52. ZAMBONELLI F., JENNINGS N., WOOLDRIDGE M. *Developing Multiagent systems: The GAIA methodology* // ACM Transactions on Software Engineering and Methodology. – 2003. – No. 12(3). – P. 417–470.

## **INDUSTRIAL APPLICATIONS OF MULTI-AGENT SYSTEMS: CURRENT STATE AND PROSPECTS**

**Vladimir Gorodetsky**, St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences, St. Petersburg, Doctor of Science, professor (gor@iias.spb.su).

**Oleg Bukhvalov**, St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences, St. Petersburg, (psyhvet-er@gmail.com).

**Petr Skobelev**, Institute for the Control of Complex Systems of the Russian Academy of Sciences and Samara Aerospace University, Samara, Doctor of Science, professor (petr.skobelev@gmail.com).

**Igor Mayorov**, Software Engineering Company “Smart Solutions”, and Samara State Technical University, Samara, (imayorov@smartsolutions-123.ru).

*Abstract: Multi-agent system are perceived as one of the most promising technologies for distributed systems development. However current evidences show a lack of industry-level applications. The paper examines the main trends and prospects in development of industrial applications of multi-agent systems and technology, as well as analyzes the recent forecasts and the current state-of-the art with its practical applications. A survey of currently present researches, techniques, frameworks and standards is provided and supported with the most recent statistical data. The paper also analyzes external and internal obstacles for broad commercialization of multi-agent systems and technologies and the lessons learnt through these analyses. The paper describes properties and types of current and future applications for which multi-agent technology has unquestionable advantages including enterprise management, aviation and space, energetics, transportation etc. The paper also shows that multi-agent systems and technologies currently have no alternatives in self-organized control of large-scale objects of network structure.*

**Keywords:** multi-agent systems, industrial applications, self-organization, networked objects.

*Статья представлена к публикации членом редакционной коллегии Д.А. Новиковым.*

*Поступила в редакцию 17.03.2017.*

*Опубликована 31.03.2017.*