2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)
October 11-14, 2020. Toronto, Canada

Multi-Agent Technology for Industrial Applications: Barriers and Trends

Vladimir Marik
CIIRC, Czech Technical University
Prague, Czech Republic
vladimir.marik@cvut.cz

Vladimir Gorodetsky
St. Petersburg State Electrotechnical University
St. Petersburg, Russia
vladim.gorodetsky@gmail.com

Petr Skobelev
Institute of Automation and Information Technology
Samara State Technical University
Samara, Russia
petr.skobelev@gmail.com

Abstract—**Multi-agent systems (MAS) have been an area of high expectations of the industrial IT community. However, in reality, these expectations are still not met and, in practice, the industry very rarely uses the MAS design methodologies, technologies, and software tools despite the appearance of many new classes of applications for which the MAS paradigm could be the perfect match. This paper analyzes the barriers and trends of the mismatch between the recent industrial anticipations and the real state of the practical use of MAS. It identifies engineering problems with very little re-use of code that currently stops economics of scale and impedes the extensive industrial MAS deployment and the ways to overcome them.**

Keywords— Multi-agent systems, smart system, industrial applications, decision making, agent interaction, self-organization, eco-system of smart services, systems of systems

I. Introduction

Agent is an autonomous software program (system) that is capable to goal-directed proactive behavior in dynamic unpredictable environments without external intervention [1].

Multi-agent system (MAS) is a network of weakly coupled software agents solving particular problems, situated in common environments and interacting with each other to compete or cooperate and to coordinate their behavior in order to achieve their common goal or their particular ones.

Interaction is the key feature of MAS in addition to autonomy. "From interaction and autonomy comes … emergence" [1]. Moreover, in the EU roadmap [2] MAS is described as a paradigm of computations as interactions.

Both the autonomous agents and the MAS models were proposed near to mid-1980's, and quickly attracted interest of the academic and industry IT-community. New features include modularity of the conceptual models, organic decomposition of the design process and simplification of the complex system software engineering. Multi-agent technology considered as a set of methodologies, frameworks, architectures, design patterns and software development tools for building MAS applications.

For over a quarter of a century, MAS have been considered as one of the most promising technologies for conceptualization, software development and implementation of distributed Artificial Intelligence (AI) solutions. The world's leading rating agencies have included MAS-systems and technologies into the list of the topmost promising information technologies (IT). For example, MAS is one of new technologies mentioned in TOP-10 by Gartner in 2019, with the view on"Autonomous Things" (https://www.gartner.com/smarterwithgartner/gartner-top-10-strategic-technology-trends-for-2019).

But the real situation is different: the IT industry is holding off on using this technology whereas the world community of MAS scientists actively continues to develop new agent-based solutions. At the same time, scientific activity concerning a novel theoretical basis and methodologies for agent technology remarkably decreased. As a result, in practice, "classic" logic-based specification languages and the BDI-models of autonomous agents [3] and corresponding technologies remain to be the key means at disposal of designers.

However, the application landscape has crucially changed last decades and progress in MAS technology weakly matched the modern application requirements and thus has remained unnoticed by the IT-industry.

The first signals of high complexity of developing industrial MAS solutions for real-life applications have been clearly discovered in last decade [4–7]. It become clear that MAS require mentality shift in distributed thinking vs centralized, big investments and maturity of industry. One of the detailed overviews on industrial MAS applications was undertaken in 2011 [7]. 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 include real-time constraints, integration with the physical hardware and legacy systems and lack of MAS tools.

One of the most extensive surveys on MAS case studies in the industry was made for 152 MAS in different areas [8]. It covers industrial MAS maturity, vertical sectors, and the usage of programming languages and platforms.

More advanced engineering study on industrial MAS was made in 2013 [9]. 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". Secondly, important engineering trends and issues to be solved were identified: required convergence of

The work was supported by the Ministry of Education and Science of the RF in the framework of contract agreement № 075-15-2019-1691 – ID is RFMEFI60419X0224.

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, learning, etc.

The number of industrial MAS solutions and applications for solving extremely complex problems of adaptive resource management is presented in 2014 [10] and 2018 [11]. These publications provide a first outlook on the barriers for developing industrial MAS from an engineering viewpoint. The main idea is that massive application of MAS-technologies in industry will require full reconsideration of basic paradigm in formalization of agent's and MAS models (e.g. BDI model) as well as the engineering and technological aspects of their design and implementation. Instead of following the formal logic, industrial MAS needs to be driven by engineering models and methods of self-organization with high re-use of code.

The paper [11] contains a comparative analysis of the expectations regarding the industrial applications of MAS-technologies and their practical use. It analyzes the causes that make the introduction of MAS-technologies into practice difficult and proposes the ways to overcome these problems. In the last section, modern application areas are identified for new industrial MAS developments.

II. Practical use of MAS technologies: Expectations and state-of-the-art

A. Expectations

The concept of MAS was proposed in the mid-1980's. Within the first two decades, its basic theoretical foundations were built, and active development in the field began. In 1996, FIPA (*Foundation for Intelligent Physical Agents*) 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 by academic and industry communities that MAS were about ready to capture the leadership as a principally new design paradigm for industrial distributed systems.

This concept was very attractive for designing complex systems in analogy with the living systems and human society, particularly, through autonomy, interactions and underlying self-organization. Since the very beginning and up to the present days, the ideas of MAS were especially attractive for distributed networking systems like collective robotics [12], IoT [13], etc.

B. State-of-the-art

However, in the beginning of the 2000's, something went wrong. The perception of MAS by industrial players reduced [6]. Large-scale developments by Apple, Facebook, Google, and SAP were not related to MAS, at least, in the public perception.

S.A. DeLoach who is behind the *O-MaSE* methodology and *agentTool* [14], clearly indicated the absence of the expected progress in the industrial applications of MAS. According to his opinion, despite more than twenty years, this area is still at an early stage of maturity compared to, for example, the object-oriented approach (OOP). He also noted several gaps in the field of MAS technology, which still need to demonstrate the ability to deliver self-organized systems. A comparison was made between the ages of MAS-technologies and OOP: The OOP language *C++* was created 32 years and *JAVA* 39 years after the birth of the OOP concept. Now, the age of MAS-technologies is approaching 40 years, but very little new has happened so far.

The authors [1] complain about the weaknesses of methodologies for designing MAS but by that time (2005), several well-developed MAS methodologies were created and tested; *Gaia* [15], *Tropos* [16], *MaSE* [17], *ADELFE* [18], *MESSAGE* [19], *Prometheus* [20]. The issue was that these methodologies were not supported by software tools. The intensity of designing MAS methodologies and supporting software tools was quite high almost until 2010. However, all these methodologies as of 2005 were far from industry.

As a response to this issue, the IEEE-FIPA Design Process Documentation and Fragmentation (FIPA-DPDF) Working Group was established in 2010 [21]. It was aimed "at providing the possibility of representing design processes and method fragments through the use of standardized templates, thus allowing the creation of easily sharable repositories and enabling an easier composition of new design processes" [22]. No doubt, that the contribution of this group was significant. According to our views, the main contribution of this FIPA-IEEE Group is the so-called Situational Method Engineering (SME) that is based on best practice of Object-Oriented Design. The main idea of SME is to combine different methodologies, formalize and structure them as a database with the meta-model of methodology on the top. In modern semantic technology, it, actually, is an ontology-based approach. Several instances of meta-models can be found, for instance, in [23, 24].

However, [1] most claimed the SME idea, but not brought to the level of industrial maturity. Indeed, each particular SME-based methodology presented in [1] uses only design fragments of own methodology but the methodology design fragments themselves remained as of 2010, thus preserving all the drawbacks of the previous steps of their developments. It would be of worth to notice that not all of these methodologies and corresponding software tools if any are practically supported nowadays. Another problem remaining in SME is that no inference mechanisms are proposed so far to assist the developers in design fragments search and form the usage scenario. Along with BDI agent-oriented methodologies and technologies, several other lines exist today. Among leaders of industrial MAS developments is Rockwell Automation, Inc., which made indispensable investments into the MAS research [25-26] and spent nearly two decades in making pioneering development in the application of holonic MAS for industrial automation. They went through a path from the first simple prototypes of holonic control systems (holoblocs) to the development of a comprehensive bundle of advanced methodologies, practices, and tools, that cover all aspects of MAS design and implementation [26].

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, 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. The results showed even in 2011 [26], the necessity of providing customers with the analysis of the benefits of agent technology compared to "classical" technologies.

Probably the most unexpected fact recognized after 2010, was that the speed of new MAS application developments began to slow down gradually – reflecting recognized complexity of MAS solutions. It was shown that many promising MAS mentioned in [2] finally were converted into service-oriented technologies, grid computing, ubiquitous and cloud computing.

The paper [8] became the first one to clearly state the alarming prospects of MAS-technologies in competition with other IT technologies. The MAS-applications developed until 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 [8]: industrial systems or those close to them – 46 out of 152; research software prototypes of the industrial level, which were tested on real data, but not used in real work – 55; laboratory research prototypes, which were used for educational, scientific, and other purposes – 46. It was found that less than one-third of the analyzed applications turned out to be sufficiently mature, and their total number was five times less than the roadmap forecast [2]. In addition, about half of them were developed by university teams, who were more interested in research rather than in the industry solutions.

The next step of agent development was declared in relation with the service-oriented architectures (SOA) [27]. SOA as a style of software design is not being considered as a product or a follower of MAS. However, there are many parallels between the agents and web services [26]. Holonic multi-agent system, introduced in a legendary paper [28], 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. [29]) which can recursively form holarchies as bottom-up structures (compared to the top-down hierarchies). The holonic MAS solutions were mostly used for control of near-to-physical layer devices in manufacturing exploring the IEC 16499 standard [30], later were applied for solving complex problems of resource management [11]. The holonic MAS put focus on self-organization based on specific classes and role models of agents and protocols of their interactions [31].

The new trend is the application of semantic technologies (ontologies) to enhance the capabilities of MAS for representing and exchanging knowledge, and thus to increase the MAS openness, intelligence, and flexibility [25].

The conclusions of this brief overview are the following:

1. MAS-technologies are developing at a much slower pace than expected. The majority of MAS research and development projects are carried out, mainly, within the scientific community.

2. Agents and holons have provided a new abstraction metaphor for designing intelligent systems featuring new properties: autonomy, interaction, survivability, robustness, adaptation, learning, and self-organization. However, the decision makers in industry are still reluctant to take the risk of being the first adopters [32]. Thus, "classical" centralized and hierarchical rigid architectures are still predominantly used in industry [26]. The interest of industrial community regarding MAS has decreased.

3. However, simultaneously, high latent turbulence, regrouping and restructuring of the AI market have been observed. Thus, the MAS received a lot of new opportunities and challenges, which can be offered to industry.

4. The number of new classes of applications have been also increasing, and they have great prospects for MAS: distributed computations, green energy, autonomous vehicles, swarms, etc.

Therefore, it is vital to understand the engineering reasons and barriers that are stop-factors for the practical exploration of the benefits of MAS technologies in industry.

III. WHAT ARE THE BARRIERS FOR THE PRACTICAL USE OF THE GREAT POTENTIAL OF MAS?

One of the most valuable things that MAS has offered for industry is conceptualization of complex systems. This is valid for systems of almost any complexity, which attracts the attention of researchers and developers. But complexity of the MAS formal design and modeling created a barrier for efficient engineering implementations as well as for maintaining solutions on the client side.

The last decades show that the MAS are difficult to develop, understand and use for practitioners – they require knowledge and skills in object-oriented programming, AI models, semantic knowledge representation, methods for collective decision-making support, parallel programming, telecommunications, machine learning, etc.

However, the problems were much deeper – let us outline the main engineering problems and obstacles in this respect:

1. *Lack of generally accepted understanding of MAS' key concepts.* This is indicated particularly in [14]. The absence of definitions and non-doubtful agreements on the concepts of MAS greatly hinders the mutual understanding between researchers and developers. For example, majority of programmers agree with the definitions of basic OOP concepts, such as classes, objects, inheritance, and encapsulation. At the same time, the MAS developers have different understandings of agent, role, negotiations, plan and others. One of the ways to overcome this problem is *to develop basic ontology for domain-independent concepts* used in MAS. An example of the ontology for domain-independent behavioral concepts of the BDI-model can be found in [33].

Authors of [23] attracted attention to this issue were, perhaps, the first who proposed a short version of the agent-oriented list of key concepts with their definitions (Tab. 1).

TABLE I. AGENT ORIENTED ONTOLOGY CONCEPTS (THE SOURCE – [23])

Entity	Definition
Goal	A desirable state; goals capture organizational objectives
Role	Capture behavior that achieves a particular goal or set of goals
Agent	Autonomous entities that perceive and act upon their environment; agents

	play roles in the organization
Organizational Agent	A sub-organization that functions as an agent in a higher-level organization
Capability	Soft abilities (algorithms) or hard abilities of agents
Domain model	Captures the environment including objects and general properties describing how objects behave and interact
Policy	Constrain organization behavior often in the form of liveness and safety properties
Protocol	Define interaction between agents, roles, or external Actors; they may be internal or external
Actor	Actors that exist outside the system and interact with the system
Plan	Abstractions of algorithms used by agents; plans are specified in terms of actions with the environment and messages in protocols

2. *Lack of generally accepted notations for specification of MAS models*. 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 [14]. Standardization of MAS models and means for their specification is necessary but still not realistic.

Let us remind that SME mentioned above requires the solution of this task and several authors of the Handbook [1] proposed their own versions of the notation in question. However, this list of key concepts has no approval of the standardization body so far.

3. *Conceptual and computational complexity of logical formalization of the BDI-model of agents and MAS*. For almost three decades, the main efforts of MAS researchers has been aimed at developing models of *intelligent* agents with its own knowledge base, or at least, with a knowledge model, capability of goal selection and planning the goal-oriented behavior in an unpredictable environment. MAS' specialists were competing to provide the agent with the most intelligent abilities in terms of autonomous behavior, and the ability to understand the intentions of other agents. Thus, it led to constant over-sophistication of the agent and MAS formal models.

From the very beginning, the *BDI*-model (*Belief-Desire-Intention*) [3, 34] 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 both the agent and the MAS and their logical formalization. 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 be understood for the application developers and is a significant barrier in the interpretation of the basic concepts of MAS [14], including the interpretation of the term "*BDI"* itself.

However, this formalization of BDI 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. It is of worth to note that such an approach violates the core point of the multi-agent paradigm: *computations as interactions* [1]. A convincing example of such a situation is the *BDI*-model of collective behavior of agents, which have been actively funded by DARPA for almost a decade [35]. Since the beginning of the 2000's, the logical models of *BDI*-agents found out to be disruptive for agent teamwork model and DARPA stopped funding these researches.

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. However, FIPA was founded by the *scientific, mostly academic community,* that advocated the logical model of *BDI*-agent and MAS. Thus, the same model was accepted as a basic one by FIPA in the development of standards.

For example, the standard communication language of agents *ACL* (*Agent Communication Language*) [36] uses a very complex language to describe the content of messages exchanged by agents. *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.

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 [37]. It uses only the necessary and sufficient means providing the needed expressiveness and thus achieving computational efficiency. The concept, methodology and language of agents negotiations on the virtual market of demand-resource networks was introduced in [10, 38]. This pragmatic approach, when agents buy and sell their services and solve complex problems by competing and cooperating on virtual market was used in the MAS developments of *Magenta* company and now is significantly advanced by *Knowledge Genesis Group* and it leading company *Smart Solutions, Ltd*. The concept of *scene as the model of situation based on ontology* is actively used here for the representation of the current state of world and decision-making is fully based on self-organization.

A different model of messaging, as compared to the FIPA standard, is also used in *Cougaar* [39], which was developed within the DARPA project. 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. The discussed models of MAS used in *Smart Solutions* and *Cougaar* technologies proved to be more successful in terms of industrial developments, although they do not use the standard FIPA platform. Another disadvantage of FIPA-standards is that they completely ignore 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 be used in the industry [6].

5. *Lack of flexible industrial technologies for design and implementation of MAS-applications*. This paper [14] notes many agent methodologies but the lack of industrial software tools for them. 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 re-wrapping of software objects with the help of OOP, which provides some advantages, but considerably reduces the agent possibilities and does not provide any models,

methods, and tools for collective and coordinated decision making. The aforementioned SME technology assuming the flexible construction of application-oriented usage design scenario is a perspective way to make agent technology attractive for industry is however still far from maturity.

6. *Positioning of MAS in various application domains.* As already noted, there are many classes of MAS solutions that were positioned as very promising for many applications. However, in practice these MAS were implemented at the industry level using other technologies. 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. To date, the area in which MAS applications and technologies have undeniable advantages has not been determined and is only just emerging.

7. *Low maturity of guided self-organization models and methods that are non-deterministic by nature*. 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, learning and self-organization as new paradigms are still not well studied and developed algorithmically. 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 IFAC devoted to consensus-based decision-making [40].

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, i.e. for resource management: What is the quality of the solution – how far it is from global optimum (in comparison with traditional combinatorial approaches)? 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, because it requires reducing complexity to make results comparable. But still how to compare batch and real time solutions dependable from the 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 [41 – 43]. Very interesting fundamental theoretical results related to Sandholm's contract-net protocols, to our knowledge, could be found in [44], 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". Taking into consideration the individual domain-specific preferences and constraints of the problem, it became possible to solve even NP-hard problems but with no guarantee of optimum yet. For example, in [45] the problem of resource allocation is solved for grid-networks in a 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 [46]. In all these cases, the solution of a complex problem is found as unstable equilibrium, which represents competitive equilibrium and consensus between agents, as a result of emergent intelligence.

In practice, as it has already been proven in several cases, the very complex problems are being solved by MAS and provide reasonable results in real time. In these most cases MAS finds not global optimum, but provide acceptable quality of solution, doing it much faster in comparison to existing methods [44-46]. MAS-technologies prove these results in scheduling resources for large-scale manufacturing, transportation or logistics networks, which could be solved only in a distributed way – providing high scalability, performance, and reliability [47]. It can be said that it is time for MAS-technologies to adapt and narrow down the possible range of such applications.

IV. 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 development and makes the first steps from laboratories into industry, as this processing practice is never linear. As noted in [10], 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:

- difficult to estimate how far is the process from being "optimal";
- results depend on the history of events' occurrence (non-Markov's processes, pre-history sensibility, etc.);
- "butterfly effect": small input leading to an unexpectedly dramatic change of output;
- system reaction can be unexpectedly slowed down in case of transition from one attractor to another;
- in case of system re-start, the result of scheduling can be different;
- it is difficult to "roll back" the system decisions (irreversibility);
- system decisions can hardly be explained to the user.

Our own practice proved the following issues of MAS developments:

- The innovative MAS development processes do require an in-depth involvement of the domain experts and they take long periods (from 3 to 24 months).
- MAS solutions require increased effort and time, usually 3–5 times more than what was expected.

- The development efforts for MAS takes no more than 25% of the total time, while the rest is spent on other issues (system integration, human/machine interface building, data exchange problems).
- The approximate ratio of labor input (as %) for the main phases of MAS development project is, at average, the following: design – 10, development – 20, testing – 15, delivery, implementation – 35 and support – 20.
- The developed system must "survive" under the conditions of permanent user errors, with incomplete data for design, and getting of "incorrect" data, resistance of the end-users, etc.
- Users should be able to manually intervene MAS and interactively re-work and finalize solutions.

Overcoming the negative trends in practical use of MAS would certainly require considerable effort, but if successful, the results will pay off all the investment, providing in practice the advantages of MAS-technologies.

V. NEW TRENDS FOR MAS TECHNOLOGY FOR INDUSTRY

The landscape of modern industrial information technologies is now rapidly changing and currently it differs crucially compared to the time when AgentLink III published the MAS roadmap [2]. Indeed, the information and computing world is becoming highly populated with networked distributed systems, including autonomous and mobile objects of various nature (physical, virtual, social) composing together large-scale systems with sophisticated collective behavior. These applications gave rise to many novel frameworks, like SOA, Internet of Things, digital twins in the Industry 4.0 notation, etc.

Accordingly, the MAS theoretical landscape has to be changed in many respects. It should remarkably shift to methods and tools for collective intelligence, standards for data representation and interaction protocols, distributed coordination, autonomous real-time resource planning, self-organizing networks, software and communication platforms.

The corresponding industrial implementations based on the generic frameworks are becoming extremely cumbersome without exploration of well-structured knowledge ontologies introduced by MAS technologies. Also, such paradigms like software as a service, collective behavior, swarm and group control, scenario representation, situational awareness and activity synchronization, dynamic routing, etc. can be explained as a specific MAS feature. The future MAS solutions need to handle in a new way robots and humans, enterprises and their networks, Internet of Things with the following features:

- the tremendous number of the MAS nodes interacting as autonomous agents, sharing the goals and tasks, etc;
- service-oriented architectures with autonomous MAS and new conceptual model of agents [48];
- semantic inter-operability of autonomous MAS based on ontological knowledge stored in a specialized knowledge structures"owned" by semantic agents [49];
- sharable networks of resources - the key problem here is on-line adaptive resource planning and scheduling and resolution of conflicts between MAS;
- global behavior of MAS as a complex systems, to be achieved, requires the bottom-up design strategy, whereas up-to-date information technologies and tools mostly use the top-down approach; an exclusion is ADELFE 2.0 methodology and tool [1];
- multi-task mode of MAS operation: in the same software and communication environment, many tasks can run concurrently and, thus, they can cooperate and compete for the same services and resources;
- cooperative groups: in each particular application in multi-task mode, a subset of tightly cooperating nodes can be defined – in such a way a local MAS community is created and represented to outstanding agents as a specific MAS node with MAS principles used inside (holonic principle).

There exist driving technologies supporting MAS re-entering the industrial scene, e.g. Industry 4.0 where the digital twins can be easily enhanced into the form of agents (by a simple agentification process). The needs in communication and negotiations among these twins/agents can lead to MAS conceptual solutions quite easily [50].

AI brings quite novel opportunities and challenges for development of MAS solutions. The self-* properties of complex production systems (e.g. self-learning, self-sustainability, self-awareness, self-reconfigurability...) are strongly required by industrial top managements as reported in the recent AAAI Report from August 2019 [51].

Fault-tolerant behavior is also highly appreciated [52].

VI. CONCLUSION

To summarize results, multi-agent technology found out too complex and sophisticated: it should integrate technologies from multiple, sometimes, very different areas like complex systems, software engineering, artificial intelligence, distributed decision making and communication technologies, etc. One of the fundamental barriers for MAS engineering is luck of battle-proved models, methods, tools and algorithms for solving complex problems in distributed and non-deterministic way based on bio-inspired methods of self-organization and evolution. Additionally, the developed MAS methodologies and software development tools for MAS designing were found to be not efficient for developers. As a result, to date, many MAS potential applications are successfully implemented using other technologies, which are not always fully utilize the potential power and benefits of MAS. As a result, engineering difficulties and barriers of MAS implementation have been significantly underestimated by both research and industry communities.

The paper contribution is the thorough analysis of the basic reasons of such state-of-the-art of the industrial applications of modern MAS technologies forming the barriers against wide industrial use of the MAS technologies. A number of papers published after 2010, (see, e.g., [4 – 9, 17, 52, 53, 54, 55, 56]) indicated and validated different drawbacks of the existing MAS development methodologies, their algorithmic implementations, and supporting MAS platforms and software tools based on qualitative and quantitative assessments and the authors' own experience. This paper integrates together the aforementioned assessments and adds some engineering findings resulting from their own long-term (more than 20 years) practical experience in development of algorithmic support of MAS methodologies and software tools with the subsequent exploiting of them for the designs, implementations, and deployments of many particular MAS-applications in various areas. Most part of such applications (more than decades of them) corresponds not only to the research software prototypes but also to the full-scale industrial applications dealing with real businesses. They cover such applications as manufacturing, supply chains, transportation logistics, air traffic control, B2B agent networks, computer networks and information security, swarm satellites planning, project management and others.

The generalized and justified list of barriers preventing the wide industrial-level use of modern MAS technologies includes

- Lack of generally accepted understanding of MAS' key concepts and design principles;
- Overcomplicated and not efficient notations for specification of MAS design and behavior models;
- Conceptual and computational complexity of logical formalization of the BDI-model of agents and MAS;
- Too many simplifications and limitations in initial FIPA standards;
- Low level of MAS software code generalization and re-use for designing solutions and their customization;
- Industrial MAS developments requires documented design-patterns of most efficient MAS architectures;
- MAS solutions require a number of software components to support full scale implementation and delivery, including data storages, user interfaces, etc.;
- Lack of methodologies and platforms, technologies and software development tools for industrial MAS design, implementation, testing and integration;
- No ready-to-use collections of agent models, methods of decision making and protocols of agents interaction which can be easily combined in deployed MAS solutions;
- Issues with end-users interactions to capture domain-specific knowledge and cooperation in the process of decision making;
- Difficulty to compare and benchmark classical and multi-agent solutions in solving complex problems.

The paper contribution also outlines ways to overcome the aforementioned barriers thus determining the future works. New generation of MAS solutions will be designed as a digital eco-systems of smart services, open for third party developers and based on DevOps platforms for full-scale developments, testing and deployment of agents.

The design of industrial solutions should ensure a number of additional properties those are security, business continuity, scalability, reliability, robustness, fault-tolerance, modularity and interoperability [54], accountability guaranteed by design [57]. It is necessary to bridge the gap in the agent development framework and technology for mobile devices [58]. Intelligent MAS modeling technology as a part of design technology and as a component, implementing digital twin based control will play an important role in MAS-related technology efforts [59]. Finally, an important issue of the future research and development is networking agent problems inspired by modern frameworks like IoT and cyber-physical systems [56]. All the aforementioned aspects of MAS technology form the basic directions of the roadmap for the forthcoming years.

ACKNOWLEDGMENT

The work was supported by the Ministry of Education and Science of the Russian Federation in the framework of contract agreement № 075-15-2019-1691 – the unique ID number is RFMEFI60419X0224.

REFERENCES

[1] Cossentino, M., Hilaire, V., Molesini, A., Seidita, V. (Eds.). Handbook on Agent-Oriented Design Processes. Springer. 2014. 569 pp.

[2] M. Luck, P. McBurney, O. Shehory and S. Willmott (eds.), Agent Technology, Computing as Interaction: A Roadmap for Agent Based Computing, Available form: <AgentLink web site: http://www.agentlink.org/ roadmap/>. [2 March 2020].

[3] Weiss G. (Ed). Multi-agent Systems – A Modern Approach to Distributed Artificial Intelligence. 2d Edition MIT Press, 2013. 920 p.

[4] P. Leitão, "Agent-based Distributed Manufacturing Control: A State-of-the-art Survey", Engineering Applications of AI, vol. 22, no. 7, pp. 979–991, 2009.

[5] M. Pechoucek and V. Marik, "Industrial Deployment of Multi-agent Technologies: Review and Selected Case Studies", Autonomous Agents and Multi-Agent Systems, vol. 17 (13], pp. 397–431, 2008.

[6] V. Marik and D. McFarlane, "Industrial Adoption of Agent-Based Technologies", IEEE Intelligent Systems, vol. 20 (1), pp. 27–35, 2005.

[7] P. Leitao and P. Vrba, "Recent Developments and Future Trends of Industrial Agents", in HoloMAS-2011, Springer, pp. 15-28, 2011.

[8] J. Műller and K. Fisher, "Application Impact of Multi-Agent Systems and Technologies: A Survey", in Agent-Oriented Software Engineering, Springer, pp. 1–26, 2013.

[9] P. Leitao, V. Marik and P. Vrba, "Past, present and Future of Industrial Agent Applications", IEEE Transactions on Industrial Informatics, vol. 9 (4), 2013.

[10] G. Rzhevski and P. Skobelev, Managing Complexity. London-Boston: WIT Press, 156 p., 2014.

[11] P. Skobelev, "Towards Autonomous AI Systems for Resource Management: Applications in Industry and Lessons Learned", in PAAMS 2018, Springer, LNAI 10978, pp. 12-25, 2018.

[12] G. Kaminka, "Robots Are Agents, Too!", in AAMAS-2007, Publisher Association for Computing Machinery, Inc., pp. 488-495, 2007.

[13] "The First International Workshop on the Internet of Agents (IoA)", Oct. 13, 2016, [Online]. <http://www2.cs.siu.edu/~salqithami/ioa/>. [2 March 2020].

[14] S.A. DeLoach, "Moving multi-agent systems from research to practice", Int. J. Agent-Oriented Software Engineering, vol. 3, no. 4, pp. 378–382, 2009.

[15] F. Zambonelli, N. Jennings and M. Wooldridge, "Developing Multi-agent systems: The GAIA methodology", ACM Trans. on Software Engineering and Methodology, vol. 12, no. 3, pp. 417–470, 2003.

[16] J. Mylopoulus and J. Castro: “Tropos: A Framework for Requirements-Driven Software Development”, in CAiSE 2000, Springer, pp. 108-123, 2000.

[17] S. DeLoach, “Analysis and Design using MaSE and agentTool”, in MAICS 2001, Miami Univ., pp. 1-7, 2001.

[18] C. Bernon et al., “ADELFE: A Methodology for Adaptive Multi-Agent Systems Engineering”, in ESAW 2002, Springer, pp. 156–169, 2002.

[19] G. Caire et al., “Agent Oriented Analysis using MESSAGE/UML”, in AOSE 2001, Springer, LNCS 2222, pp. 119–135, 2002.

[20] L. Padgham and M. Winikoff, “Prometheus: A Paradigm Methodology for Engineering Intelligent Agents”, in OOPSLA 2002, Sydney: Centre for Object Technology Appl. and Res., pp. 97–108, 2002.

[21] M. Cossentino, V. Hilaire, A. Molesini, Fipa design process documentation and fragmentation working group. http://www.pa.icar.cnr.it/cossentino/fipa-dpdf-wg/, 2010.

[22] M., Molesini, V. Hilaire, and V. Seidita, Introduction. In Cossentino, M., Hilaire, V., Molesini, A., Seidita, V. (Eds.). Handbook on Agent-Oriented Design Processes. Springer, 2014, pp. 1 – 5.

[23] S.A. DeLoach, and J.C. Garcia-Ojeda, The O-MASE Methodology. In Cossentino, M., Hilaire, V., Molesini, A., Seidita, V. (Eds.). Handbook on Agent-Oriented Design Processes. Springer, 2014, pp. 253 – 287.

[24] L. Cernuzzi, A. Molesini, and A. Omicin, The Gaia Methodology Process. In Cossentino, M., Hilaire, et al. (Eds.). Handbook on Agent-Oriented Design Processes. Springer, 2014, pp. 141 – 172.

[25] P. Leitao and P. Vrba, “Recent Developments and Future Trends of Industrial Agents”, Springer, LNCS 6867, pp. 15–28, 2012.

[26] P. Vrba, P. Tichy, V. Marık, et al., “Rockwell Automation’s Holonic and Multiagent Control Systems Compendium”, Transactions on Systems, Man, and Cybernetics, part C, vol. 41, no. 1, pp. 14-30, 2011.

[27] M. Mendes, P. Leitao, F. Restivo, and A. Colombo, “Service-oriented agents for collaborative industrial automation and production systems,” in HoloMAS 2009, Springer, LNAI 5696, pp. 13–24, 2009.

[28] H. Brussel, J. Wyns, P. Valckenaers, L. Bongaerts, “Reference Architecture for Holonic Manufacturing Systems: PROSA”, Computer in Industry, vol. 37, no. 3, pp. 255-274, 1998.

[29] P. Skobelev, “Multi-Agent Systems for Real Time Adaptive Resource Management”, in Industrial Agents: Emerging Applications of Software Agents in Industry, Paulo Leitão, Stamatis Karnouskos (Eds.), Elsevier, pp. 207-230, 2015.

[30] V. Vyatkin, “IEC 61499 as Enabler of Distributed and Intelligent Automation: State-of-the-Art Review”, IEEE Trans. on Ind. Informatics, pp. 768-781, 2011.

[31] V. Marik, M. Fletcher, M. Pechoucek, “Holons & Agents: Recent Developments and Mutual Impacts”, in ACAI 2001, Springer, LNAI 2322, pp. 27-35, 2002.

[32] K. Hall, R. Staron, and P. Vrba, “Experience with holonic and agent-based control systems and their adoption by industry”, in HoloMAS 2005, Springer, LNAI 3593, pp. 1–10, 2005.

[33] I. Gorodetsky, V.V. Samoylov and D.V. Trotskii, “The Reference Ontology of Collective Behavior of Autonomous Agents and Its Extensions”, Journal of Computer and Systems Sciences International, vol. 54, no. 5, pp. 765–782, 2015.

[34] M. Wooldridge. An Introduction to Multi-Agent Systems. John Wiley & Sons, 368 p, 2009.

[35] M. Tambe, “Towards Flexible Teamwork”, Journal of Artificial Intell. Res, no. 7, pp. 83–124, 1997.

[36] ACL – Agent Communication Language. Available: <http://fipa.org/specs/fipa00061/SC00061G.pdf>. [2 March 2020].

[37] RoboCup Soccer Server. Available from: <http://dangermouse.brynmawr.edu/pyrobot/tars/rcsoccersim.pdf>. [March 18, 2020].

[38] V.A. Vittikh and P.O. Skobelev, “Multi-Agent Interaction Models for Constructing the Needs-and-Means Networks in Open Systems”, Automation and Remote Control, vol. 64(1), pp. 162–169, 2003.

[39] Cougaar Agent Architecture. Available from: http://cougaar.org/wp/documentation/tutorials/. [2 March 2020].

[40] International Federation of Automatic Control (IFAC). Avail. from: https://www.ifac-control.org/ [2 March 2020].

[41] D. Ye, M. Zhang, and A. Vasilakos, “A Survey of Self-Organization Mechanisms in Multiagent Systems”, IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 47, no. 3, pp. 1–21, 2016.

[42] V. Gorodetskii, “Self-Organization and Multi-Agent Systems: II. Applications and the Development Technology”, Journal of Computer and Systems Sciences International, vol. 51 (3), pp. 391–409, 2012.

[43] T. De Wolf and T. Holvoet, “Design Patterns for Decentralized Coordination in Self-Organizing Emergent Systems”, in ESOA 2006, Springer, LNCS 4335, pp. 28–49, 2007.

[44] Y. Shoham and K. Leyton-Brown. Multi-agent systems: Algorithmic, Game Theoretic and Logical Foundations. Cambr. U. Press, 505, 2009.

[45] Granichin **O.N.,** Volkovich **Z.,** Toledano-Kitai D. Randomized Algorithms in Automatic Control and Data Mining. Springer, Series: Intelligent Systems Reference Library 2015, 251 p.

[46] I. Mayorov and P. Skobelev, “Toward Thermodynamics of Real Time Scheduling”, International Journal of Design & Nature and Ecodynamics, vol. 10, no. 3, pp. 21–223, 2015.

[47] O. Bukhvalov and V. Gorodetsky, “P2P Self-Organizing Agent System: GRID Resource Management Case”, in DCAI 2015, Springer, AISC 373, pp. 259–269, 2015.

[48] U.D. Atmojo, et al. “A Service-Oriented Programming Approach for Dynamic Distributed Manufacturing Systems.” IEEE Trans. Ind. Inform., vol. 16, no 1, pp. 151-160, Jan. 2020.

[49] V. Jirkovsky, P. Kadera. “Data Exchange Ontology for Interoperability Facilitation within Industrial Automation Domain.” Industrial Applications of Holonic and Multi-Agent Systems. LNAI no. 11710, pp. 145-158, Springer, Heidelberg, 2019.

[50] E. Negri, L. Fumugalli, and M. Macchi. “A Review of the Roles of Digital Twin in CPS-Based Production Systems”, Proc. Manufacturing, vol. 11, pp. 939-948, 2017.

[51] A 20-Year Community Roadmap for AI Research in the US, August 2019, https://cra.org/ccc/visioning/visioning-activities/2018-activities/artificial-intelligence-roadmap/#roadmapdoc [2 March 2020].

[52] H. Yang., et al. “Fault-Tolerant Cooperative Control of Multiagent Systems: A Survey of Trends and Methodologies”. IEEE Trans. Ind. Inform., vol. 16, no 1, pp. 4-17, Jan. 2020.

[53] V. Julian, and V. Botti, Multi-Agent Systems. Applied Sciences, 9, 1402, 2019, pp. 1 – 7.

[54] M. Falco and G. Robiolo, A Systematic Literature Review in Multi-Agent Systems: Patterns and Trends, 2019 XLV Latin American Computing Conference (CLEI), Panama, Panama, 2019, pp. 1-10.

[55] S. Karnouskos, and P. Leit˜ao. Key Contributing Factors to the Acceptance of Agents in Industrial Environments IEEE Transactions on Industrial Informatics 13(2), 2017, pp. 696-703.

[56] V. Jirkovsky, M. Obitko, P. Kadera, and V. Marık, “Towards Plug&Play Cyber-Physical System Components”, IEEE Transactions on Industrial Informatics, vol. 16, no 4, pp. 2803-2811, 2018.

[57] M. Baldoni, C. Baroglio, et al. Computational accountability in MAS organizations with ADOPT. Appl. Sci. 2018, 8, 489.

[58] I. Boztepe, R. Erdur, Linked Data Aware Agent Development Framework for Mobile Devices. Appl. Sci. 2018, 8, 1831.

[59] P. Novak, et al. “Engineering Roles and Information Modeling for Industry 4.0 Production System Engineering”. Proc. 24th IEEE Conference on Emerging Technologies and Factory Automation. IEEE NJ, pp. 1669-1672, 2019.