

# SYSTEM ENGINEERING VIEW ON MULTI-AGENT TECHNOLOGY FOR INDUSTRIAL APPLICATIONS: BARRIERS AND PROSPECTS

**Vladimir Gorodetsky**

St. Petersburg State  
Electro-technical University  
(LETI)  
Russia  
vladim.gorodetsky@gmail.com

**Petr Skobelev**

Institute of Automation and  
Information Technologies  
Samara State Technical University  
Russia  
petr.skobelev@gmail.com

**Vladimir Marik**

Czech Institute of Informatics, Robotics, and Cybernetics  
Czech Republic  
vladimir.marik@cvut.cz

Article history:

Received 24.04.2020, Accepted 17.06.2020

## Abstract

For over a quarter of a century, multi-agent systems have been considered as one of the most promising technologies for conceptualization, software development and implementation of Artificial Intelligence (AI) solutions. Not so widely but multi-agent technology is also considered as a way of designing complex adaptive systems based on bio-inspired principles of self-organization and evolution. However, in practice, the industry rarely uses multi-agent technology, despite the appearance of new classes of applications for which it is the perfect match, for example, smart cyber-physical systems with digital twins of controlled objects. The paper analyzes the recent anticipations and real achievements in the practical use of multi-agent systems at the industry level. It also identifies the engineering problems that currently impede the extensive industrial implementation of multi-agent systems and technologies as well as the ways to overcome them. Finally, prospects for development of these technologies are evaluated up to the level of industrial implementation.

## Key words

Complex problems, multi-agent systems, industrial applications, decision making, agent interaction, self-organization, Internet of Things.

## 1 Introduction

For over a decade, the world's leading rating agencies have been including multi-agent systems (MAS-

systems) and technologies (MAS-technologies) into the list of the most promising information technologies (IT) of the near future.

Many experts in the IT sphere also share optimistic feeling about the prospects of MAS for designing Artificial Intelligence (AI) solutions with distributed problem solving.

However, for some reason, the IT industry is holding off on using such systems, even though scientists and specialists of many leading universities in the world continue to develop this technology.

On the other hand, no one can deny the appearance of new complex problems and types of software applications, which seem to have been specially created for MAS' application. First and foremost, this applies to a many kinds of extremely complex systems and not fully formalized mathematical problems (including resource management, text understanding, data mining, etc.), wide class of mobile applications, cloud services and the Internet of things, collective robotics, human-machine systems and all other networking applications. One of very new promising application for MAS is smart cyber-physical systems with digital twins of controlled objects and positive and negative feedback loops in the networking grids of such systems.

One of the new and most booming areas of applications for multi-agent systems is the complex networks in cybernetics [Fradkov, 2015], including computational networks of microprocessors, smart transport systems,

swarms of mobile robots, etc. The new vision of such applications assume that the complex system consist from the many partially or fully autonomous elements with the ability to take decisions and negotiate these decision with their neighbors locally or with any other globally.

Such kind of applications are also required for natural science for modelling immune systems and organisms behavior, preventing ecological catastrophes, simulations of historical events, designing and analyzing new drags side effects, etc. Not so widely, but multi-agent technology is also considered as a generic tool for designing complex adaptive systems of any nature based on bio-inspired fundamental principles of self-organization and evolution.

The world community of MAS scientists actively develops new collective decision-making models, including forming coalitions of agents and robots. In the focus of R&D one can find logic-based specification languages to provide the required expressiveness and improved efficiency of the BDI-models of autonomous agents, adaptive learning mechanisms for agents, theoretical methods for verifying distributed MAS, study and develop new principles for functioning of agent communities and their cooperation, etc.

Nevertheless, the industry hardly reacts to new models, methods and architectures of MAS, and to the technologies offered. Why does this happen and when can we expect, if any, the use of multi-agent technologies at the industry level? Will it ever happen at all?

Academic and industry communities of experts demonstrate optimism that the time for MAS technologies will come in the immediate future. But the first signals of high complexity of developing and delivering industrial MAS solutions for real-life applications have been clearly discovered in last decade [Luck et al., 2005] [Leitao, 2009] [Pechoucek and Marik, 2008] [Marik and McFarlane, 2005].

One of the first attempts to systemize and structure all the activities in the industrial applications of MAS were taken by Agent Link – an association of EU researchers and developers that produced the Agent Roadmap [Luck et al., 2005]. Agent Link has made a big impact on MAS developments and applications in the industry; for example, Industry Truck became one of most practical parts of AAMAS conferences.

First barriers for MAS in industry were identified in [Leitao, 2009] [Pechoucek and Marik, 2008] [Marik and McFarlane, 2005] including required mentality shift (distributed thinking vs centralized), the lack of investments and industrial non-maturity of the technology.

One of the detailed overviews on industrial MAS applications was undertaken by P.Leitao and P.Vrba in [Leitao and Vrba, 2011]. It was shown that there are not so many real industrial MAS applications in the market and the existing ones are mainly found in academic environments. Among engineering issues, the topmost ones include real-time constraints, integration with the physical hardware and legacy systems, lack of methodologies and

tools to make MAS developments simpler. However, already in [Leitao and Vrba, 2011], it was also stated that MAS requires “consideration of bio-inspired techniques, and particularly self-organization and emergent behavior, enhancing multi-agent systems to support the engineers of more robust, adaptive, reconfigurable and responsive systems”.

One of the extensive surveys on MAS case studies in the industry was made for 152 MAS in different areas in comparison with previous AgentLink case studies [Muller and Fisher, 2013]. This survey covers industrial MAS maturity, vertical sectors, and the usage of programming languages and platforms. It shows the coming stage of disillusion and overcoming of hype — but as it was reported in the paper that the key engineering problems and show-stoppers were not possible to identify because of the lack of required data.

More advanced engineering study on industrial MAS was made in [Leitao et al., 2013]. Firstly, starting with real time control, it was learned that the delivery of MAS solutions requires a “radical change in the way control solutions have been designed, implemented and maintained for decades”, and it is strongly required “to move from procedural, task-oriented, and controller-centric programming to object and/or service-oriented programming” [Leitao et al., 2013]. Secondly, important engineering trends and issues to be solved were identified: required convergence of multi-agent technology and service-oriented architectures, and symbiosis of execution and simulations among other issues. Thirdly, and the most important, it was highlighted that MAS needs more intelligent behavior, reasoning and interaction, based on agents’ ontologies, internal world representation models, sophisticated decision making, learning, etc. Looking from the industrial point of view, “one of the major technological roadblock is the inability of the new technology to respect contemporary industrial requirements for real-time capabilities, robustness, availability of mature engineering tools, safety, and standardization” [Leitao et al., 2013]. There is also low performance, lack of live ready-to-use solutions and industry demonstrations, difficulties with ROI analysis, etc.

The number of industrial MAS solutions and applications for solving extremely complex problems of adaptive resource management is presented in [Rzevski and Skobelev, 2014]. Research and design and commercial issues of these MAS developments as well as valuable benefits for customers and lessons learned from industrial applications of MAS for managing trucks and factories, field services, supply chains, aerospace, railways, satellites and some other are systemized and analyzed in [Skobelev, 2018].

This paper provides an outlook on barriers for developing industrial MAS from engineering viewpoint.

The main idea of this paper is that massive application of MAS-technologies in industry will require full reconsidering of basic paradigm in formalization of agent’s

and MAS models (namely BDI model) as well as the engineering and technological aspects of their design and software implementation. Instead of following the formal logic, industrial MAS needs to be driven by engineering models and methods of self-organization; for example, based on virtual market mechanisms with iterative auctions and competitive equilibriums, which are already proven to solve various extremely complex problems of adaptive resource management in real time.

A renewal of attention paid to the industrial applications of MAS is expected in connection with penetration of the Industry 4.0 ideas. The digital twins introduced by Industry 4.0 can be easily converted into agents by a simple agentification process, which adds the shared goals and communication capabilities to each of the twin [Wahlster, 2019]. The agentified twin is linking the world of physical manufacturing with the virtual processes of negotiations, decision making and control. It is simply bridging the physics of manufacturing with the virtual world of information processes. When considering the manufacturing facility as a complex system to be controlled, the capability of linking the physical and virtual worlds brings a new challenge for efficient implementation of Cyber-physical Systems (CPS) and IoT within the frame of complex cybernetic systems. Coming back to the cybernetic insight, this potentially opens quite a new perspective for enhancement the MAS area by principles of machine learning, feedback control, and stability management. The world of physical manufacturing cannot be controlled without using the principles of cybernetics.

For these reasons, this paper contains a comparative analysis of the expectations regarding the industrial application of MAS-technologies and their practical use (Section 2). It analyzes the causes and problems that make the introduction of MAS-technologies into practice difficult (Section 3) and proposes the ways to overcome recognized problems (Section 4). In the last section, new areas of applications are identified for industrial MAS developments.

It appears that the Internet of everything, including documents, people, machines, devices and any other physical an abstract things, which has begun to develop recently, will require a completely new agent-based paradigm and it will bring value not only for Industry 4.0 aimed at automation of manufacturing, but also for Industry 5.0, which, as expected, will be more focused on human development and support for self-organization processes in the society.

## 2 On the Way to Manage Complexity

Growing complexity of models of objects and processes in natural science, healthcare and agriculture, engineering and business requires new generation of methods and tools for solving complex problems in cases, which cannot be solved with traditional mathematical

methods and tools, for example:

- Models of business growth;
- Model of cells, plants or any other living organism;
- Models of biocoenosis in nature;
- Models of economics and markets;
- Model of human brain;
- Models of social networks behavior;
- Models of human team work, etc.

These all objects and processes are not linear and not deterministic by the nature and all based on positive and/or negative feedback mechanisms but our existing models are mainly based on well-known linear laws in mechanics, electricity, optics, etc.

Any well-defined generic methods and tools do still not cover this growing gap between complexity of considered objects and processes and simplicity of applied models.

And researchers and developers are still lacking suitable technologies for researching such complex phenomena like formation of organic molecules, physiology of living organisms, interactions in social revolution and insights, etc.

It is interesting, that to address the issue of complexity one can discover the same basic trend in very different natural sciences:

- **Physics:** Complex Adaptive Systems by Nobel Prize Laureate Prof. Ilya Prigogine (1977): our world is open dissipative system; auto-catalytic reactions are the basis for evolving complex systems, self-organization is a way how complex phenomena is constructed;
- **Psychology:** Society of Mind by Prof. Marvin Minsky, (1986): human decision making and reasoning are not linear and logical; our thoughts are always competing and cooperating; human insights are examples of auto-catalytic reactions; emergent intelligence is self-organization;
- **Biology:** Holonic Systems by Artur Kestler, introduced in the book “The Ghost in the Machine” (1967), are considered as self-organization and evolution like in biocenosis in nature. A. Kestler concluded that, although it is easy to identify sub-wholes or parts, wholes and parts in an absolute sense do not exist anywhere, all things linked.

Moreover, such trends can be discovered not only in natural sciences but also in mathematics, engineering and business, which become interested to use the same principles:

- **Mathematics:** step from centralized optimization – to game theory and to distributed optimization;
- **Engineering:** replace of big supercomputers – by networks of personal computers;
- **Space Industry:** step from very big and expensive satellites – to swarms of nanosatellites;

- **Business:** reforming large corporations into networks of small innovative companies, which operate autonomously;
- **Medicine:** European medicine is mainly focused on detailed study of different organs, but Chinese medicine – on interaction between parts of organism.

Looking from cybernetics point of view, the main trend could be formulated as a step from centralized, pre-defined, monolithic, hierarchical structures to open, distributed, networking and interacting ones with horizontal and vertical feedback loops. It means that any complex system in nature science or in engineering or business could be considered, analyzed or designed only as a self-organizing system, which parts are fully or partially autonomous, can take decisions, interact dynamically and evolve in time. The behavior of such systems based on unstable equilibria is always not linear by definition and it is difficult to specify, simulate, analyze and understand, because it is counter-intuitive. A number of relevant researches are already on the way [Plotnikov et al., 2019], [Aleksandrov et al., 2019], [Plotnikov and Fradkov, 2019] defining totally new mathematics for describing, specifying and modelling of such kind of systems.

In this context, multi-agent technology could be considered as a one of new advanced models and tools for designing, implementation and analyzing of such kind of complex systems applicable not only to natural science, but also to new business, engineering, agriculture, etc.

In future, with the use of bio-inspired multi-agent technology, one will have chance to develop digital twins of cells, immune systems, plants, human organisms, etc. However, from the other side, in the same way one can design enterprise resource planning systems and systems of systems for the whole supply chains of smart enterprises of future.

Following this way, the multi-agent technology could become a driving engine for designing new generation of “digital eco-systems” with the colonies of autonomous AI systems based on competition and cooperation of smart systems [Dig, 2020].

### 3 Practical Use of Multi-agent Technologies: Expectations and State-of-the-art

#### 3.1 Expectations

The concept of MAS was proposed in the mid-1980’s. It was highly appreciated by both the scientific and industrial communities immediately. Within the first two decades, its basic theoretical foundations were built, and active developments in the field of technology and tools for its support began. In early 2000s, several well-defined methodologies and some software tools for MAS design and implementation had been proposed. In 1996, FIPA (*Foundation for Intelligent Physical Agents*) was established. Its main objective was scientific substantiation of standards in the field of agents and MAS. In

2005, FIPA became one of the standardization committees in IEEE. At that time, it was expected that MAS-technologies are ready to capture the leadership as principally new design paradigm and technology for distributed systems of the industry level and for the widest range of applications in MAS and industrial community.

At that time, it seemed that there are enough foundations for such optimistic expectations. Indeed, this concept appeared very attractive and natural for understanding and implementations because it suggested solving of problems and building systems like they are done in the living systems and in the human society, particularly, through interactions and underlying self-organization. The basic principle of MAS-applications conceptual modeling is based on the representation of a complex problem as a set of relatively simple, near autonomous, interacting subtasks. Their solution is entrusted to software agents, which are developed and programmed almost autonomously, operate asynchronously and in parallel, and interact through the exchange of the messages represented in terms of a language close to the natural one, i.e., like how it is done in the human society. The interactions, implemented through dialogues and protocols, can be quite complex and diverse. Simultaneously, agents can generate events and send messages to other agents, develop and negotiate variants of decisions, forward data, evaluate results of solving their subtasks, create new tasks for other agents, maintain synchronization of collective actions, transmit feedback signals, etc. This concept seems to be very generic and natural for the applications involving many different participants with their own interests or any other relatively autonomous entities. For example, this applies to the tasks of transport logistics, where individual orders and cargoes generate demand, while vehicles, drivers, repair and refueling stations are the resources supporting for the plan fulfillment. The same applies to production logistics, where agents can represent orders, individual production operations, workers, equipment, robots, products, instruments and materials, etc.

Since the very beginning, the ideas of autonomous agents and MAS were especially attractive for applications in the area of individual and collective robotics, to such an extent that the terms *agent* and *robot* usually were not differentiated [Kaminka, 2007], so much that it remains true even up to date. At the same time, classes of potential MAS applications continued to expand. New wide class of applications for which MAS-technology seems ideal is born by the approaching era of the Internet of things, where exactly the interactions of distributed autonomous objects of the physical, virtual and social natures are in the focus of complex system performance [IoA, 2020].

One of the most attractive aspects of MAS-paradigm is its ability to support natural programming of interactions of the numerous components of the program. In MAS concept this task is separated from programming of agents and is implemented through designing

of protocols of interaction which are not directly linked with agents classes implementation. It is important to note that MAS-paradigm originally was formulated as a paradigm of *computation as interactions* [Luck et al., 2005]. The concept of autonomy of software object and message exchange for implementing interactions proved to be very attractive and productive in practice. It is not surprising that later, message exchange became a norm in other architectures and technologies for developing complex systems.

The project *tAgentlink III* of the European Commission FP-6 (2004–2005) became a first indication of the maturity level of developments in the field of MAS theory and practice. The main result of this project was the *Road Map: “Agent Technology: Computing as Interaction”* [Luck et al., 2005]. This document virtually summed up the twenty years of development of the paradigm and gave a highly optimistic forecast of the prospects for industrial applications of MAS.

The Industry 4.0 is based on system integration principles – physical manufacturing entities (drilling machines, robots, transportation vehicles, conveyers, products and semi-products, etc.) can be represented by individual agents. These agents can also embody the corresponding CAD and simulation models together with the algorithms for physical entities control. The complex system integration itself is happening on the communication and negotiation level in the virtual space of informatics. So, principles of cyber control applied locally to the physical entity (as a part of responsibility of an individual agent) can be (and must be) complemented by a higher-level system integration which also explores the principles of cybernetics. Thus, the MAS vision enables the physical devices to be controlled in their local operations locally, but being coordinated and controlled as a part of a bigger system consisting of many components. There is no other efficient way how to compose, control and manage complex system consisting of units operating in the physical world in real time. Solutions offered by information systems industry cannot operate elements of physical world easily.

One of the new promising application is cyber-physical MAS with digital twins of controlled objects, which can be also synchronized with real objects. Examples of such objects could be not only machines or enterprises, but also living organisms, for example, in wheat production digital twin of plant can represent main phases of plant developments with ongoing re-forecasting of the harvest and generating advices for farmers [Laryukhin et al., 2019]. The next step could cover the whole biocoenosis of the field – according to modern research, plants can detect insect attacks by ‘sniffing’ each other’s aromas [King, 2020].

A real revival of MAS ideas in connection with the Industry 4.0 is expected.

### 3.2 State-of-the-art

However, in the beginning of the 2000’s, something in the development of MAS theory and technology went

wrong. Public perception of works in the field of MAS became less significant [Marik and McFarlane, 2005]. Successful developments in the field of intelligent applications that were implemented during this period by the world’s leading IT companies, particularly Apple, Facebook, Google, SAP, were not at all related to MAS or MAS-technologies, at least, in the public perception. For example, in the invited paper of John Strassner at the AAMAS 2007 conference [Strassner, 2007], one of the key topics was the analysis of the state of industrial MAS-applications. The author found no more than six MAS-applications developed over the first 20-year history of this technology, which, although approximately, could be qualified as industrial ones.

Scott A. DeLoach, a well-experienced specialist in MAS-technologies, under whose guidance the *O-MaSE* methodology and *agentTool* were developed [DeLoach, 2009], clearly indicated the absence of the expected progress in the industrial applications of MAS. According to his opinion, despite more than twenty years of agent technology age, this area is still at an early stage of maturity compared to, for example, the maturity of the object-oriented approach (OOP). He also noted several significant gaps and deficiencies in the field of MAS theory and technology requiring clarification (see discussion in Section 3). According to his opinion, MAS-concept and technology should, first, demonstrate their ability to create complex adaptive and self-organizing distributed systems, but this has not yet happened.

In Jorg Muller with co-author paper [Muller and Fisher, 2013], after detailed analysis of the real state of the industry MAS developments, it was indicated: “While there is ample evidence that Multi-Agent Systems and Technologies are vigorous as a research area, it is unclear what practical application impact this research area has accomplished to date”. This work is interesting in several aspects. First, it provides a detailed overview of MAS applications developed till 2012 by evaluating the quality of the forecast given for the same period in the Agent Link Roadmap [Luck et al., 2005]. Second, it shows notable statistics allowing for a deeper evaluation of the real state of MAS applications.

While forecasting the MAS-application developments, the roadmap [Luck et al., 2005] stated that, for MAS technology, it is unreal to reach the industrial maturity level for 20 years (by 2005) of researches and developments. In this context, a comparison is made between the ages of MAS-technologies and OOP, noting that the latter has gained practical significance after over 30 years. For example, the OOP language C++ was created after 32 years, and JAVA after 39 years after birth of OOP concept. Now, the age of MAS-technologies is approaching 40 years, but very little new has happened.

The authors [Luck et al., 2005] also complain about the weakness of methodologies for designing MAS developed by that time, although it is a questionable opin-

ion. Indeed, by that time, several well-developed MAS methodologies were created and tested; for example, *Gaia* [Zambonelli et al., 2003], *Tropos* [Mylopoulos and Castro, 2000], *MaSE* [DeLoach, 2001], *ADELFE* [Bernon et al., 2002], *MESSAGE* [Caire et al., 2002], *Prometheus* [Padgham and Winikoff, 2002] although they were not supported by adequate tools. The intensity of research activities intended to design MAS methodologies and supporting software tools was quite high almost until 2010, when more than a dozen of new developments was completed.

However, currently only a small number of methodologies and tools remain to be the subjects of further developments.

In the period after 2005, the intensity of research and developments of MAS-technologies and supporting software tools did not decrease. However, the forecast of MAS developments growth given in [Luck et al., 2005] was found out too optimistic in comparison with the reality [Muller and Fisher, 2013],[Leitao and Vrba, 2012]. The survey [Muller and Fisher, 2013] analyzes 152 applications available to the author either from the authors or from scientific literature. However, later, the authors did not confirm more than half of them. Thus, the level and the very existence of these developments remain questionable [Muller and Fisher, 2013].

The most successful developments of MAS-applications of different maturity levels and with different ratio between automatic code generation/reuse of ready components and manual programming were performed in the fields of manufacturing, logistics, aerospace and power engineering.

Among leaders of industrial MAS developments is Rockwell Automation, Inc which made indispensable investments into the research and design activities on MAS [Vrba et al., 2011] and spent nearly two decades to make pioneering research and development in the application of holonic and MAS to the industrial automation domain. They went through a path from the first simple prototypes of holonic control systems (holoblocs) as an extension of the function blocks that were created more or less spontaneously and ad hoc, to the development of a comprehensive bundle of advanced methodologies, practices, and tools that cover all aspects of design, implementation, validation, and monitoring of agent control systems [Vrba et al., 2011].

Rockwell Automation started from the classical “old-fashioned” approaches to “novel” control paradigms. For the very first time they implemented Holonic Agent Architecture, Object-oriented design, Distributed Control Systems (DCS) with a clear differentiation between higher-level and lower level agents, sniffers to observe and evaluate the communication traffic, Agent Applications, Simulation Support including one the very first agent-based simulation tool for agent-based systems MAST, developed semantic technologies and ontology services provided by specialized agents, etc.

This case has shown that the novel control paradigms

based on MAS technology require introducing completely new methodologies, architectures, and tools that differ considerably from the traditional centralized approaches. It was declared that a new way of thinking of the system engineers and users is required to be successful in applications.

The result of two decades’ development showed in 2011 [Vrba et al., 2011] stated first issue — the necessity of providing potential customers with the quantitative analysis of the benefits of agent technology compared to “classical” technologies available on the market. The second issue is to educate future engineers to a distributed rather than strictly centralized way of thinking, and thus, prepare a future generation of skilled distributed system engineers. Third issue — to strengthen the collaboration between academia and industry and creating acknowledged centers with testbeds used for validation and verification of the agent concepts. Nowadays all the points are still actual. However, in total, the number of real applications was near twice less than it was predicted by the road map [Luck et al., 2005] for 2015. Moreover, probably the most unexpected fact was that the speed of new MAS application developments began to slow down gradually — reflecting recognized complexity of MAS solutions.

Identified difficulties triggers a real dissatisfaction and drop in the industrial companies’ interest in MAS-technology.

Practice has shown that many applications for which MAS-technology was considered as the most promising one [Luck et al., 2005], by 2013 were successfully implemented using other technologies. Among them, the most competitive ones were service-oriented technologies, grid computing, autonomous, ubiquitous and cloud computing, etc. The aforementioned technologies appeared much later but were able to quickly outperform the MAS-technologies in engineering aspects.

The paper [Muller and Fisher, 2013] became the first one to clearly state the alarming prospects of MAS-technologies in competition with other modern IT technologies. The MAS-applications developed till 2013 were analyzed from different viewpoints, but for this paper, it is essential to assess their maturity level. From this standpoint, the applications are divided into 3 groups [Muller and Fisher, 2013]:

1. Industrial systems or those close to them — 46 out of 152;
2. Research software prototypes of the industrial level, which were tested on real data, but not used in real work — 55;
3. Laboratory research prototypes, which were used for educational, scientific, and other purposes — 46.

The authors of [Muller and Fisher, 2013] lacked the reliable information about the remaining five developments. Thus, less than one-third of the analyzed applications turned out to be sufficiently mature, and their to-

tal number was five times less than the forecast of the roadmap [Luck et al., 2005]. However, only about half of them were ordered by industrial companies, but the rest were developed by university teams, i.e., were performed by groups which were subjectively interested in research, but not in the industrial use.

An important indicator of the trends in research and development of MAS-technologies during that period is the active use of methodologies and development tools — in 72% of cases, authors of developments used some or other methodology and/or software tools, usually the ones that were developed by themselves. This indirectly reflected the fact that MAS-applications during that period were developed mainly for testing of methodologies and tools, not for real applications.

The forecast on the types of applications that the authors of the road map [Luck et al., 2005] believed to be of the greatest interest by 2015 proved correct, in general. As it was predicted, the leading fields using MAS-applications at the industrial level were and still are in the following areas: transport and manufacturing (9 applications), telecommunications (9), e-commerce (4), aerospace (4) military applications (3), power engineering (2) and business process management (2). Unexpected outsiders, according to this aspect, found that healthcare systems (2), robotics (2), financial sphere (0), administrative management (0) and some other classes paid much better prospects in [Luck et al., 2005].

In addition to discussed MAS developments it is important to mention recently developed new software engineering technologies which are related to MAS as a supporting technologies or utilize the principals of MAS under new up-to-date labels (SOA, Twins, IoT or Cyber-Physical Systems, etc.).

Cyber-Physical Systems (CPS) refers to a specific category of systems with integrated computational and physical capabilities. Interconnected CPS components (sensors or actuator networks in combination with decision making logic) collaborating in peer-to-peer way facilitate to meet increasing requirements for flexibility and re-configurability. Such networks are named industrial cyber-physical systems (iCPS) [Klima et al., 2017]. But still they use the basic principles of MAS, despite the fact that the authors do not like to mention this. In case that CPS has on-board computational model of real object which is used for mirroring reality and forming actions it could be more sophisticated CPS (and MAS also can play a role of such model) but in practice many authors do not differentiate CPS from classical automation systems.

The next step of agent development was declared in 2009 in relation with the service-oriented architectures (SOA) [Mendes et al., 2009]. SOA as a style of software design is not being considered as a product or a follower of MAS. But there are many parallels between the agents and web services: agents provide capabilities to other agents in the same way as services are provided in service-oriented systems and also similar but less for-

malized messages are used to exchange data [Vrba et al., 2011].

Holonic multi-agent system, introduced in legendary paper of V. Van Brussel, P.Valkenaers and co-authors [Brussel et al., 1998], can be considered as a spin-off of multi-agent systems with pre-defined classes of agents: orders, products, resources and staff (later advanced by task agent, function of satisfaction and bonuses, etc. [Skobelev, 2015]) which can recursively form holarchies as a bottom-up structures (comparing with top-down hierarchies). In the beginning the holonic MAS solutions were mostly used for control of near-to-physical layer devices in manufacturing exploring the IEC 16499 standard [Vyatkin, 2011] but later were applied for solving complex problems of management [Skobelev, 2018]. The holonic MAS paradigm put focus on self-organization mechanisms based on specific classes and role models of agents and protocols of their interactions [Marik et al., 2002].

The new trend is also the application of semantic technologies (ontologies) to enhance the capabilities of agents for representing and exchanging knowledge, and as a consequence, to increase the openness, intelligence and flexibility of MAS solutions [Vrba et al., 2011].

These technologies significantly change engineering landscape of MAS solutions with specific features, models, methods, or tools for software implementation – providing more opportunities for designing complex systems as originally intended. Even under use of specific terminology it becomes finally clear that complex system can be modelled as a MAS-based system only in the case that there is self-organization of fully or partially autonomous agents which are able to take decisions and coordinate their decisions Without this context the mentioned above technologies can't form solid basis for advanced AI solutions matching complexity of the real world.

The following conclusions can be drawn from this discussion:

1. MAS-technologies are developing at a much slower pace and more problematically than the scientific community would like, and even more so compared to the needs of the industrial community. The existing grant-based financing of science also negatively contributes to it, as it requires the researchers to not “get stuck” in the field of specific technologies, but to develop new directions or solve specific and very narrow application problems.
2. The majority of MAS research and development projects is carried out, mainly, within the scientific community. Industry, which initially was the initiator and catalyzer of MAS developments and previously played the leading role in stimulating and financing them, have now, in fact, completely stopped its supports (Motorola, Siemens, etc.). Some companies and organizations, for example, IBM, Daimler, NASA and Google, actively use the concept

of agents as a part of their developments, but do not call them multi-agent and do not emphasize the agent components of their developments. Several companies (e.g. British Telecom) have reduced the volume of multi-agent system related developments [Muller and Fisher, 2013].

3. Agents and holons have provided a new abstraction metaphor for designing highly distributed and intelligent industrial control systems featuring new properties as autonomy, robustness, survivability, adaptation and reconfiguration. Despite these undoubted advantages, the exploitation of the developed concepts in industrial practice is still very low. The decision makers are reluctant to take the risk of being the first adopters of this technology at large scale due to many negative factors, such as higher investments, anxiety over effects of emergent behavior and lack of skilled maintenance personnel [Hall et al., 2005]. Thus, “classical” centralized and hierarchical architectures are still predominantly used in industry [Vrba et al., 2011].

4. The interest of industrial community regarding multi-agent systems and technologies has now decreased — new marketing flyers include neuron networks, deep learning, etc. But these systems do not have a common sense and are not solving complex problems in a way which humans do, fail to match complexity and require continuous re-training in conditions of uncertainty and dynamics.

Deep learning and other network based learning algorithms can be applied just locally. Much more sophisticated, knowledge-based models should be deployed on the system integration level. Such structural learning models are still missing. Thus, the Industry 4.0 solutions can explore the available deep learning algorithms locally. Cybernetic principles of complex system management are expected to shift the whole field ahead.

That is why, simultaneously, high latent turbulence, regrouping and restructuring of the AI market have been observed. In this market, multi-agent systems and technologies have a lot of new opportunities, which can be actively offered in combination with other solutions.

5. It is obvious that in recent years, the number of new niches and new classes of applications have been constantly increasing, and they have great prospects for MAS: distributed computations, green energy, autonomous vehicles, swarms of drones, etc.

Therefore, it is vital to understand the engineering reasons that currently hamper the practical use of the enormous potential of MAS technologies in industry.

#### 4 What are the Barriers for the Practical Use of the Great Potential of MAS?

Perhaps the most valuable thing that the theory and practice of MAS have initially offered for software in-

dustry is the conceptualization of complex systems. This natural conceptualization model and implementation architecture is valid for systems of almost any complexity, which attracts the attention of both the researchers and application developers. The analysis of literature on MAS-applications, as well as topics of papers presented at the leading conferences on MAS, proves that the clear majority of them are devoted to conceptual modeling and implementation architectures of applications. It should be noted that this trend is observed even now, although it is not as obvious as it was until around 2010. This is natural, because at this stage of application development, MAS-paradigm offers a comprehensible and attractive modeling option.

However, at the next stage, when it was time to build the engineering agent models, create the architecture and infrastructure supporting agent classes execution and interaction and specify the agent’s communication language, it did not turn out so easy. The reasons for this cannot be called subjective. The complexity of the MAS formal design and modeling created a barriers for efficient engineering implementation and delivery stages as well as for maintaining solutions on client side.

The MAS solutions was difficult to develop, understand and use for practitioners — it requires knowledge and skills for object-oriented programming, AI models and tools for collective decision making, tools for semantic knowledge representation in the form of ontologies, methods for combination and exploration semantics from different sources, parallel programming, telecommunications, etc.

However, the problems were much deeper — let’s outline the main engineering problem and obstacles in this respect:

1. *Lack of generally accepted understanding of MAS’ key concepts.* This is indicated particularly in [DeLoach, 2009]. The absence of clear definitions and non-doubtful agreements on the basic concepts of MAS greatly hinders the mutual understanding between researchers and developers. For example, most computer technology professionals agree with the definitions of basic OOP concepts, such as classes, objects, inheritance, and encapsulation. They easily handle these concepts in practice. At the same time, the experts in the field of MAS have different understandings of these basic concepts as agent, role, negotiations, plan, and possibility among others. The real problem in this field is that it is necessary to clarify these basic concepts and to reconcile their relationship with other similar concepts of OOP that are used to implement the agents and MAS. Perhaps, one of the ways to overcome this problem is *to develop basic ontology for domain-independent concepts* used in MAS. One of the examples of ontology for domain-independent behavioral concepts of the BDI-model can be found in [Gorodetsky et al., 2015].



2. *Lack of generally accepted notations for specification of MAS models.* Since there are generally no accepted definitions of basic agent concepts and interrelations [DeLoach, 2009], the absence of general notations for specification of the concepts and relations makes it difficult to study and compare the various models of MAS in practice. The standardization of MAS models and means for their specification is necessary, otherwise it is difficult to compare different approaches, models and architectures of MAS, which may lead to inadequate evaluations of the new MAS-technologies.
3. *Conceptual and computational complexity of logical formalization of the BDI-model of agents and MAS.* For almost three decades, the main efforts of researchers in the field of MAS theory have been aimed at developing models of *intelligent* agents. Already at the inception of these studies, several properties, which the agent must have were formulated. In general, each agent was considered as a truly intelligent entity with its well-developed knowledge base, or at least, with a knowledge model, capability of goal selection and mechanisms for planning goal-oriented behavior in an unpredictable external environment. This point of view has quite actively been promoted since many years. Specialists in the agents and MAS formal modeling were almost competing to provide the agent with more and more new intelligent abilities in terms of autonomous behavior, and the ability to understand the intentions of other agents. Naturally, it was not possible to specify and implement such intelligent capabilities of the agent by simple means. Thus, it led to constant sophistication of the agent and MAS formal models.

From the very beginning, the *BDI-model (Belief-Desire-Intention)* [Wooldridge, 2009] was chosen as the basic formal model of the intelligent agent. In this model, the knowledge, beliefs, intentions, and the agent's reasoning mechanisms are specified in terms of predicate calculus extended with modal and temporal operators.

Most researchers now adhere to the concepts of the BDI-model of the agent and MAS and their logical formalization. It is important to note that the *conceptual basis of the BDI-model* itself in terms of behavioral and motivational concepts is quite natural and convincing. However, its logical formalization is definitely difficult to understand for the application developers and is a significant barrier in the interpretation of the basic concepts of MAS [DeLoach, 2009], including the interpretation of the term "*BDI*" itself.

If we turn to the history of Artificial Intelligence (AI), we can recall that for a sufficiently long period, logical languages for knowledge representation and logical inference as a reasoning mechanism have been dominating. An example is the in-

famous Japanese computer project of computers of the fifth generation, which relied on the logical language Prolog, and which completely failed in the 1980's. One can easily find a firm analogy between the aforementioned fact of the AI history and what was happening. Unfortunately, it is still happening now in the field of agent and MAS theory. Moreover, it is proposed to overcome the weak expressiveness that is applied to autonomous agents and its interaction modeling through enriching the predicate calculus with modal and temporal operators. Thus, it received the name of the *BDI-model*.

However, this model is theoretically much more complex than the predicate calculus of the first order, and one should hardly hope for its practically acceptable efficiency in the industrial-level applications. In addition, in the *BDI* agent model, the interaction of agents is basically specified in the form of simple dialogues, and therefore, it does not have the dedicated means to formalize complex scenarios of agent behavior. This task is entrusted with logical models, and agents themselves must create models of other agents and of external environment to take decisions using minimum of interactions. In practice, it turned out to be impossible and stops software engineers in developing industrial applications.

It is worth to note that such an approach violates the core point of the multi-agent paradigm: *computations as interactions* [Luck et al., 2005].

The result of such active promotion of logical formalization of the *BDI-model* was completely predictable: just like the logical AI model, it generated many new mathematical problems and very interesting tasks for the specialists in the area of non-classical logical calculi, and what greatly contributed to the pure mathematics dealing with formal logics. However, it did not contribute to the efficient and effective formal modeling of agents and MAS in any way. Moreover, it significantly slowed down the practical use of MAS-technologies, in fact, for decades. Currently, there is an extensively developed logical theory of the agent *BDI-model*, which theoretically allows for building agents of a high level of intelligence that are capable of planning goal-oriented behavior and autonomous decision-making as well as distributed behavior coordination in rather complex situations. However, these abilities correspond to the level of existence theorems in mathematics, so when trying to use the logical model of the *BDI-agent* for relatively simple applications, serious and even unsolvable problems associated with computational complexity are met. A convincing example of such a situation is the *BDI-models* of collective behavior of robots, which have been actively funded by DARPA for almost a decade. In the late 1990's and until now, the agent

and MAS community evaluates them as the most influential achievements [Tambe, 1997],[Sycara and Sukthankar, 2006] and of course, they are, however, in pure mathematics but not in the area of Autonomous Agents and MAS (AAMAS). What concerns AAMAS it is the fact that all these results were demonstrated only by the authors and only by using toy applications. Since the beginning of the 2000's, information about novel development of these projects has not been available in literature, and both the projects have not been financed since then. One can say that the *logical model* of BDI-agents and MAS found out to be disruptive for agent teamwork model. At the same time, many other models of agents were also being developed, including simple reactive models, holonic models and some others [Muller and Fisher, 2013],[Leitao and Vrba, 2012].

4. *FIPA standards*. The constituting of standards in MAS-technologies became the topic of research as early as in the mid-1990's, when the public organization FIPA was established. It was aimed at the scientific justification of standards in MAS-technologies. However, FIPA was founded by the *iscientific, mostly academic community* that advocated the logical model of BDI-agent and MAS. Thus, the same model was accepted as basic one by FIPA in the development of standards.

For example, the standard communication language of agents *ACL (Agent Communication Language)* [ACL, 2020] uses a very complex language to describe the content of messages exchanged by agents. At its core, it is also specified in logical language, which began to develop in the 1970's, when it was positioned as a language for knowledge representation in the AI systems with predicate calculus as its basics. Its modern version, called *ACL*, is a fairly powerful and expressive language of an interpretative type that manipulates the concepts of ontology and can represent the content of messages exchanged by agents in a language close to the natural one. However, it brings into the standard all the features of the logical model of *BDI-agent* with all the ensuing consequences because of the computational complexity and communication channel overload problems. On the other hand, it is difficult to understand and use by application developers. For example, the freedom for developers to modify performatives and introduce new ones is mainly used only by the research community experts.

In practice, in most cases, it is possible to get the same using considerably simpler specialized languages. An example of a specialized language is the message exchange language adopted in the RoboCup server [Rob, 2020]. It uses only the necessary and sufficient means providing the needed expressiveness and thus computational efficiency. A completely different, simpler, and more prag-

matic language is used in the MAS model that was first proposed in the tools of *Magenta* company advanced now by *Smart Solutions* Ltd. [Vittikh and Skobelev, 2003], where the concept of *scene (the model of situation based on ontology)* is actively used for the representation of the current state of real world and decision making is fully based on virtual market concept where agents continuously buy and sell their services.

A different model of messaging, as compared to the FIPA standard, is also used in the software tool *Cougaar* [Cou, 2020], which was developed in the USA within the DARPA project for its use in the military applications. In this tool, messaging is supported by the blackboard architecture, in which agents offer their services and search for the required services, as and when necessary. In this case, the blackboard significantly expands the set of available services due to web services via *UDDI*-protocol [Tec, 2020]. This feature is implemented by a special component of the blackboard, i.e., servlet. It supports communication with Internet objects via *http*-protocol, and this service is available for use by all the connected modules of the site.

The discussed models of MAS used in *Smart Solutions* and *Cougaar* technologies proved to be the more successful in terms of industrial developments, although they do not use the standard FIPA platform.

Another great disadvantage of FIPA-standards is that it completely ignores all the aspects associated with *parallel programming*, whereas for MAS, this aspect is a basic one.

After years of developments the FIPA standards are not ready to use in the industrial application and FIPA misses many of the particularities imposed by industrial environments [Leitao and Vrba, 2011].

On the other hand, existing raw standards in MAS are additional constraints of freedom that push away young software engineers.

5. *Lack of flexible industrial technologies for design and implementation of MAS-applications*. The paper [DeLoach, 2009] notes that developers of industrial multi-agent systems face a lot of agent methodologies and, in most cases, the lack of industrial level software tools for their support. This may be because the new methodologies are not sufficiently flexible and are difficult to exploit in a wide range of applications. In most tools, agents only serve for additional repackaging of software objects with the help of OOP, which provides some advantages, but considerably reduces the possibilities of the agent approach and not supporting collective decision making at any level. Therefore, an important task is the integration of existing development methodologies and tools for their support into a sin-

gle well-defined technology and providing models, methods and tools for collective decision making. One of the reasons for the negative trends in the industrial use of MAS technologies is not the best strategy accepted in design methodology and supporting software tools. Indeed, from the very beginning, MAS researchers paid great attention to the development of the design methodologies and software tools. During 2005–2010, such research and developments were carried out most intensively. The general opinion was that a good design methodology and a powerful software tool would help to largely automate the design of MAS-applications of the industrial level, and thus, by simplifying and streamlining the MAS application development – but semantics of the problem domain was completely kept outside of the development loop. To develop reliable and efficient semantic models of a product, its assembling process together with the semantic models of the supply and transportation do represent quite a demanding tasks, requiring a lot of highly skilled knowledge engineers.

Before 2010, more than a dozen of MAS design and development methodologies were developed, besides many other less significant methods and methodologies. Among the most promising and well-developed methodologies, *Gaia* [Zambonelli et al., 2003], *Tropos* [Mylopoulos and Castro, 2000], *MaSE* [DeLoach, 2001], *ADELFE* [Beron et al., 2002], *MESSAGE* [Caire et al., 2002], *Prometheus* [Padgham and Winikoff, 2002] and *SADDE* [Sierra et al., 2004] can be mentioned along with several others. Most of these methodologies were also supported by sophisticated software tools (see, for example, [Luck and Gomez-Sanz, 2009]). They used either the author's development methodology or one of the listed above. For example, by that time tools such as *AgentTool* [Garcia-Ojeda and DeLoach, 2009], *Zeus* [Hyacinth et al., 1999], *AgentBuilder* [Age, 2020], *PASSI* [Burratato and Cossentino, 2002], *MASDK* [Gorodetsky et al., 2009] and several others had been developed.

However, all these tools were focused on low-level software developments and not focused on decision making — but even under these conditions it required long-term efforts of large team of professionals, developers, programmers, and testers to develop a practically interesting application. At that, development and testing of a methodology and the software tool supporting it themselves usually took at least 10 years.

In general, the advanced methodologies of this period used the concept of *model-driven software engineering*. In methodologies of this type, the conceptual model of MAS application, formal models of its standard components (agent models, ontology, protocols, messages, etc.) as well as the components supporting interaction model and the architecture of

target software were described in some formal languages, with preference being given to graphical languages.

Such a formal language had to automatically maintain consistency of component models of different levels of abstraction as well as generation of protocols of agent interaction in the system. It was assumed that the formal model of the application, built in this way, will then be compiled into the code of the program in a high-level language (for example, in *C++*, *Java* code or in another language of the same level). After that, the system should be supplemented with components for which the code can only be written manually (this concerned “non-agent” components of the application under development). It was expected that the high-level code obtained in this way should be further compiled into an executable code.

It is not difficult to see that a software tool intended to support such a methodology for design and software implementation stages of MAS-application development is objectively too “heavy”, especially in those cases when the logical model of the BDI-agent and MAS is used. Typically, these methodologies and tools relied on FIPA standards. For this, as well as for several other reasons, it was not possible to achieve the desired efficiency of design methodology and software implementation of MAS applications as well as computational efficiency of the target application. As a result, such methodologies and supporting software tools, with some rare exceptions, did not justify itself. It was possible to ramp up to the experimental models only the individual software prototypes. A certain exception is the software tool Living Systems ® Technology Suite of the Whitestein Technology company [Rimassa et al., 2006], which, however, was not used for mass development of MAS applications because of its labor intensity. Besides, nothing is known about its commercial effectiveness.

The described strategy of MAS application development was actively promoted and practically used during 2005–2015. And even after the said period, in the literature, one can find information about several hundreds of MAS-applications developed during this period across various application domains. However, most of them have not reached a more mature development level than their software prototypes. The same fate has befallen most of the methodologies and software tools supporting them: the corresponding development activity is currently discontinued.

AAMAS series of conferences also pay very little attention to this issue, e.g., agent and MAS methodologies and tools are the very secondary topics of AAMAS 2018, although Panel sessions of the several latest AAMAS issues actively discussed why

the industrial community pays so little attention to the agent-based architectures and technologies.

6. *Positioning of MAS and MAS-technologies in various application domains.* As already noted, there are many classes of applications that were previously positioned as the MAS architecture and technology application domains. However, practice has shown that many of them have already been successfully implemented at the industry level using other technologies, while their multi-agent implementations are definitely worse, or these applications currently do not have any multi-agent implementations at all. The reason behind this is that, from the very beginning, MAS was viewed as a fairly universal IT paradigm and technology; however, practice has shown that this is not the case, and it would be very useful to narrow the scope of MAS applications rather than rely on its universality in the engineering aspects (it is not about methodology).

To date, the area in which MAS applications and technologies have undeniable advantages has not been determined and is only just emerging. The enumeration of classes of potential applications in [Luck et al., 2005] was based, mainly, on a too broad class of applications, and therefore, the forecast given in this roadmap has not been confirmed. This document, rather, gives a list of those application areas in which MAS can be used along with some other technologies. Consequently, more mature technologies proved to be more successful in most cases, which contributed to negative opinions about MAS-technologies and decreased the interest of the industrial community in their use.

Small communities widely use the holonic approach for interactions of low-level control sensors and state it like true MAS implementations. But in fact they mostly implement holonic agent architecture and build the clever agent layer on top of it. This leads astray many scientists from the right understanding of MAS.

On the other hand, a lot of developers make their applications using SOA, MAS or equal technologies but do not want to publish and declare their results as multi-agent solutions in order to avoid the need of imposing standards and restrictions assigned to the multi-agent approach. They simply do not want to lose freedom of creativity.

7. *Low maturity of guided self-organization models and methods that are non-deterministic.* Distributed large-scale self-organizing systems operating in non-deterministic environment are the most prominent areas of MAS architecture and technology, in which the latter practically have no competing technologies. However, self-organization as a new paradigm is still not well studied and weakly-developed algorithmically as applied to the aforementioned class of systems. Nondeterministic mathematical models of self-organization began to

be actively studied by the scientific community very recently; an example of which is the growing number of research papers of the International Federation of Automatic Control (IFAC) [IFA, 2020] devoted to consensus-based decision-making.

8. *Difficulty to compare and benchmark classical and multi-agent solutions in solving complex problems.* In industrial applications for solving complex problems, it is often needed to compare different solutions and provide clear answers to such questions, e.g. for resource management:

- What is the quality of solution — how far it is from global optimum (in comparison with traditional combinatorial approach)?
- How can you guarantee efficiency of the method?
- What can you say about the stability of results?
- How quickly does productivity degrade with increasing dimension of the task?

Practically, it is not always possible to answer these questions theoretically, and often the result can be justified only experimentally. It is also difficult to compare MAS with classical methods — just because it requires to reduce complexity to make results comparable — but still how to compare batch and real time solutions dependable from momentum of time?

However, the big step towards it is the appearance of new virtual market models and methods of self-organization for solving complex problems [Granichin et al., 2015], [Ye et al., 2016], [Gorodetskii, 2012a], [Gorodetskii, 2012b], [Wolf and Holvoet, 2007]. Very interesting fundamental results related to Sandholm's virtual market approach, to our knowledge, could be found in [Shoham and Leyton-Brown, 2009], [Easley and Kleinberg, 2010], where it is formally proven that the power of multi-iterative auctions equals linear programming and provides global optimum, specifically, for some kind of assignment problem. Many notable properties of such algorithms are identified in these papers as “intuitive, provably correct, naturally parallelizable, appropriate for deployment in distributed systems settings, and tend to be robust to perturbations of the problem specification”. It was also stated that “A generalization of the auction-like procedure can be applied in case of NP-hard problems like scheduling too. The price for this new level of complexity is that the generalized algorithm will not come with the same guarantees”. But taking into consideration the individual domain-specific preferences and constraints of the problem, it became possible to solve such kind of extremely complex problems. For example, in [Granichin et al., 2015] the problem of resource allocation is solved for grid-networks in quasi-linear way using MAS

with randomization. There is also an analogue between models of self-organizing schedules and the theory of complex adaptive systems with nonlinear thermodynamics, when “stable non-equilibrium” is formed in MAS and chains of changes close to “autocatalytic chains” are observed [Mayorov and Skobelev, 2015]. In all these cases the solution of complex problem is found as un-stable equilibrium which represents consensus between agents.

In practice, it has already been proved, experimentally in several cases, that very complex problems are being solved — providing reasonable results in real time (not global optimum, but with acceptable quality of solution) much faster in comparison with existing methods, and providing possibility to consider many individual preferences and constraints [Ye et al., 2016], [Gorodetskii, 2012a], [Gorodetskii, 2012b], [Wolf and Holvoet, 2007].

In this regard, MAS-technologies could be considered as the technology for solving such type of NP-hard complex problems of scheduling for large-scale manufacturing, transportation or logistics networks, which could be solved only in a distributed way – providing high scalability, performance, and reliability. In such problems, classical methods of optimization or heuristics are not able to provide solutions at all or its simplification produces not feasible solutions, whereas MAS-technologies are able to build at least acceptable feasible and reasonable solutions [Shoham and Leyton-Brown, 2009], [Easley and Kleinberg, 2010].

It can be said that it is time for MAS-technologies to adapt and narrow down the possible range of applications; for example, focusing on solving extremely complex problems of planning and scheduling, knowledge extraction and pattern recognition, design of sophisticated engineering objects and many others, which are critical for upcoming new digital economy [Rzevski and Skobelev, 2014].

## 5 Prospects for Development of MAS-technologies of Industrial Level

Despite the above-mentioned aspects of the history of MAS theory and technology development, it still quite fits into the classical scheme of new technology developing and it makes first steps from laboratories into the industry, as this process in practice is never linear. At the beginning of its development in the 1990s, MAS became a challenge in software industry and required great efforts of scientists and programmers for the first successful developments — let us recall the blue screens of Norton Commander in DOS and the first Internet modems.

Naturally, not all decisions turned out to be successful. It is not difficult to predict that the development process of this kind of new technology will continue to be highly non-linear, and that it would require repeated attempts to transform it and not all of them would be success-

ful before MAS-technology would become truly mature, productive and commercial.

As noted in [Rzevski and Skobelev, 2014], MAS’ developers would be facing serious technical, organizational, commercial and other problems due to the competition on the IT market. Examples of such problems, for example, in resource management, include:

1. difficult to estimate how far we are from “optimal”;
2. results depend on the history of events’ occurrence (non-Markov’s processes, pre-history sensibility, etc.);
3. “butterfly effect”: small input leading to an unexpected big output;
4. system reaction can be unexpectedly slowed down in case of transition from one attractor to another;
5. in case of system re-start, the result of scheduling can be different;
6. it is difficult to “roll back” the system decisions (irreversibility);
7. real time interaction with users becomes more sophisticated;
8. system may become too “nervous” making re-scheduling, and a lot of computations with insignificant effect on optimality;
9. system decision can be hardly explained to the user (loss of causticity of results).

The process of searching for solutions in MAS is distributed and non-deterministic by definition if it is based on self-organization, which, in practice, contradicts to the traditions of hierarchical management and, more importantly, the traditions of designing systems on the “top-to-bottom” principle, since self-organizing systems are developed more on the “bottom-top” basis [Gorodetskii, 2012b],[Wolf and Holvoet, 2007]. However, then the solution of problem at any stage can be harmonized from the top by well-known Kaufman principle — local interaction generate global structures which affect initial local interactions.

It is also possible to form hybrid models in which the initial solution is found by traditional batch methods, and then it is modified depending on the events processing in MAS.

The commercial and engineering issues of MAS development include the following:

1. The sales of innovative MAS developments require an in-depth involvement of the domain experts, and not just developers. Moreover, it takes longer periods of time (from 3 to 24 months).
2. Development of MAS-applications that are important for business (based on experience of large projects) requires increased effort and time, approximately 3–5 times more than what was expected at the beginning.
3. The development efforts design and implementation of the multi-agent system (“engine”) itself takes no more than 25% of the total time, while the rest is

spent on other issues associated with accounting, databases, web user interface, integration.

4. Development of the first version of MAS is the most labor-intensive in nature and takes from 3 to 6 months (minimum), even with experience in using MAS-technology.
5. Implementation of MAS is often more time-consuming than the design itself, since it requires the identification and learning of rules for decision-making and their verification. Besides, integration with existing information systems is also required. The approximate ratio of labor input (as %) for the main phases of MAS development project is, on average, the following: design — 10, development — 20, testing — 15, delivery, implementation and training — 35, and support — 20.
6. The developed system must “survive” under the conditions of permanent user errors, with incomplete data for design, and getting of “not-well” data.
7. Users should be able to manually re-work and finalize solutions, as there are always factors that cannot be considered by automatic decisions in the system.

Overcoming the negative trends in practical use of MAS would certainly require considerable effort, but if successful, the results will pay off all the investments, providing in practice the advantages of MAS-technologies. Essentially, active work is currently being carried out to eliminate the issues of MAS-technologies and overcome their imperfections.

Using these results multi-agent systems have all chances to reach the productivity plateau at the level of industry standards, which apparently cannot be expected earlier than 2020–2025 and among the most promising areas of their use one can anticipate the following:

1. *Aerospace industry* — collective self-organizing behavior of unmanned space and aerial vehicles, management of small satellite groups, simulators for pilots and air traffic controllers, space logistics, etc.
2. *B2B networks of manufacturing and transport enterprises*, strategic planning and operational production management, network logistics (transport, etc.) and GRID resources management, where agent solutions and technologies are already being used, but there is still a large field for the use of network of MAS-technologies [Bukhvalov and Gorodetsky, 2015],[Bukhvalov et al., 2013].
3. *Military applications* — there are many indirect signs and evidence of activity regarding industrial developments in this area, but this information is classified.
4. *Collective robotics, autonomous missions of robots* — this field is still at the stage of exploring opportunities; however, specialists from all over the world consider it as one of the most promising areas for multi-agent applications.
5. *Smart grids, virtual power stations* and other ap-

plications in the field of *power production*. These developments started 6–8 years ago, and there has been an increase in their activity by using MAS-technologies.

6. *Healthcare* — environmental solutions to support public health (*ambient assisted living, personal healthcare*, etc.).
7. Tasks in the field of Internet of things where it is necessary to develop distributed *security systems*, sensor networks, smart spaces, etc. [IFA, 2020],[Gorodetskii, 2012b],[Wolf and Holvoet, 2007].
8. *Mobile applications* — according to Gartner’s estimations, up to 40% of future mobile applications over the next 10 years will be built based on MAS-technologies.
9. Virtual (overlay) networks and *peer-to-peer* (p2p) applications (*Software-Defined Networking*) — this paradigm of interaction between distributed applications is currently in huge demand.
10. *Manufacturing and other distributed applications* — Shop floor scheduling for a large scale production is a classical NP-hard problem. Constraint programming working only when few operations should be scheduled and where only a few production operators/workers involved, As the production grows, the time to find a solution is growing to an extreme. Agent approach allows to solve this task separately by an autonomous units. Agent can autonomously validate input data, solve the given problem using various strategies, analyze quality of solution, and negotiate with other agents to consider eventual changes in the solution. All these operations may run in most cases in parallel on one or multiple machines [Klima et al., 2017].

However, even now there are several MAS-applications demonstrating their advantages in practice.

One can find various examples [Muller and Fisher, 2013],[Leitao and Vrba, 2012], where some of them have become successful at the industry level, exhibiting examples of best practice.

## 6 Conclusion

For a long time, practically from its appearance, MAS-technology was considered as a powerful technology that could successfully compete against other technologies regarding a wide class of distributed large-scale intelligent solutions on industrial level.

However, this early and current belief of MAS researchers is not shared by industrial community yet. In fact, such optimistic expectations regarding MAS technology were not justified so far. To date, however, a lot of MAS potential applications are successfully implemented using several other newer technologies, whereas the interest of industrial and investment communities in exploiting MAS technologies practically went down to

zero. Accordingly, it is important to understand the core factors and reasons of such a situation.

Clearly, the loss of MAS technology's credibility is the consequence of many of these factors. One of very serious negative factor is the underestimation of the complexity and labor-intensiveness of MAS-applications' development. Indeed, MAS technology puts forth several strong requirements for the qualification of MAS technology developers and users, and high mathematical qualifications for decision-making and discrete optimization, artificial intelligence and telecommunications, knowledge representation, object-oriented programming, including parallel and asynchronous programming, among others. These requirements are complicated by the fact that some of them are difficult, if possible at all, to fulfill due to the absence of some necessary scientific results.

Some serious internal problems of MAS technologies occur due to several theoretically incorrect beliefs of MAS community. One of them is the belief that basic model of *BDI* agent, grounded on logic calculus and expanded with temporal and modal operators, is capable of working efficiently. Indeed, this model is very attractive because it is sound in nature and possesses rich enough expressive capabilities. This belief resulted in significant contribution to pure mathematical logic but nothing relating to MAS technology.

Other important factors of cooling the initial optimism of the industrial community regarding quick MAS perspectives were the underestimated difficulties to develop efficient design methodology and supporting software tool. Additionally, the developed methodologies and software tool were found to be too expensive to use and complex for developers.

Simultaneously, the number of successful industrial applications of MAS continues to grow and cover new areas. Although the pace of this growth is still far from the desired, and from the forecasts made in this respect in the early 2000s, multi-agent models and technologies are still promising for a large number of modern applications. In particular, they are promising for those that are solving complex problems, have networking structure, comprise autonomous and interacting entities, operate in not fully structured and uncertain environment and are of very large scale.

Obviously, the main application area of new generation of MAS should be oriented towards the Internet of People, Internet of physical and abstract Things and Internet of Data and Documents ("Internet of Everything"), where each thing should sooner or later become a "smart agent", and also the user should get an agent-assistant for problem solving in real life. Moreover, each thing will have sooner or later sensors and execution devices to manage real object as well as a tools for decision making for coordinated behavior [Gorodetsky, 2017].

Flexible and reconfigurable manufacturing systems and the ability of products to control their manufacturing processes are the key ideas of Industry 4.0 [Jirkovsky

et al., 2018]. The ability of autonomous AI systems to autonomously react on events and make planning in combination with learning from own experience will become the key features of the future development in Industry 5.0.

The area of Industry 4.0 strongly stimulates the exploration of the cybernetic methods in management and control of physical devices and other physical elements connected with the production processes. The agentified virtual twins do represent a corner stone for the knowledge-based integration as the main concept of Industry 4.0. There is no other efficient technology how to control and manage complex system consisting of units operating in the physical world in real time. The Industry 4.0 allows the exploration the already existing modelling, control and simulation algorithms connected with individual devices with higher level cybernetics-based coordination, management a control principles to be used among the components of the complex production system. According to the AAAI report [Gil and Selman, 2019], the global manufacturing companies require as an ultimate solution to get manufacturing facilities with self-X capabilities (self-learnable, self-adjustable, self-reconfigurable) – without the orchestrated exploration of cybernetic and AI principles inside both the autonomous units operating locally and the integrated complex global systems, the required flexibility cannot be achieved.

Over the past few years we have been witnessing the next significant evolutionary step in the area of holonic and agent-based industrial automation systems. In the past it was the creation of agent communication standards by the FIPA consortium and appearance of the first compliant agent platforms like JADE or FIPA-OS. A wide acceptance of FIPA's Agent Communication Language ensured the syntactic interoperability among heterogeneous agent systems. The current trend, motivated by the advancements of semantic web technologies, is the application of semantics for explicit description and interpretation of knowledge in MAS [Vrba et al., 2011].

And sooner or later the amount of data generated in practical applications will overcome all possibilities of processing it. We need to start to explore knowledge instead of data. And once this is done, the agent approach will probably gain the new "golden era" operating in Internet of everything and being supported by digital platforms and eco-systems of smart services, including personal agents of family, jobs, projects, sport and entertainment, house and cars, food etc. This approach will provide the easiest way to manage the main areas of life. The user will adjust domain knowledge or agents in agent based programming tools specifying his demands, requirements and preferences but also what is able to provide to society. Agents will not only create and integrate plans in certain areas, but also will learn from experience, recognize patterns, etc.

Simultaneously, for MAS' productive development, it becomes necessary to change the basic paradigm of for-

malizing the agent model and the software infrastructure that supports its life cycle, its interaction with the external environment and cloud services to achieve computational efficiency in real time. In the new paradigm of MAS formalization, it is rational to build them as a set of simple agents with rich interaction component and extensive use of principles of guided self-organization — ultimately, for the creation of digital ecosystems (“systems of systems”) built as fully autonomous, competing and cooperating smart services.

New efforts of scientists and practitioners will be required to understand and accept these new principles in order to develop new generation of MAS technology tools based on the paradigm of self-organization and capable to solving complex problems in a distributed, more flexible and efficient way.

### Acknowledgements

The work was supported by the Ministry of Education and Science of the Russian Federation within the framework of contract agreement No. 14.574.21.0183 — the unique ID number is RFMEFI57417X0183.

### References

- (2020). Agent communication language. <http://fipa.org/specs/fipa00061/SC00061G.pdf>. Accessed: 2020-03-10.
- (2020). Agentbuilder — an integrated software toolkit that allows software developers to quickly develop intelligent software agents and agent-based applications. [www.agentbuilder.com](http://www.agentbuilder.com). Accessed: 2020-03-12.
- (2020). Cougaar agent architecture. <http://cougaar.org/wp/documentations/tutorials>. Accessed: 2020-02-27.
- (2020). Digital ecosystems. [https://en.wikipedia.org/wiki/Digital\\_ecosystem](https://en.wikipedia.org/wiki/Digital_ecosystem). Accessed: 2020-03-10.
- (2020). The first international workshop on the internet of agents (ioa). <http://www2.cs.siu.edu/salqithami/iaa>. Accessed: 2020-03-05.
- (2020). International federation of automatic control (ifac). <https://www.ifac-control.org/>. Accessed: 2020-02-25.
- (2020). Robocup soccer server. <http://dangermouse.brynmawr.edu/pyrobot/tars/rcsoccersim>. Accessed: 2020-02-28.
- (2020). Technology reports, universal description, discovery, and integration (uddi). <http://xml.coverpages.org/uddi.html>. Accessed: 2020-03-04.
- Aleksandrov, A., Fradkov, A., and Semenov, A. (2019). Delayed and switched control of formations on a line segment: Delays and switches do not matter. *IEEE Transactions on Automatic Control*, **65** (2), pp. 794–800.
- Bernon, C., Gleizes, M.-P., Peyruqueou, S., and Picard, G. (2002). Adelfe: A methodology for adaptive multi-agent systems engineering. In *ESAW 2002*, pp. 156–169.
- Brussel, H., Wyns, J., Valckenaers, P., and Bongaerts, L. (1998). Reference architecture for holonic manufacturing systems: Prosa. *Computer in Industry*, **37** (3), pp. 255–274.
- Bukhvalov, O. and Gorodetsky, V. (2015). P2p self-organizing agent system: Grid resource management case. In *DCAI 2015*, pp. 259–269.
- Bukhvalov, O., Gorodetsky, V., Karsaev, O., Kudryavtsev, G., and Samoylov, V. (2013). Privacy-preserved distributed coordination of production scheduling in b2b networks: A multi-agent approach. In *IFAC Proceedings Volumes*, pp. 2122–2134.
- Burrafato, P. and Cossentino, M. (2002). Designing a multi-agent solution for a bookstore with passive methodology. In *AOIS 02 at CAISE 02*, pp. 119–135.
- Caire, G., Leal, F., Chainho, P., Evans, R., Garijo, F., Gomez-Sanz, J., Kerney, J. P. P., Stark, J., and Masonet, P. (2002). Agent oriented analysis using message/uml. In *AOSE 2001*, pp. 119–135.
- DeLoach, S. (2001). Analysis and design using mase and agenttool. In *MAICS 2001*, pp. 1–7.
- DeLoach, S. (2009). Moving multi-agent systems from research to practice. *Int. J. Agent-Oriented Software Engineering*, **3** (4), pp. 378–382.
- Easley, D. and Kleinberg, J. (2010). Networks, crowds, and markets: Reasoning about a highly connected world. <http://www.cs.cornell.edu/home/kleinber/networks-book>. Accessed: 2020-03-08.
- Fradkov, A., editor (2015). *Problems of networks control*. Nauka, Moscow-Izhevsk.
- Garcia-Ojeda, J. and DeLoach, S. (2009). Agenttool process editor: Supporting the design of tailored agent-based processes. In *SAC '09*, pp. 707–714.
- Gil, Y. and Selman, B. (2019). A 20-year community roadmap for ai research in the us. <https://cra.org/ccc/visioning/visioning-activities/2018-activities/artificial-intelligence-roadmap/#roadmapdoc>. Accessed: 2020-03-08.
- Gorodetskii, V. (2012a). Self-organization and multi-agent systems: I. models of multi-agent self-organization. *Journal of Computer and Systems Sciences International*, **51** (2), pp. 256–281.
- Gorodetskii, V. (2012b). Self-organization and multi-agent systems: II. applications and the development technology. *Journal of Computer and Systems Sciences International*, **51** (3), pp. 391–409.
- Gorodetsky, V. (2017). Internet of agents: From set of autonomous agents to network object. In *IoA'17*, pp. 1–17.
- Gorodetsky, V., Karsaev, O., Samoylov, V., and Konushy, V. (2009). Support for analysis, design and implementation stages with masdk. In *AOSE 2008*, pp. 272–288.
- Gorodetsky, V., Samoylov, V., and Trotskii, D. (2015).



- The reference ontology of collective behavior of autonomous agents and its extensions. *Journal of Computer and Systems Sciences International*, **54** (5), pp. 765–782.
- Granichin, O., Volkovich, Z., and Toledano-Kitai, D., editors (2015). *Randomized Algorithms in Automatic Control and Data Mining*. Springer.
- Hall, K., Staron, R., and Vrba, P. (2005). Experience with holonic and agent-based control systems and their adoption by industry. In *HoloMAS 2005*, pp. 1–10.
- Hyacinth, S., Divine, T., and Jaron, C. L. C. (1999). Zeus: A toolkit for building distributed multi-agent systems. In *AGENTS '99*, pp. 360–361.
- Jirkovsky, V., Obitko, M., Kadera, P., and Marik, V. (2018). Towards plug&play cyber-physical system components. In *IEEE Transactions on Industrial Informatics*, pp. 1–8.
- Kaminka, G. (2007). Robots are agents, too! keynote lecture. In *AAMAS-2007*, pp. 488–495.
- King, A. (2020). A 20-year community roadmap for ai research in the us. <https://horizonmagazine.eu/article/plants-can-detect-insect-attacks-sniffing-each-others-aromas.html>. Accessed: 2020-03-12.
- Klima, M., Gregor, J., Harcuba, O., and Marik, V. (2017). Agent-based shop floor scheduling. *Springer*. In *Springer*, pp. 24–36.
- Laryukhin, V., Skobelev, P., Lakhin, O., Grachev, S., Yalovenko, V., and Yalovenko, O. (2019). The multi-agent approach for developing a cyber-physical system for managing precise farms with digital twins of plants. *Cybernetics and Physics*, **8** (4), pp. 257–261.
- Leitao, P. (2009). Agent-based distributed manufacturing control: A state-of-the-art survey. *Engineering App. of Artificial Intelligence*, **22** (7), pp. 979–991.
- Leitao, P., Marik, V., and Vrba, P. (2013). Past, present and future of industrial agent applications. *IEEE Tran. on Industrial Informatic*, **9** (4).
- Leitao, P. and Vrba, P. (2011). Recent developments and future trends of industrial agents. In *HoloMAS-2011*, pp. 15–28.
- Leitao, P. and Vrba, P. (2012). Recent developments and future trends of industrial agents. In *HoloMAS 2011*, pp. 15–28.
- Luck, M. and Gomez-Sanz, J., editors (2009). *Agent-oriented Software Engineering IX*. Springer.
- Luck, M., McBurney, P., Shehory, O., Willmott, S., and eds. (2005). Agent technology, computing as interaction: A roadmap for agent based computing. <http://www.agentlink.org/roadmap>. Accessed: 2020-03-02.
- Marik, V., Fletcher, M., and Pechoucek, M. (2002). Holons & agents: Recent developments and mutual impacts. In *ACAI 2001*, pp. 27–35.
- Marik, V. and McFarlane, D. (2005). Industrial adoption of agent-based technologies. *IEEE Intelligent Systems*, **20** (1), pp. 27–35.
- Mayorov, I. and Skobelev, P. (2015). Toward thermodynamics of real time scheduling. *International J. of Design & Nature and Ecodynamics*, **10** (3), pp. 21–223.
- Mendes, M., Leitao, P., Restivo, F., and Colombo, A. (2009). Service-oriented agents for collaborative industrial automation and production systems. In *HoloMAS 2009*, pp. 13–24.
- Muller, J. and Fisher, K. (2013). Application impact of multi-agent systems and technologies: A survey. In *Agent-Oriented Software Engineering*, Springer, Springer, pp. 1–26.
- Mylopoulos, J. and Castro, J. (2000). Tropos: A framework for requirements-driven software development. In *CAiSE 2000*, pp. 108–123.
- Padgham, L. and Winikoff, M. (2002). Prometheus: A paradigm methodology for engineering intelligent agents. In *OOPSLA 2002*, pp. 97–108.
- Pechoucek, M. and Marik, V. (2008). Industrial deployment of multi-agent technologies: Review and selected case studies. *Autonomous Agents and Multi-Agent Systems*, **17** (13), pp. 397–431.
- Plotnikov, S. and Fradkov, A. (2019). On synchronization in heterogeneous fitzhugh–nagumo network. *Chaos, Solitons and Fractals*, **121**, pp. 85–91.
- Plotnikov, S., Lipkovich, M., Semenov, D., and Fradkov, A. (2019). Artificial intelligence-based neurofeedback. *Cybernetics and Physics*, **8** (4), pp. 287–291.
- Rimassa, G., Greenwood, D., and Kernland, M. (2006). The living systems technology suite: An autonomous middleware for autonomic computing. In *ICAS'06*, pp. 33–38.
- Rzevski, G. and Skobelev, P. (2014). *Managing Complexity*. WIT Press, London-Boston.
- Shoham, Y. and Leyton-Brown, K., editors (2009). *Multi-agent systems: Algorithmic, Game Theoretic and Logical Foundations*. Cambridge Univ. Press.
- Sierra, C., Sabater, J., Augusti, J., and Garcia, P. (2004). Sadde: Social agents design driven by equations. In *Methodologies and Software Engineering for Agent Systems: The Agent-Oriented Software Engineering Handbook*, pp. 1–24.
- Skobelev, P. (2015). Multi-agent systems for real time adaptive resource management. In *Industrial Agents: Emerging Applications of Software Agents in Industry*, pp. 207–230.
- Skobelev, P. (2018). Towards autonomous ai systems for resource management: Applications in industry and lessons learned. In *PAAMS 2018*, pp. 12–25.
- Strassner, J. (2007). Using agents and autonomic computing to build next generation seamless mobility services. In *AAMAS-2007*, pp. 1–16.
- Sycara, K. and Sukthankar, G. (2006). Literature review of teamwork models. cmu-ri-tr-06-50. [http://www.ri.cmu.edu/pub\\_files/pub4/sycara\\_katia\\_2006/sycara\\_katia\\_2006.pdf](http://www.ri.cmu.edu/pub_files/pub4/sycara_katia_2006/sycara_katia_2006.pdf). Accessed: 2020-02-25.
- Tambe, M. (1997). Towards flexible teamwork. *Journal of Artificial Intelligence Research*, (7), pp. 83–124.
- Vittikh, V. and Skobelev, P. (2003). Multi-agent interac-

- tion models for constructing the needs-and-means networks in open systems. *Automation and Remote Control*, **64** (1), pp. 162–169.
- Vrba, P., Tichy, P., Marik, V., Hall, K. H., Staron, R. J., Maturana, F. P., and Kadera, P. (2011). Rockwell automation’s holonic and multiagent control systems compendium. *Transactions on Systems, Man, and Cybernetics*, **41** (1), pp. 14–30.
- Vyatkin, V. (2011). Iec 61499 as enabler of distributed and intelligent automation: State-of-the-art review. In *IEEE Trans. on Ind. Informatics*, pp. 768–781.
- Wahlster, W. (2019). Artificial intelligence solutions for industry 4.0. In *The Fourth Sino-German Workshop on Digital Transformation of Manufacturing Industry, Artificial Intelligence and Digital Manufacturing*, pp. 12–23.
- Wolf, T. D. and Holvoet, T. (2007). Design patterns for decentralized coordination in self-organizing emergent systems. In *ESOA 2006*, pp. 28–49.
- Wooldridge, M., editor (2009). *An Introduction to Multi-Agent System*. John Wiley & Sons.
- Ye, D., Zhang, M., and Vasilakos, A. (2016). A survey of self-organization mechanisms in multiagent systems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, **47** (3), pp. 1–21.
- Zambonelli, F., Jennings, N., and Wooldridge, M. (2003). Developing multi-agent systems: The gaia methodology. *ACM Trans. on Software Engineering and Methodology*, **12** (3), pp. 417–470.