24

Bio-Inspired Multi-Agent Technology for Industrial Applications

Prof. Petr Skobelev
Institute of Control of Complex Systems of Russian Academy of Science
"Knowledge Genesis" Group of Companies
Smart Solutions, Ltd
Russia

1. Introduction

Basic ideas of multi-agent technology (MAT) began to form in the latest decades of the 20th century at the edge of artificial intelligence, object-oriented and parallel programming, and telecommunications [1-4].
The European Union Association of multi-agent systems (MAS) developers AgentLink [6] presents the Road-map of MAT up to 2020-2030 years with the slogan "Computing as Interactions". This slogan shows value of the technology not only for developing modern fully distributed systems which are now rapidly growing everywhere (what is called "ambient" and "ubiquitous" intelligence [5]) but also for solving complex problems which are difficult or even impossible to solve by classical mathematical methods or algorithms, for example, in scheduling and optimization, pattern recognition, text understanding, clustering, etc.
This relatively new area of using MAT for solving complex problems is also based on ideas of interactions. But it helps to turn complex systems from large centralized, monolithic and sequential programs with fixed hierarchical structure to distributed communities of small autonomous programs working asynchronously and in quasi-parallel with opportunity to form networking structures and interact, compete and cooperate for complex problems solving.
The value of such MAS for modern complex and rapidly changing world is difficult to overestimate. This is supported by the impressive statistics of scientific community increasing interest to these subjects. In the end of 80s MAS workshops gathered together 25-30 researchers and developers from 5-7 countries. Nowadays the situation has changed greatly. For example, in the last year World conference of Autonomous Agents and Multi-Agent Systems (AAMAS-2009) participated more than 600 delegates from 45 countries representing the results in the area of agent reasoning logic, knowledge presentation methods, platforms for multi-agent systems developments and application systems in the wide range of applications, from social processes modelling to robot control. However if the amount of scientific works in this area is rising rapidly the commercial projects and practical applications are not so well developed in spite of the fact that more than 25 commercial companies and 100 university projects in this area are world well-known nowadays.
The reason of this fact is the complexity and novelty of this new very attractive paradigm of software engineering which strongly demands new methods and tools for industrial

applications. But despite all difficulties there is a number of first successful industrial MAS projects all over the world from of network-centric logistics applications for military applications to optimization of energy consumption for cottages. The list of commercial companies that are actively developing MAS began rise distinctly at the turn of the century: LostWax (The Great Britain) – 1996 [7], Whitestein Technology / Living Systems (Switzerland) – 1999 [8], NuTech Solution (USA) – 1999 [9], etc.

According to the AgentLink Association [6] the list of these companies includes the British-Russian company Magenta Technology [10], co-founders are professor G.A. Rzhevsky from the Open university (The Great Britain) and the author of this paper. The company was founded in 1999 on the basis of the software engineering company "Knowledge Genesis" and gained valuable experiences of developing multi-agent systems for industrial applications, growing from the small group of enthusiasts in Samara in 2000 to 150 highly qualified programmers with central office in London in 2010.

The results achieved in many ways are based on the complex systems research which was held in the Institute of Control of Complex Systems of Russian Academy of Sciences under the leadership of Professor V.A. Vittih without whose strong personal support these works would not get a chance.

In this paper we will discuss main concepts, give overview of first generation of our MAT platform and MAT products for industrial applications, present new key ideas for next generation of our MAT platform and describe our current more complex and advanced MAT projects under development in "Knowledge Genesis" Group of Company and Software Engineering Company "Smart Solutions" fully specialized in real time scheduling.

We hope that presented results will stimulate new MAT developments in many new different industrial applications.

2. Theoretical background of MAT for solving complex problems: from swarm intelligence to emergent intelligence

The birthday of Samara's scientific school of multi-agent systems can be considered by June 15th, 1990. This day professor George Rzevski (Open University, London, UK) gave a series of lectures in the Institute for the Control of Complex Systems (ICCS) of Russian Academy of Sciences (previously known as the Samara's branch of Institute of Machines of Russian Academy of Sciences) about the new global real-time economy, modern theory of complexity and multi-agent systems as the new paradigm for solving complex problems. He was invited to Samara by professor V.A. Vittih the director of ICCS that time.

This series of exciting and innovative lectures became the starting point for long-term partnership of scientists from London and Samara and establishing in Samara the new R&D activities in MAS for real time logistics, text understanding and clustering and a number of other applications.

At this time e-commerce MAT applications for the Internet were the main direction of developments. It was considered that software agents will quickly become "virtual personalities" having developed mechanisms of perception, cognition, reasoning and learning based on Prolog-like inductive or deductive machines and similar tools.

At the same time well-known combinatory methods and algorithms (for example method of branches and borders) were dominating in multi-agent applications for solving complex problems, for example resource optimization.

Comparing with this approach our developments initially took completely different direction of R&D work that is called now “bio-inspired” or “swarm intelligence” approach [11]. This approach for complex problems solving is based on fundamental concepts of self-organization and evolution similar to living organisms, for example, as colonies of ants or swarms of bees.

The solution of any complex problem in such types of systems is being formed evolutionally in the process of ongoing competition and cooperation of hundred and thousands of simultaneously working very simple software agents organized as a small autonomous programs. Autonomy of agents means that agent can be invoked as state-less method in object-oriented programming but could be only asked by other agents to implement required task and can accept or reject the task because of previously agreed obligations to other agents.

For example, for solving complex optimization problems Ant Colony optimization method was developed, where the behavior of the getting food ants is modeled. The success of one ant in getting "food", i.e. taking some decisions, prompting other ants a correct direction, but after some time pheromone signs on this successful direction is "fade" requiring new trials of ants. In this case the solution of complex optimization problem can be found by interaction of a big number of relatively very simple agents continuously making trial and error attempts to get better results. But building such a solution in “swarm intelligence” approach can take rather long time while the result cannot be guaranteed and this approach is difficult to be applied in real-time.

But at this first stage “swarm intelligence” has proved that self-organization becomes an important alternative to the classical mathematics and also to traditional vision of “artificial intelligence” systems. In this approach “intelligence” should be considered not as a “mechanical” assembly of some intelligent “blocks” like “deduction”, “induction” and some other (as assembly of mechanical parts in car industry). In Swarm Intelligence the “intelligence” is not located in any of block – it is considered to be the emergent property of the system as the result of interaction of huge number of not-intelligent elements. The fact that intelligence of one ant or bee is relatively small but intelligence of colony of ants or swarm of bees is a powerful organization with high level of “adaptive intelligence” allowing to defense the nest, discover new territories, find food and solve many other tasks in continuously changing environments.

An important step in development and research of this area was done by Artur Koestler who presents the concept of holonic systems where “holons” representing properties of “parts” and “wholes” where considered as a new type of actively working building “blocks” for creation of self-organized systems [13].

The first implementation of this concept in PROSA system was done by Hendrik van Brussel, Paul Valkenaers and other authors from Christian university (Belgium) [14]. In this approach agents of “orders”, “products” and “resources” were introduced as well as “staff” agent which keeps all knowledge for decision making and advise other agents when required. This approach was successfully developed in multi-agent systems for first industrial projects for manufacturing by the team of professor Vladimir Marek (Prague Technical University, Check Republic) [15]. This approach was applied for creation of manufacturing systems for Skoda factory, control of submarine subsystems for Rockwell International, etc. This approach was advanced by the team of professor Paulo Leitao (Polytechnic Institute of Braganka, Portugal), for example, for developing ADACOR system for manufacturing [16]. These all works results in starting International Conference of Holonic and Multi-Agent Systems in Manufacturing (HoloMAS).

On the top of these developments we formed our vision of MAS considering and highlighting the following key features (Fig. 1).

Traditional systems	Multi-agent systems
• Hierarchies of large programs	• Large networks of small agents
• Sequential execution of operations	• Parallel execution of operations
• Instruction from top to bottom	• Negotiations
• Centralized decision	• Distributed decisions
• Data driven	• Knowledge driven
• Predictability	• Self-organization
• Stability	• Evolution
• Striving to reduce the complexity	• Striving to thrive with the complexity
• Total control	• Support for growth

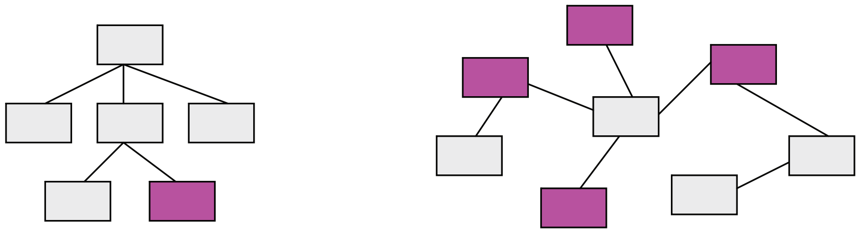


Fig. 1. Specific features of Multi-agent systems

We proposed to create multi-agent systems as a set of quasi-parallel autonomous agents designed as a co-programs with direct communication as well as with not direct communications. For not direct communication we introduced ontology-based specifications of scenes in common and shared memory of our systems which represents by semantic networks of instances of concepts and relations of domain ontology.
For these reasons the new own MAT platform was developed over Windows/Linux operation system.
In this platform agents have the following features:

- Work autonomously in platform environment, that is the agent cannot be called as a simple method but should have its own state and runs constantly. Agent cannot be forced to do something he can be only asked to do some task and he can either accept or decline the proposal depending on its goals, current state, active scenarios, etc.
- React on events and take or change decisions, including selection of scenarios and estimation of results. Agent should be able to terminate executing scenario, percept new information and react on new events.
- Communicate and negotiate with other agents what includes both direct and indirect negotiations. In the process of negotiations agents can ask each other questions, inform about the changes of their state, confirm agreement or disagreement with proposals of other agents, etc.

As a result we combine swarm-based approach with more advanced team work of agents when they can coordinate their decisions by specific advanced protocols of negotiations on virtual market of demands and resources which we will consider in more details below.

The reviewed features differ the developed platform from such well-known platforms as JADE, Cougaar, Agent Builder, JACK and others which are using different separate threads of activity for work of every separate agent what leads to significant slowdown of the whole system.

The advantage of the proposed platform is the opportunity to solve complex problems in real-time, openness, high flexibility and performance of designed systems that can be specifically applied in different spheres.

But the main feature of designed systems is their ability to demonstrate very complex behavior generated by millions of transactions of very simple agents tat generate spontaneous unpredictable revision of taken decisions at unforeseen moments of time by different agents – showcasing such phenomena of complex adaptive systems based on un-linear thermodynamics as a chaos and order, catastrophes, oscillations, etc [17].

This type of intelligence that we call "Emergent Intelligence" became the main research topic in the Samara scientific school of MAS applications and the basis for developing first generation of really intelligent systems for industrial applications in different domains.

3. Resource and demand networks of agents

3.1 Basic classes of agents in RD networks

As a one of the first problem domains we considered real time logistics of mobile resources. The problem of real time allocation, scheduling and optimization of resources is one of the most important modern problems that is characterized with a high level of uncertainty and dynamics, requires individual approach to users with conflicting interests, etc.

In our approach [18-20] was proposed to advance resource - demand networks of agents (RDN), where we define the agents (roles) of resources and demands as an entities with opposite interests, that operate on the virtual market according their economic reasons and can compete or cooperate with each other.

In this case, RDN of any domain is formed by the needs (demands) and abilities (resources) of its elements. In the simplest case, orders and resources is constantly striving to find each other and establish links.

The demand role is to get "ideal" results and the resource role – to provide best possible options in "reality". Thus, each vehicle in multi-agent logistic system knows his route, point of destination, what cargo is loaded, etc. Receiving proposals from various trucks, order can decide which of them he is best suited. But, on the other hand, the truck itself may create a new "needs", specifying exactly what orders he needed at the current time to be fully loaded or get nearest fuel station or maintenance service, driver, etc.

In an increasingly complex world of freight transportation logistic an RDN model can take into account the needs and abilities of customers and orders, trucks and cargo, travel routes, stores and warehouses, truck drivers, repair shops, fuel stations, etc. In this case, the order is constantly looking for the best truck, and truck, from the opposite side, is looking for the best order, and also the best route and the driver, etc. As a result RDN model will become more and more close to the real world transport network. This model can be expanded by the introduction of new classes of agents representing the interests of new various physical or abstract entities and with increasing number and variety of classes of interaction protocols between these agents.

A number of new RDN-based methods and tools were developed for designing first generation of industrial multi-agent systems [21-28].

In this approach each agent can be formed by the swarm of other low level agents and join the community of such agents or at any time can leave it if he is not satisfied with conditions. For example, one agent of orders allocated for one of trucks, can decide to leave swarm of truck – but as a result the conditions for other agents will change and maybe another few agents of orders will also decide to leave the truck. As a result a group of agents may leave the truck in a snowfall transition process that will become "bankrupt" on virtual market of the system – that will system chance to change attractor and optimize resulting schedule.
This approach allows us to combine the "selfish" interests of individual virtual market agents with the interests of groups of agents using the common principles of self-organization and evolution.

3.2 The virtual market of RDN agents

The core part of our any MAS is a common virtual market on which agents can buy or sell their services according to their economic reasons (Fig. 2).

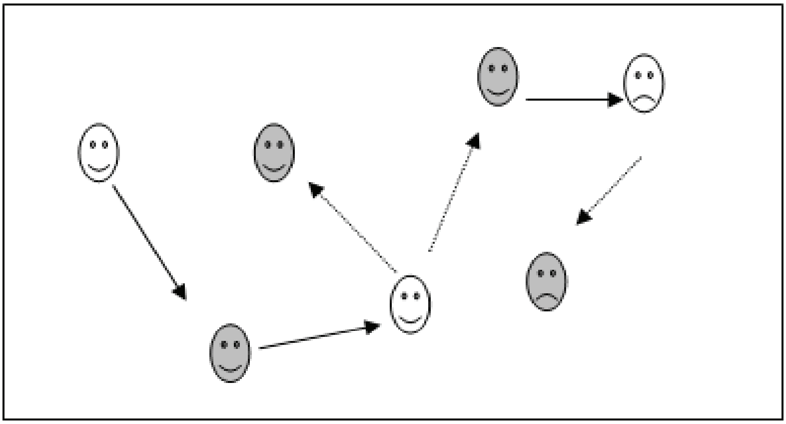


Fig. 2. The virtual market scene: demand agents (white) and resource agents (gray). Faces of agents mean satisfaction level. Different types of links shows different stages of negotiations (pre-booking, etc.)

The constant activity of all agents, from both the resource and demand sides, force multi-threads negotiations on the virtual market, going quasi-parallel [24]. In this case, each agent is designed as a state machine returns control to the dispatcher after each step of negotiations. Each agent is constantly trying to achieve its goals going into links defined on the scene with other agents (the order is booked on a truck, truck on the driver, etc.). These links could be changed by agents through recognizing and resolving conflicts, generated by events coming from outside or generated by the system itself. Agent's decisions cause a change in conditions for other agents and thereby trigger the process of self-organization in the system, leading to a reconstruction of the schedule in response to event.
Thus, the RDN always represent current solution of complex problem where agents never stop in the process of self-organization and provide adaptability of the solution, for example, form and adapt schedule in real time ("living schedule").
Rules for agents decision-making on the virtual market are determined by the microeconomic model of RD-networks, that define the virtual cost of such services, the

penalties and bonuses, rules for sharing their profits, what taxes and under what conditions should be paid, etc.
That is designed to give agents an opportunity to accumulate virtual money which plays the role of energy in the system and use them to create new or adapt fragments of existing solution.

3.3 The RDN-based MAS architecture

The RDN-based MAS architecture is presented on Fig. 3.
A key component of our MAS architecture is a virtual market which provide demand and resource role models, microeconomics, taxes, etc.
Ontology (knowledge base) for agents decisions, can be separated from the program code and updated by users. This system is provided with a special software tool to support and manage ontology and scenes [23]. In this case, every real situation could be described and stored in the system as a scene - instantiation of the concepts and relations from ontology, linking specific instances of objects (the name of the client, the driver's name, vehicle number, etc.).

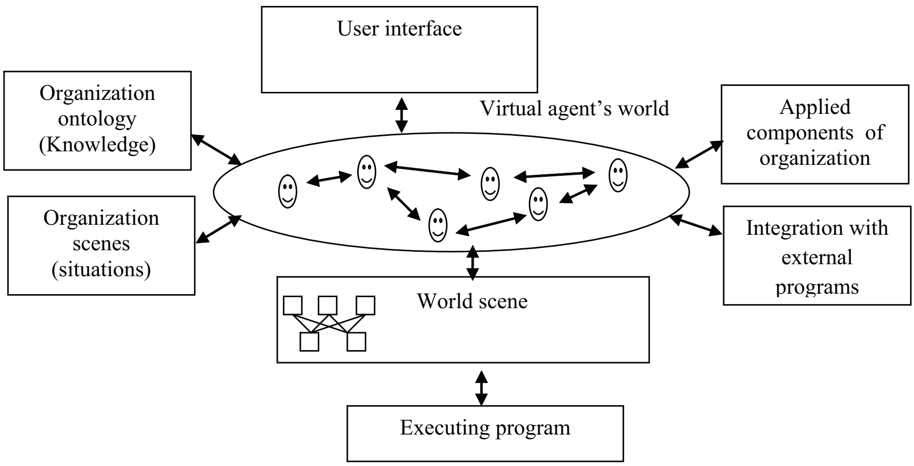


Fig. 3. MAS architecture for RD-networks implementation

The user can interact with system through web or desktop user interface. Important element of user interface is the queue of events.
When the new event coming (a new order e.g.) the system creates his demand agent, who on behalf of this order comes into interaction with the agents of resources to find the best match to place the order. If the best resource is busy, the system is fixing the conflict and making attempts to find a solution or resolution by shifts, swaps, etc. During this process, the resource (the busy one) may offer orders earlier placed on it to look for a new allocations. This process, like a chain reaction can influence and change other orders-resources allocations, forming a wave of changes (such as from the stone thrown into the water).
Similarly, if for any reason the selected resource (for example, vehicle) becomes unavailable (breakdown, accident, etc.), his agent has to find all orders scheduled on the truck and tell

inform them about the unavailability of the resource. Then the agents of the orders wake-up and activate and begin to look for other vacant trucks.
Output solutions (e.g. schedule of resources), as it was mentioned, is not considered as "static" data structure - result of a single algorithm application, but as a delicate balance (or "unstable equilibrium"), supported by the interaction of two main classes of agents of "demands" and "resources" playing the role of opposite entities of any kind like "yin" and "yang".
We consider the result is reached so the system can complete its work if there is no one chance for the agents to improve their condition.
Integration with 3rd systems is made through special integration modules.

3.4 The compensation method for balancing interests of RDN agents

To implement the developed approach a number of methods and tools uses the various swarm organizations of RDN were proposed.
The main idea of developed method of compensation [25] is that new agent needs to compensate losses for other agents which change their allocations under the request from new agent. This can made the "wave" of negotiations, in a simplified form shown in Fig. 4 - 6.

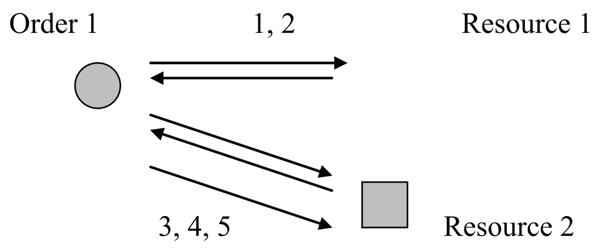


Fig. 4. The agent of the order detects two resources available and books the second one: 1 - request to the first resource, 2 - answer and suggestion from the first resource, 3 - request to the second resource, 4 - answer and a suggestion from the second resource, 5 - the second resource selected and booked

The developed approach utilizes many of the modern ideas of optimization, presented in meta-heuristics, creating an environment of competing and cooperating algorithms (agents). Thus, the agents can remember and avoid bad decisions by using their memory, they can inform each other about intermediate options, stop searching according time constraints for decision-making, etc.
An important advantage of this resource optimization technology is the ability to build adaptive plan when it is not constructed anew each time when an event occurred (as it is done in the classical optimization techniques) but adjusted in real time with the appearance of events.
This adaptation is carried out continuously by identifying conflicts in schedules, negotiating and compromising between the agents of orders and resources. This allows the system to operate in real time.
This adaptation is carried out continuously by identifying conflicts in schedules, negotiating and compromising between the agents of orders and resources and this allows the system to operate in real time.

It is interesting to note that this approach was successfully applied not only to solve the logistic tasks, but also in texts understanding, clustering, web-marketing and several others.

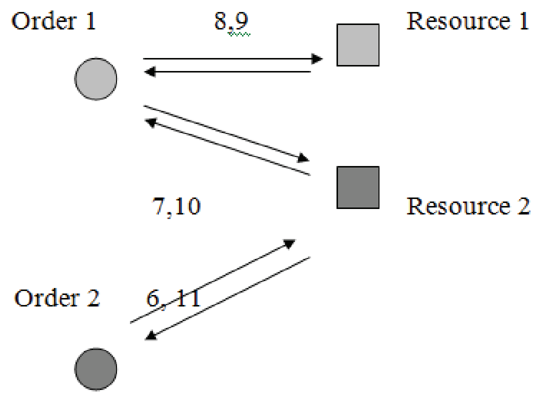


Fig. 5. The second order arrival and rebooking the second resource to the second order, with compensation: 6 - request to the second resource, 7 - divert suggestion to the first order from the second resource, 8 - request to the first resource, 9 - positive response and the offer, 10 – allowance for the second resource re-booking, 11 - agreement with the sum of compensation to the first order

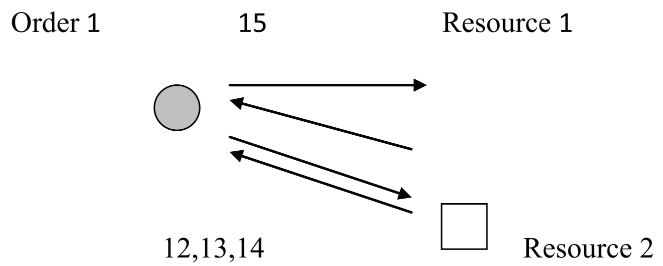


Fig. 6. The free resource appearing in the system revise the situation and reschedule booked resources, even during the execution of the order
12 - free resource initiate the running order, 13 – rebooking to the second resource is beneficial for the first order, 14 - confirmation and the second resource booking, 15 - the first resource notified about the route changing and cargo reloading to the second resource

4. The first MAS platform and the first applications (1999 - 2003)

On the basis of software engineering company "Knowledge Genesis" the Magenta Technology company was established in London in 2000 for developing first generation of multi-agent systems for solving complex problems (http://www.magenta-technology.com). One of the main catalysts of new company and its development centre in Samara was prof. G.A. Rzhevsky, whose forward-thinking vision and great knowledge and experience in complexity science made big impact on all developments. From that moment an intensive creative work on the development of multi-agent systems has begun in Samara. It was largely on the enthusiasm, overcoming any boundaries of the working time team that created real "collective intelligence" of the company. The jointed international team constantly generated new ideas many of that were implemented and grown in industrial applications.

At that period of time a few multi-agent platforms for building of multi-agent systems were developed among which the most well-known are:

- Jade (Java Agent Development Platform) – the most widely used multi-agent platform. It provides a middleware compatitable with FIPA standards to simplify the process of development of multi-agent systesm. It is promoted by Telecom Italia but supervisory board also includes Motorola, Whitestein Technologies AG, Profactor GmbH and France Telecom R&D.
- Cougar (Cognitive Agent Architecture) – also a Java-based platform for distributed multi-agent systems developed as the result of long-term project of famous military organisation DARPA in research projects ALP and Ultra*Log. It includes not only run-time engine but some visualization, data management and other tools.
- JACK Intelligent Agents – a Java-based multi-agent platform developed by commerce organisation Agent Oriented Software, Ltd. The platform of the third generation implements Procedural Reasoning Rules and supports Distributed Multi-Agent Reasoning System (dMARS). JACK uses BDI model of agents logics and built-in formal-logical means of agents work planning.

The platforms that have described previously can be all characterised by the following features:

- focused on R&D developments;
- provide very low-level programming tools;
- do not provide any specific methods and tools for dynamic scheduling and resource optimization in real-time;
- requires more efficient tools for industrial use.

The first experience of development of industrial systems has shown that for industrial applications it is required:

- to use agents world framework that encapsulates base classes of demand and resource agents and separate protocols, what enables flexible re-distribution of orders in time and among resource;
- to support large-scale schedules it is required to create a platform that supports a big number of agents that can be very simple but intelligence in work with schedule is provided as the result of their interaction and constant self-organisation and evolution;
- to create specific dispatcher and messaging system to support a big number of agents. It allows hundreds and thousands of agents to work in one thread of operational system.

In the short run the first generation of Magenta multi-agent platform was developed [26-27]. It was originally developed in Object Pascal for modelling applications and at the same time in C++ for Internet applications.

The developed platform included following components:

- *Run Time Multi-Agent Engine* - agents dispatcher designed as an operating system that can transfer control between a large number of agents (up to 100 thousand) and can provide fast transmission (up to 10 000 messages per second);
- *Ontology & Scene Constructor* - ontology constructor allows to create an ontology in the form of semantic networks for the specification of the problem domains, as well as scenes for specifying the situation and the results [28].

Key features of this platform was high performance and use of ontologies for domain knowledge specifications. On this basis a number of first prototypes of industrial systems for transport logistic were developed including the MAS solutions for tankers, taxi and trucks, couriers, train scheduling, the car windows supply and other supply chain management projects [29-36].
But also on the same platform several systems in completely different application areas were set up: smart Intranet system, real time clustering and text understanding systems, an airplane wing collective design, diet management, Internet campaigns optimizing, etc [37-48].
Later on developed multi-agent platforms were united into a one platform in the Java language for building J2EE applications, which has become de facto the industrial standard for many customers.
Let's consider the most interesting examples of the above systems.

4.1 Multi-agent clustering system

Many companies have accumulated large databases with data about customer orders, production and sale of goods, movements of goods in warehouses, hiring and firing employees, etc. In this case, gigabytes of data stored there are "dead weight" as a rule, although they contain a significant hidden knowledge about models of customer behaviour, typical characteristics of orders, patterns of decision-making, etc.
There are many software solutions designed for visualization and analysis of such data, including data mining (extracting knowledge from data), but none of them can run in real time. Besides that the task for hidden patterns revealing is still not resolved. The problem is that: generally these systems require at least prior knowledge and initial classification of the data, for example, studying the age of customers you should decide in advance what age groups (from 16 to 19 years, from 20 to 23 years and so on) will be taken into account. However, the structure of these groups by itself is a subject of research and cannot be fixed in advance. In addition, the structure of these data may vary in real time, for example, when new users come to an Internet store.
In this regard, we have developed the multi-agent approach for data clustering [38] in which the structure of the clusters are reviewed in the course of the new records coming, so we use data sets self-organization, similar as was shown above in adaptive planning.
Each cluster and each record gets his agent, acting on behalf of them. The new record can log into an existing cluster, create a new cluster, joining with the neighbours, or stay out of the cluster structure. For the record it may be better to enter a large or very dense cluster because managers start data analysis and make proposals, including rebates to its participants, from it. In some cases, managers are interested in very large clusters, but in the others - most rapidly growing or changing direction from growth to reduce, etc.
Decisions on such a virtual market every agent takes for himself, guided by the value rules given by the manager and virtual money. When it comes to sales, agent of record receives a percentage of the transaction and can spend virtual money to join one or more clusters. Similarly, if the cluster "lets in" the record (when to both of them this decision is profitable) the other records will be notified about this event. Moreover records have the right to reconsider their decision and leave this cluster, if the value for them has decreased (e.g. cluster expanded and its average density had fallen), exactly like in the truck's logistic, when an order may leave the truck, if the conditions for others have changed.
It is possible that leaving of one record from the cluster immediately can greatly worsen the situation for everyone, and then 4 more records can come out, and then another 16, etc. As a

result the cluster will "crash" and falls apart, but instead a set of other clusters, possibly having a completely different structure will form, that gives extremely interesting results for users.
This approach for data clustering was applied to analyze the Duty Free shops at the Heathrow Airport. Worked through a certain time interval the system found the fact (hidden patterns) that on Friday nights all the passengers of London-Amsterdam flights bought mostly perfume. To the store managers, this fact may indicate that at this time there is simply nothing else to buy, or on the other hand that people need this kind of products and corresponding changes in assortment can significantly increase the volume of sales and as a result the profit of the store with the same volume of rent space.
The developed system was advanced for un-structured data (texts) and applied to other number of tasks and challenges, in particular, to analyze documents of insurance company [42].
During a short period more than 20 thousand documents were analyzed, saved of about 4 man-years work for the company.

4.2 Multi-agent system of text understanding

One of the most complex modern problems, that is so far unsuccessfully solved by methods of traditional computational linguistics, is the problem of text understanding. The scope for such systems apply in the Internet is extremely broad: annotating and materials sorting, intelligent information retrieval (as opposed to the "key words" search), "smart" text editors that understands the edited text, automatic recognition of SMS messages of terrorists, etc.
The developed multi-agent approach offers an absolutely new solution of this problem, using the principles of self-organization and evolution [42]. In this approach, every word in a recognized sentence, and every semantic sense of the word in vocabulary gets his agent acting on its behalf. It should be noted that, for example, the English word "Table" in Oxford dictionary has 18 different meanings, corresponding to different contexts: Table – as a table for placing objects, Table - as the chart in Excel, etc. The purpose of agents words and agents meanings is to establish links between them in the right way, restoring the context of the scene, in this way to construct a semantic network of concepts and relationships (the scene), reflecting the situation presented in the text.
Developed approach gave a chance to create a number of industrial systems for the text understanding.
One of the system designed for automatic abstracting of articles in molecular biology to the world-known database Medline, filled up to 1 million articles on medicine every year. In the first version of the system ontology contains just 140 concepts and relations ("experiment", "organism", "belongs to", "a gene", "chromosome", etc.), but on a sample of 1000 abstracts the right search was up to 85%, although only 25% words in sentences in the abstracts were recognized by the system. That indicates a well-known redundancy of natural language and high sensitivity to the semantic meaning of phrases. This confirms the famous fact: for foreign text understanding in a narrow subject area, knowing the ontology of the world domains and understand only a small part of dictionary, is enough.
Other successful projects was for processing documents of the insurance company mentioned above [42], when it was necessary to process a large array of documents in the area of property, cars insurance, etc. In combination with clustering system, that provided the scenes construction obtained during the recognition of proposals, for few hours of work

the system found out exactly what concepts and relations are most popular in contracts and create samples of contracts that were most responding to the changing customer's requests. Approximately that requires a group of qualified experts and several years of manual work.
This approach was also used to develop the system for fax recognition [43], data meta-search in the Internet [44] and other [46-47].

4.3 Multi-agent system of collective wing design

In this project the customer had an industrial system for design of mechanical objects of an airplane wing. The main problem was to check the compatibility of engineering solutions.
The matter of fact of the problem was the high level of dependence of each individual engineer changes. The specialist could make the changes in dimension or physical and chemical properties of the materials or others, which may dangerously affect the adjacent elements of the wing. To conduct inspections of geometric objects a special system for analyzing of the geometric intersection of objects (clash analysis) used, that reveals such conflicts about once every 2 weeks, requiring time-consuming processing.
To solve this problem the multi-agent approach was proposed, where the wing was represented as a semantic network of its elements [22]. In this case, the agent that have changed the wing (e.g. landing-gear), could identify itself to a unit of semantic network of wing and immediately get the information about his neighbours units of this assembly. Comparing with previous inefficient full checks, as it was previously done by the customer, we proposed checks only of the neighbouring elements of changed parts, that reduces processing time.
As a result, the time to verify of all airplane wing units decreased by few times, what shows the high efficiency of the proposed solution.

4.4 Multi-agent system for diet management

In this project, the client worked on a Web portal that offers its users an individual diet tuneable in real time. For example, if a person ate an "extra" cake during the lunch, then he should inform the system about that so the system could reschedule menu for dinner or even breakfast for the next day, by introducing additional constraints, or in the future, offering sports activity to burn the calories, such as an evening ride on the bike.
As a solution of the problem the multi-agent system was proposed [22]. All products in it, diets and users had their agents. Agents of food themselves evolved in combinations, based on user preferences, requirements and compatibility of diet foods (for example, an agent of meat could invite an agent of red wine), taking into account personal national, religious and other characteristics of each user.
Agent of selected diet, for example, with constrain in 2000 calories, checked the resulted menu and if the amount of calories extend beyond 2000, asked the most high-calorie dish to leave the menu. All invited by that dish products were followed. The vacant place was occupied by a smaller calorie dish that provided no conflicts with other dishes. If such conflicts arise, they were resolved on the basis of user preferences so the other dish left or the dish placed before. As in others systems developed leaving of one object caused a chain of changes of the variable for not foreseen length of such wave.
As a result, the menu on the day was built as a self-organizing system, easily changing with new events occurring and deviations from the plan.

4.5 Multi-agent system of optimizing Internet campaigns

The objective of this project was to find in real time the best way of putting company's banners on websites depending on how user clicks are going.

To solve this problem, a multi-agent system was developed, which has been used successfully in a number of marketing companies [48].

If the advertising campaign is successful, the campaign agents and sites are quite satisfied in this solution. If the banner of any site is "clicked" poorly by users, it is necessary to move it to a more suitable site. If the new site placement is occupied, it needs to reconsider the position banners stationed there. Again we have the wave of changing's that can give results by self-organization through the review of decisions of banners and sites during their negotiations.

In marketing agencies in the current time such work is done by a special manager, but in practice it is very difficult to keep track of hundreds or thousands of simultaneous going campaigns to take into account the relationship between sites and campaigns, to monitor all changes in real time, etc.

The developed system can significantly improve the effectiveness of advertising campaigns and reduce the complexity and cost of this process.

5. Industrial applications in transportation logistics (2004-2008)

5.1 Brief survey of existing scheduling methods and tools

In spite of significant progress regarding development of large-scale Enterprise Resource Planning (ERP) systems, opportunities of the enterprises on development of adaptive scheduling systems remain very limited.

Traditionally the ERP systems include subsystems of orders collection, large databases for orders and resources, accounting and reporting subsystems and a lot of other components. However in these systems batch or manual scheduling of orders is supported, that was already discussed above. The schedulers offered by such large companies, as SAP, Oracle, Manugistics (it was recently bought by JDA), i2, ILOG and others usually realize various versions of Constraint programming methods, based on combinatory search of options in depth, for example, a method of branches and borders [49].

To reduce the number of options considered in combinatorial search new methods consider various heuristics and meta-heuristics (the term "heuristics" is usually understood as a set rules, defining what option is the best, and "meta-heuristics" means a rules to choose heuristics), allowing to provide good decisions for reasonable time and reducing search iterations [50].

Well-known heuristics in optimization are "greedy" methods. In such methods the decisions are taken by a choice of the best of options on each step, and once made decision is never reconsidered. Various other methods of local optimization are more complex, where initial solution which then is improving by local changes can be changed randomly or in some pre-defined way, if the good final solution is not reached, and the process repeats many times.

As one of the most known meta-heuristics we can consider Simple Local Search Based Meta-heuristics (SLSBM) – local optimization meta-heuristics. Here one of heuristics can implement casual choice of one candidate from the list of the best, another one - looking forward or randomizing of criteria, etc. One more meta-heuristics developing recently is Simulated Annealing which is based on modeling of process of cooling. This method represents an expansion of methods of local optimization in which many options could be formed on each step and it is possible to consider not only the best options, but also some worsening decisions with the probability calculated as function from some attribute, analogue of temperature.

The main idea of becoming more and more popular Tabu Search is the usage of history of decisions of local optimization when some investigated options are becoming prohibited (tabu) and consequently they are not considered on a following step.

One more new meta-heuristic is Ant Search, in which the behaviour of the ants, getting food is modelled. The success of one ant in getting of "food", i.e. taking of some decision, during some time prompts other ants a correct direction, but in due course signs on this successful direction "fade". In last period of time also many other meta-heuristics become more and more popular inheriting physical or biological concepts. Another example here is Adaptive Memory Programming method which inherits the use of common memory of decisions. In last developments researchers apply mixed miscellaneous meta-heuristics, in which several parallel algorithms are acting, and each of them suggest their own decision.

At the same time, even in view of considered methods and tools of local search of variants require greater expenses of memory and time for producing schedules. For example, producing of the optimum plan for the large transport company in one of available software packages takes about 8-10 hours. During this time the volume of orders can be essentially changed that will require to start planning all over again. At the same time the technology for planning in real time remain rather primitive, and an opportunity of flexible adaptation on the base of happening events refer mainly to an opportunity of manual plans updating. As a result, according to the estimations of transportation logistics experts, the created schedules are feasible only on 40 %, which compels many large transport companies still to contain staff of very skilled and expensive operators on planning and to carry out time-consuming manual or semi-automatic planning.

This, certainly, is promoted by both high complexity and labour intensity of planning, unpredictability of dynamics of a stream of events, by requirements of an individual approach to each order and resource, constant change of conditions of functioning of the enterprise forced by clients and competitors, and also necessity of the account of many other very specific features in each business. For example, the operator of trucks fleet should constantly keep in a head preferable time windows of loading/unloading of warehouses and shops, conditions of contracts with clients, rules of compatibility of cargoes, experience of the concrete driver and even such specific facts, that the certain road became impassable for greater wagons because of rank branches of trees.

As a result many of existing classic methods of planning and resource optimization have a number of very important limitations in practice:

- Do not consider complexities of the modern business operating in thousand of orders and resources, supporting interdependency between all operations, reflecting and balancing interests of many parties involved, having a lot of their own features;
- Do not provide opportunities for adaptive planning in real time which requires dynamic event-driven conflict solving in schedule;
- It's usually supposed that all orders and resources are identical but in practice they all have their own individual criteria, preferences and restrictions, which can change during the system work (service level, time of delivery, costs and profits, risks of delivery, etc);
- Do not give the tools for the acquiring knowledge which are specific to every enterprise, influencing quality of provided schedules;

- Do not allow an operator to explain and adjust decisions easily and in convenient way.

All this not only reduces productivity and efficiency of existing methods and tools but also in practice in many respects stops their use.

To provide opportunity to build adaptive schedulers on the top of existing ERP systems and eliminate the specified lacks in scheduling mobile objects multi-agent approach was offered which is based on the RDN concept [18-20].

It helps significantly to increase quality and efficiency of scheduling and make results more clear, understandable and adjustable for end-users and also to reduce delivery time.

5.2 Architecture of systems for adaptive scheduling

To implement the developed approach in scales of the large enterprises the architecture of system for adaptive scheduling is offered, it's presented on Fig. 7.

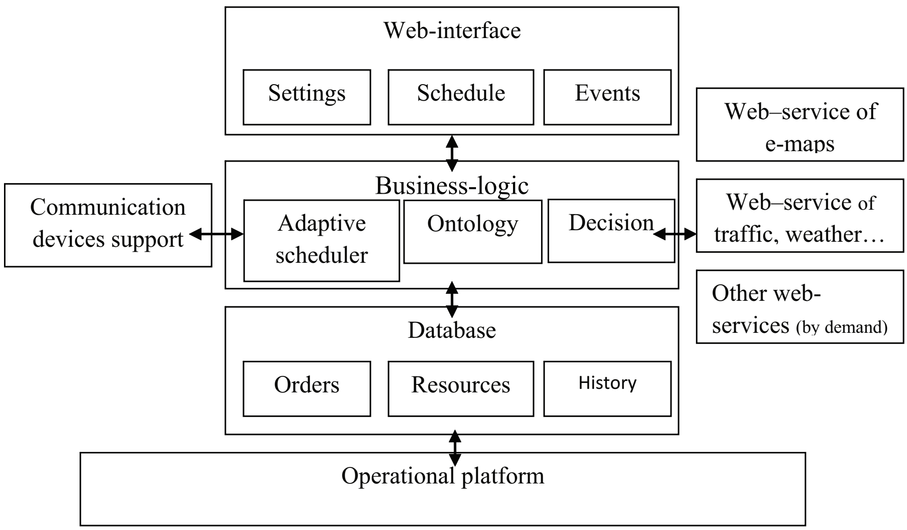


Fig. 7. Architecture of Systems for Adaptive Scheduling

Let's consider in detail the basic components of the given architecture.

This architecture implements a three-tier architecture including servers for web-interface, business-logic and databases, and also can get the operative information from external web-services and cooperate with communication devices of users (for example, drivers).

Web-interface layer of the system gives an opportunity to make settings and process orders and resources of the enterprise, etc. Through a web-interface the system operator can see the current schedule of system formed by the adaptive scheduler, in the form of Gantt chart (the schedule on each resource) or in a tabulated mode, from the side of both orders, and resources. At last, one more important component is for a display and processing of events of different type which can be transferred to scheduling manually or automatically. It is important to note, that internal and external events processing report is available for a user, that allows to explain the decision making logic of the system to an operator.

If necessary a user can be provided both with a desk-top interface for more convenient work at the local machine using web-start technology.

The layer of business-logic actually provides a reaction to events, adaptive scheduling and delivery of results. A basis of this part of the system is the adaptive scheduler constructed using the described above multi-agent approach. For each problem there can be developed a new scheduling engine, but at the same time there are certain opportunities of adaptation of existing "engine" according to new requirements by ontology configuration. The tools of ontology support allow to describe objects and attitudes of a problem domain, and also the scene describe current position of resources and orders in a transportation network at the moment of time. On this basis the rules of decision-making are formed, which can be switched on and off, be modified or adjusted by the user. The logic of decision-making is supported by the set of components, allowing to carry out calculations of distances or costs, which are specific for transport logistics, and other functions.

The database layer allows to save the information of concrete orders and resources, and also history of changes of the schedule.

The adaptive scheduling system can integrate itself with client platform or to use components of the offered platform including tools of a security control and management rights of users, provide visual reports, etc.

On the basis of given representation of architecture there can be developed the solutions on adaptive scheduling of resources for enterprises of various domains, considering the specific requirements and restrictions.

The examples of industrial application of the described approach and solution architecture are given below.

5.3 Tankers scheduling system

This system is used for management of large-capacity tankers, carrying out transcontinental transportations of oil. Everyday a company, carrying out up to 70 % of world transportation in a considered class of vessels, gets 10-15 inquiries about oil transportation [51]. Operators of the company should make the analysis of a situation in real time, sometimes even on the phone, to analyze situation, provide all economic calculations and make a decision, on what tanker it is necessary to execute the order.

At the same time it is necessary to keep constantly in a head arrangement and traffic schedules of own vessels, and also positions of competitors, count routes of traffic, consider features of passage of Suez canal if it is necessary (time for partial unloading of oil is required), consider, what ships can enter into what ports, where and when it is better to refuel the tanker, what are weather conditions, etc. To solve this problem the system of adaptive scheduling has been developed, which was integrated with a data management system. Due to small horizon of scheduling (the number of orders planned forwarding advance), in this system the arrival of a new event entails long chains of possible changes including up to 7 exchanges of orders between tankers. Thus, the new order can affect changes of a lot of tankers and even alteration of contracts with a number of clients.

Coordinator of the AgentLink European Multi-Agent Systems Roadmap Michael Luke from Southampton University (UK) explains [6] the benefits of agents in the developed system as: "By modeling each tanker as an individual agent is achieved the ability to see the options and ability to respond rapidly to emerging events in real time."

The system was developed for regular operations allowing to simulate orders allocations and schedule chosen orders more effectively. The cost of one day of idle time of such tanker is about $100,000. That allows to estimate economic benefit of the system implementation.

At the same time, the opportunity to take and formalize valuable domain-specific knowledge of operators which are necessary for decision-making turned out also very important for the customer for decreasing of human factor influence.

5.4 Corporate taxi scheduling system

This system allows the company [52], to schedule adaptively about 13 thousand orders a day at presence of several thousand cars with GPS, up to 800 from which are always on the road.
The company basically works through call centre in which 130 operators simultaneously accept calls. The company tries to provide an individual approach to each client, allocating only cars of the necessary class or a class above, with well-reputed driver, give on demand the car for disables, with the trailer, for smoking passengers, for transportation of animals, etc. The drivers work in the company as freelancers, deciding themselves what number of days and hours per week (with some restrictions) to work, renting cars at the company. At the same time they can come to work at any time. The drivers have handheld computers which allow the driver to appear on "radar" of the system when starting to work.
At occurrence of the new order the system automatically finds the best car and preliminary reserves the order. On the average the submission of the car takes about 9 minutes. From the moment reception of the urgent order, the system continues to redistribute orders for concrete time continuously in view of appearing of new resources, and does not make of the final decision till dynamically defined moment when it is necessary to send the car to client.
It is important to note that the system first tries to maximize profits. However, it also takes into account several other criteria that are important to the business. For example, when selecting from two options are roughly equal, the system gives the order to the driver, who have not received orders, thus avoiding claims to dispatchers, who previously could give a good order "their" driver. In addition, when a driver finishes, the system picks up his orders on the way home, which increases the earnings of drivers and reduces employee turnover.
The first month of implementation the number of sold orders increased by 7% with the same fleet, now 97% of all orders taxis are scheduled automatically, without the participation of managers; in 3,5 times (up to 2%) orders executed at the wrong time decreased, for 22.5% the taxi idle run, every taxi now serves on two additional trips per week at the same time and fuel costs, which is reflected in the increased profitability of each machine at 5%; taxi order is 40% faster, while the training of new operators decreased by 4 times, the website working more effectively where it comes from the 16% of orders the company.
This system has received a national award for best British innovative solution in the business in 2009, and was shown on Russian television in the "Times" program on Channel One.

5.5 Truck scheduling system

This system provides the truck scheduling for world famous networks of supermarkets. Among the transported goods there are food stuffs and drinks, including the frozen products, household electronics, clothes, etc.
The level of orders in corporate network – about 4 000 a day, the fleet of the company includes about 300 trucks of various volume, and a number of them is equipped by the additional equipment (refrigerators, etc.), the delivery network includes about 600 geographical locations all over the country.
The complexity of a problem in many respects is connected with presence of warehouses of intermediate storage, necessity of splitting of greater orders for some trips and, on the

contrary, consolidations of small orders of different volume, requirements of compatibility of cargoes, different opportunities of acceptance of trucks in different warehouses, etc.

For solving this problem the scheduling system was developed [53]. It automates all main steps of orders execution: from orders receiving and adaptive splitting and consolidation, routing and scheduling – to reports making. This system turned out to be the most difficult, where architecture of the virtual market includes a lot of agents acting together and proactively.

In particular, the orders are dynamically broken to sub-orders that are then consolidated in groups, and the trips also are formed dynamically from groups, and they, in turn, are planned for trucks. If the order has been splitted unsuccessfully, and it was not possible to plan good trips, it is made re-splitting and routing and scheduling begins anew. The big number of active agents (tens and hundred thousand) has led to necessity of application of more developed mechanisms of scheduling of agents, when only the most perspective agents get activity, competing with each other.

In present time the system is on implementation step, and decision making logic tuning is taking place. Before the deployment started the operators planned trips manually on the basis of numerous Excel tables. In this connection a lot of time was spent for adjustment of initial data in which there were many issues, including different versions of names of the same warehouse, etc.

It’s expected with the system introduction we’ll get not only significant economic benefit of more effective scheduling resources, but also the number of operators will be essentially reduced.

5.6 Car rental scheduling system

Client has about 100 stations and each of them has on the average up to 150 cars of different classes. Customers can order the car by phone, directly come to station or book car via the Internet.

For convenience of clients it is possible to agree about delivery of the car during necessary time to the necessary place. But then it is required to send the car with the driver which can work at stations as in the certain days, and overtime. Also it’s necessary to send drivers, to take away cars from clients, therefore in some cases it is necessary to send several drivers in one car, someone will bring the car to the client, and someone will take away the used cars.

For solving the problem the adaptive scheduling system was developed [54], that allows to re-schedule operatively the delivery of cars for new-coming orders and also in case of different kind of events and make schedule for drivers who bring or take cars. At the same time the system also addresses to an e-map and shows drivers recommended routes of traffic, and also sends them in real time all other necessary instructions.

Now development and testing of system on real data is finished and its expansion at first five stations is begun, up to the end of current year introduction in all other stations is expected.

The economic benefit of system consists in distributing cars in view of a situation for the whole country and to estimate more precisely, from what station it is necessary to give the car.

Reduction of the total number of cars in the network up to 10% and savings on fuel expenses and salaries of drivers are expected as the number of the superfluous trips, involved drivers and amount of overtime will be reduced.

6. New generation of MAT platform (2009-2010)

The results achieved help to generalize first experience and design second generation of platform to increase quality and efficiency of MAT solutions, cut down costs and time of development, minimise risks in industrial projects.
Key features of new platform are presented in Table 2.

№	Key Feature	Benefits	Usage examples
1.	Agent design supports full life cycle: perception, planning and execution.	Support of real time solutions and applications.	Dynamic dispatching, planning and optimization of mobile resources.
2.	Virtual market based on non-linear thermodynamics.	Speed up or slow down negotiations – to control self-organization.	Improvement of quality and efficiency of real-time scheduling.
3.	Adaptive p2p networks of real-time schedulers demonstrating co-evolution of self-organized systems.	Openness, high performance, scalability and reliability of enterprise-ready system.	Supply chain management, for example, factory and transport plans coordination.
4.	Dynamically formed bottom-up ontologies constantly renewed in the process of interaction with user.	Opportunity to fill the knowledge base on the fly without reprogramming.	In the process of interaction with a driver the system learns about such events as snowfall, road closure, etc.
5.	"Collective intelligence" support when every staff member actively participates in enterprise management.	Increase of business performance, efficiency, productivity and competitiveness.	Taxi driver using his mobile phone can signal about the groups of potential passengers.
6.	Sophisticated interaction with the system to find the problem solution.	Intellectualization of dialog with the user.	Allows the user to play with solution and adjust it on the fly accordingly.
7.	Support of work in case of uncertainty or errors in initial data.	Lack of data or errors is not a restriction for system to work.	Stability and reliability of results in case of incorrect data.
8.	Support of parallel processing.	Significant increase of performance	Planning and scheduling of large amount of resources.
9.	Combination of work in real-time with classical batch algorithms of resource planning and scheduling.	Improvement of planning quality when orders and resource are known in advance.	Strategic long-term factory scheduling.

Table 2. Key features of second generation of MAT platform

At present time we are developing a new version of such a multi-agent platform that includes the following components:

- Agents dispatcher providing control of agents states;
- Communication framework for message exchange and support of agents communication protocols;
- Framework for scene objects data models allowing to extract, make changes and save data efficiently;
- Agent design components including creation and deletion of agents, agents logic support, etc;
- Modules for dynamic scheduling, including creation of schedules, basic classes of agents, negotiation protocols, etc;
- Unser interface components including work with mobile devices;
- Basic toolset for ontology and scene management;
- Components for visualization of results (graphics and diagrams);
- Agents messages /decisions log.

For design objectives it is proposed to create the following components:

- Constructor of agents worlds, including agents classes, roles and their interaction protocols;
- Executing system for organization of parallel processing that supports multi-core processors, multi-processor systems and multi-service applications;
- Modelling system for debugging of agents worlds and methods of their interaction providing tools for generation of data, work control and showing results;
- Component of visualisation of multi-agent system behaviour;
- Agents world debugger;
- Commander as a subsystem for dialog with agents;
- Events generator for testing and playing scenarios;
- Real-time environment allowing to convert working world model into the final real-time application;
- Libraries of additional components for development specific applications.

New platform is under developments in J2EE and .Net environments.

7. Examples of new industrial applications (2009-2010)

7.1 Multi-agent inter-city transportations system

This project is created for the customer who has central office in Moscow and more than dozen of branches throughout the country. The company manages transportations by using of own fleet of 100+ trucks, equipped with GPS devices, and external carriers.

Monthly company receives hundreds and thousands of orders transport electronics, food, drinks and other products. To maximize effectiveness of truck utilization for trips from Moscow to regions and backwards it is necessary to find backhaul loads as well as to take into account contract details, minimize delivery time, possible risks and penalties, etc.

To solve this complex problem a multi-agent system that supports coordinated fleet scheduling by managers from central and branch offices was developed. While planning new order manager can simultaneously see new trucks in his region, find backhauls for them or can use these trucks to deliver his cargo. The system allocates orders to most suitable available trucks and if there no such trucks – it figures out conflicts with already allocated cargos and tries to move orders or reallocate resources by shifting, dropping or swapping orders. The system reschedules interdependent operations in case if deviation was found between planned and actual states of resources.

The system is able to automatically monitor and control business processes of order receipt, cargo loading/unloading by contacting with driver through mobile phone, as that driver has to give signals of certain operation beginning and end.

The System is integrated with 1C software to prepare related financial documents and also generates required business reports of each division effectiveness for managers and directors in real time.

7.2 Multi-agent taxi management system

Basing on the requirement of Moscow taxi company a specific low-budget taxi management system has been developed which is able to use most popular Nokia-like models of mobile phones to communicate with drivers through special Java applet.

Previously in this taxi company the whole fleet was split into groups, managed by separate dispatchers. As a result there were many situations when the taxi, assigned to one dispatcher, had to go from the northern part of Moscow to the southern part, while taxi assigned to another dispatcher goes in opposite direction, what considerably reduced effectiveness.

Another important problem was frequent taxi delays without customer notification. It was especially essential for airport transfers. Furthermore taxi company was using portable radio set which was not convenient for fleet of 500 taxis. Drivers, in their turn, always suspect dispatchers in giving most profitable orders to “favourites”. Finally, it was necessary to provide individual approach to every customer, including corporate customers.

The first version of the system was created which received practical approval on 50 cars within 3 months. Currently possibility of basic system version further functionality development and its implementation for the whole fleet of cars is being considered.

The most interesting directions for further development relating to taxi area are splitting by passengers, usage of drivers as an “intelligent sensors” and other very modern industry-specific possibilities for taxi business which become available by multi-agent technology.

7.3 Multi-agent system of airport ground-services management based on RFID technology

This project was implemented together with the University of Cologne for German Ministry of Economics and Technology within industrial consortium including Airbus, Fraunhofer institute, AutoID company RFID-tags manufacturer and BLW catering company.

The project was aimed at development of multi-agent system for modelling airport ground service operations such as food delivery on board, air stairs bringing, pickup service and luggage delivery, aircraft cleanup, defreasing and some other services.

The projects’ feature was to discover the possibility of RFID tags integration into the process of ground service management which allows to find the location of any airport resource and provide adaptive planning in real time to increase the quality and reduce costs of airlines service, increase passengers’ service level and reduce aircrafts idle time and etc.

According to the projects’ results it was concluded that it is possible to improve passengers’ service level, reduce cost and time of airlines service, reduce risks of flight delays and improve some other substantial indexes of aircraft logistics.

This solution was designed as a new generation of multi-agent systems development built as a network of cooperating schedulers each of which is responsible for its own service but at the same time coordinates tasks in close cooperation with other services.

7.4 Multi-agent real-time factory scheduling system

Developed system is made for factory workshop resource scheduling and optimization in real time including workers, equipment, materials and other.

This system was created for a large-scale airspace enterprise and can be applied for any works, that require individual approach to each production unit, nomenclature of which is constantly changing, have small production batches, require high workers qualification, have to deal with multiple unexpected situations and require high efficiency and flexibility in product realization.

To solve this problem a solution was created which allows to represent a schedule as a network of operations where agent of each operation knows who is on the right or left. The system allows easily to change the plan in case of events arising. At the same time it's possible to use different planning strategies from "just-in-time" to "as soon as possible" or "as cheap as possible", etc. It's expected that system implementation will allow to increase workshop efficiency by 15-20%.

Further system development includes adaptive network of workshop schedulers that, working separately and autonomously, will have an opportunity to compete and cooperate according to P2P scheme using enterprise service bus. In this case co-evolution and of self-organization of real time enterprise resource management systems will be demonstrated for the first time.

7.5 Multi-agent cargo planning system for International space station

This project is made by order of one of the biggest world-scale airspace corporations and is aimed at cargo transportation for International Space Station (ISS). User can build flights program, enter new launches of a spaceship, change type of spaceships and start-up time and enter other events that can change possible ways for cargo delivery.

Cosmonauts have their needs like need for water and air, fuel and food, equipment maintenance and repair, etc. As the result of system work cargo deliveries can be dynamically rescheduled, for example, amount of fuel and water, products for cosmonauts' live support and some other goods can be reallocated between spaceships flights.

At the moment this solution is developed and is at the stage of delivery.

7.6 Multi-agent system of satellites swarm management

Swarm of satellites management is one of the most leading projects based on multi-agent technologies being developed.

In this project company provides the platform for modeling cooperation in the group of orbital space satellites, fulfilling remote Earth sensing on behalf of EMERCOM and other parties. If one satellite looses an object or discovers its new features it should be investigated in more detail thus starting the cooperation with other satellites that changes their plans.

This development is focused on designing intelligence of new airspace satellites, operating as self-organized organisms and able to evolve in time due to their ability to share and reallocate tasks, collect knowledge and learn from experience.

7.7 Multi-agent scheduler of personal tasks for mobile users

This project is aimed at creation of personal tasks scheduler, which can operate or be accessible on mobile phone.

In this system on base of the ontology editor user can set up templates with sequences of actions, that then can be uploaded to the personal plan taking into consideration all semantic interdependencies between them, overlap with existing tasks, shift them, and, finally, sequence in the most convenient way for the user, constantly changing as new events occur.

With the help of ontologies used as scenario templates for tasks specification any operations chain can be managed for example for companies' business-processes control, government service delivery, taking a medicine and etc.

If necessary user can "download" to mobile device the templates that seem to be useful for business or private life situations, sport and other activities that will be planned in accordance with set up preferences and restrictions and adjusted in real time. For example if a user is at the meeting and it's time for him to go to the airport, his agent by analyzing user location with the help of GPS device, traffic jams and his current schedule finds out that user can be late and sends him a context–driven message containing offer to finish the meeting and order a taxi. Any events can overlap with the uploaded templates, causing an adaptive rescheduling of the user tasks by their shifts and drops in accordance with user preferences, interdependent tasks and etc.

In the first system version only single user tasks can be planned but since the main schedule is kept on the server later on it will be possible to collectively plan employees' work.

At the moment this project is at the stage of commercial system prototype development.

8. Conclusion

This paper gives an overview of research and application works of industrial multi-agent systems, designed and developed in Samara school of multi-agent technologies.

The main result of these studies is that the multi-agent technology allows to solve complex problems and create enterprise-ready systems based on the fundamental principles of self-organization and evolution. The developed system provides the quality and efficiency of results, reduce costs and risks and minimize dependence of human factors.

The developed methods and tools are now battle-proved, generic and applicable to a wide range of complex problems including clustering and text understanding, real time logistics, etc.

The results show wide perspectives for developing new systems for solving complex problems for many applications in real time.

9. References

[1] M. Wooldridge. An Introduction to Multiagent Systems. John Wiley and Sons Ltd, 2002, Chichester, England - 340 pp.

[2] Городецкий В.И., Грушинский М.С., Хабалов А.В. Многоагентные системы // *Новости искусств. интеллекта*. – 1998. - №2. - С. 64-116.

[3] Хорошевский В.Ф. Методы и средства проектирования и реализации мультиагентных систем // *Матер. семинара "Проблемы искусственного. интеллекта"*. - ИПУ РАН, 1999.

[4] Тарасов В.Б. Агенты, многоагентные системы, виртуальные сообщества: стратегическое направление в информатике и искусственном интеллекте // *Новости искусственного интеллекта*.–1998.- № 2.-С.5-63.
[5] Stefan Poslad. Ubiquitous Computing. Smart Devices, Environments and Interactions. – John Wiley and Sons Ltd, 2009 – 473 pp.
[6] Agent Technology: Computing as Interaction. A Roadmap for Agent Based Computing - http://www.agentlink.org/roadmap/index.html
[7] http://www.lostwax.com
[8] http://www.whitestein.com/
[9] http://nutechsolutions.com/index.asp
[10] http://www.magenta-technology.com
[11] Bonabeau E., Theraulaz G. Swarm Smarts. What computers are learning from them? // *Scientific American*. – 2000. - Vol. 282. - N 3. – P. 54-61.
[12] Thompson, *J. Ant Colony Optimisation* // School of Mathematics - Cardiff University, SWORDS, 2004.
[13] A. Koestler. The Ghost in theMachine. Arkana, 1990.
[14] Brussel H. van, Wyns J., Valckenaers P., Bongaerts L., Peeters P.: Reference architecture for holonic manufacturing systems: PROSA – Computers in Industry, 37 (3) (1998) 255 – 274.
[15] V.Marek. *Preface to Proceedings of 5-th International Conference on Industrial Applications of Holonic and Multi-Agent Systems for Manufacturing HoloMAS 2009*, Linz, Austria // Springer, 2009, pp. V-VII.
[16] P.Leitao. ADACOR:A Holonic Architecture for Agile and Adaptive Manufacturing Control. – *Computers in Industry* 2006, 57(2), 121-130.
[17] Rzevski G., Skobelev P. Emergent Intelligence in Large Scale Multi-Agent Systems // *Education and Information Technologies Journal* - Issue 2, Volume 1, 2007.
[18] Скобелев П.О. Открытые мультиагентные системы для оперативной обработки информации в процессах принятия решений. – *Автометрия*, 2002, №6, с.45-61.
[19] Виттих В.А., Скобелев П.О. Мультиагентные модели взаимодействия для построения сетей потребностей и возможностей в открытых системах. – *Автоматика и Телемеханика*, 2003, №1, с. 177-185.
[20] Виттих В.А., Скобелев П.О.. Метод сопряженных взаимодействий для управления распределением ресурсов в реальном масштабе времени // *Автометрия*, 2009, №2, стр. 78-87.
[21] Андреев В.А., Виттих В.А., Батищев С.В., Скобелев П.О. и др. Методы и средства создания открытых мультиагентных систем для поддержки процессов принятия решений. – *Известия Академии Наук. Теория и системы управления*, 2003, №1, с. 126-137.
[22] Rzevski G., Skobelev P., Andreev V. MagentaToolkit: A Set of Multi-Agent Tools for Developing Adaptive Real-Time Applications - *Proceedings of 4-th International Conference on Holonic Approach and Multi Agent Systems (HoloMAS 2007)*. – Germany, June 2007.
[23] Абрамов Д.В., Андреев С.В., Симонова Е.В., Скобелев П.О.. Разработка средств построения и использования онтологий для поддержки принятия решений // *Труды VII-ой международной конференции «Проблемы управления и моделирования в сложных системах»*, Самара 27 июня - 1 июля 2005. – СНЦ РАН, 2005, стр 435-440.

[24] Андреев В., Андреев М., Батищева Т., Олейников А., Скобелев П., Чевелёв А."Конструкция агента в системах оперативного планирования и принятия решений" // *Труды VII-ой международной конференции «Проблемы управления и моделирования в сложных системах»*, Самара 27 июня-01 июля 2005. – СНЦ РАН, 2005, стр 414-420.

[25] UK Patent Application No. 101516 – Managing a Virtual Environment – authors: G Rzevski, P Skobelev.

[26] Iwkushkin K., Minakov I., Rzevski G. and Skobelev P.: Magenta Multi-Agent Engine for Desktop Applications. *Proceedings of the 3rd Intern. Workshop on Computer Science and Information Technologies CSIT'2001*, Ufa, Russia, 21-26 September 2001. - Ufa State Aviation Technical University - Institute JurInfoR-MSU, Vol. 1: Regular Papers, 2001, 81-89 pp.

[27] Batishchev S., Iwkushkin K., Minakov I., Skobelev P. and Rzevski G.: Magenta Multi-Agent Systems: Engines, Ontologies and Applications. *Proc. of the 3rd Intern. Workshop on Computer Science and Information Technologies CSIT'2001*, Ufa, Russia, 21-26 September 2001. - Ufa State Aviation Technical University - Institute JurInfoR-MSU, Vol. 1: Regular Papers, 2001, 73-80 pp

[28] Andreev V., Iwkushkin K., Minakov I., Rzevski G., Skobelev P.: The Constructor of Ontologies for Multi-Agent Systems. In *3rd International Conference 'Complex Systems: Control and Modelling Problems*', Samara, Russia, September 4-9 2001, 480 – 488.

[29] Иващенко А.В., Ивкушкин К.В., Никифоров В.А., Скобелев П.О. Мультиагентная система поддержки управления распределением ресурсов в проектах предприятий // *Труды VII-ой Международной конференции «Проблемы управления и моделирования в сложных системах»*, Самара Самарский научный центр РАН, 27 июня-01 июля 2005, с 428-434.

[30] Rzevski G, Skobelev P, Batishchev S, Orlov A.: A Framework for Multi-Agent Modelling of Virtual Organisations. In Camarinha-Matos, L M and Afsarmanesh, H (eds), *Processes and foundations for Virtual Organisations*, Kluwer Academic Publishers, 2003, pp. 253-260.

[31] Rzevski G., Skobelev P., Korablin M.: Multi-Agent Models of Networked Organisations. *Proc. of Intern. Workshop "New Models of Business: Managerial Aspects and Enabling Technology"*, School of Management of Saint Petersburg State University, Russia, June 28-29, 2001, pp. 46-49.

[32] Батищев С.В., Орлов А.Ю., Скобелев П.О. Метод push-pull взаимодействия при составлении и адаптации краткосрочного плана производства в условиях высокой динамики спроса. // *Труды VII Международной конференции «Проблемы управления и моделирования в сложных системах»* Самара СНЦ РАН, 27 июня-1 июля 2005 с.397-402.

[33] Andreev M., Rzevski G., Ivashchenko A., Skobelev P., Shveykin P., Tsarev A. Adaptive Planning for Supply Chain Networks - *Proceedings of 4-th International Conference on Holonic Approach and Multi Agent Systems (HoloMAS 2007)*. – Germany, June 2007.

[34] Basra R., Lü K., Rzevski G., Skobelev P., Resolving Scheduling Issues of the London Underground Using a Multi-Agent System. *Lecture Notes in Artificial Intelligence* (Subseries of Lecture Notes in Computer Science) 3593, 188-196, 2005.

[35] Rajveer Basra, Kevin Lü, George Rzevski, Petr Skobelev. Resolving Scheduling Issues of the London Underground Using a Multi-Agent System – *Journal of Engineering School* , London, 2003.

[36] Rzevski G., Himoff J., Skobelev P. Magenta Technology: A Family of Multi-Agent Intelligent Schedulers. *Proceedings of Workshop on Software Agents in Information Systems and Industrial Applications (SAISIA).* - Fraunhofer IITB, Germany, February 2006.

[37] Андреев В.В., Абрамов Д.В., Карягин Д.В., Минаков И.А., Скобелев П.О. Разработка семантического Интранет компании на основе конструктора онтологий // *Труды 7-ой Международной конференции по проблемам управления и моделирования сложных систем*, Самара, - июня 2005. – Самара: СНЦ РАН, 2005. С.421-427.

[38] UK Patent Application No. 304995 – Data Mining – authors G Rzevski, I Minakov, P Skobelev.

[39] Вольман С.И., Карягин Д.В, Минаков И.А., Скобелев П.О. Разработка системы нахождения бизнес правил с использованием кластеризации на примере данных логистических компаний // *Труды VII Международной конференции « Проблемы управления и моделирования в сложных системах»*, Самара Самарский научный центр РАН, 22 июня-28 июня 2005, С.409-413.

[40] Rzevski G., Skobelev P., Minakov I., Volman S.. Dynamic Pattern Recognition Using Multi-Agent Technology. - *Proceedings of World Science and Engineering Academy and Society Conference (WSEAS 2007)* - Dallas, Texas, USA, March 2007.

[41] Minakov I., Rzevski G., Skobelev Petr, Volman S.. Creating Template Contract Documents Using Multi-Agent Text Understanding and Clustering in Car Insurance Domain. - HoloMAS 2007, pp.361-370.

[42] UK Patent Application No. 305634 – Automated Text Analysis – authors: G Rzevski, I Minakov, P Skobelev.

[43] Минаков И.А., Скобелев П.О., Томин М.С. Мультиагентная система интеллектуальной обработки факсов, используемых для обмена бизнес-данными. // *Труды VIII Международной конференции «Проблемы управления и моделирования в сложных системах»* Самара Самарский научный центр РАН, 24 июня- 28 июня 2006 с.510-515

[44] Kanteev M., Minakov I., Rzevski G., Skobelev P.. Multi-Agent Meta-Search Engine Based on Domain Ontology. - *International Workshop "Autonomous Intelligent Systems: Agent and Data Mining" (AIS-ADM 2007)* - St. Petersburg, Russia, June 3-5, 2007.

[45] Скобелев П.О. MyAgent Framework: мультиагентный подход к разработке социальных сетей с эмерджентным интеллектом. // *Труды IX Международной конференции «Проблемы управления и моделиро-вания в сложных системах»* Самара Самарский научный центр РАН, 22 - 28 июня 2007 с.511-527.

[46] Минаков И.А., Скобелев П.О. Набор инструментальных средств для интеллектуализации Интернет-порталов и социальных сетей // *Труды IX Международной конференции «Проблемы управления и моделиро-вания в сложных системах»* Самара Самарский научный центр РАН, 22 - 28 июня 2007 с.528-534.

[47] Вольман С.И., Кантеев М.К., Минаков И.А., Скобелев П.О. Разработка системы проактивных предложений релевантного клиента в режиме реального времени на основе анализа действий пользователя Интернет-порталов // *Труды IX Международной конференции «Проблемы управления и моделирования в сложных системах»* Самара Самарский научный центр РАН, 22 - 28 июня 2007 с.543-548.

[48] Вольман С.И., Минаков И.А., Скобелев П.О., Якушин А.В. Разработка мультиагентной системы поддержки принятия решений для оптимизации хода рекламных кампаний в сети Интернет. // *Труды IV Международной конференции « Проблемы управления (МКПУ-IV)* (Январь 26-30, 2009). – ИПУ РАН, 2009, стр. 1628 - 1638.

[49] Handbook of Scheduling: Algorithms, Models and Performance Analysis. Edited by J. Y-T. Leung // Chapman & Hall / CRC Computer and Information Science Series. – 2004.

[50] Stefan Vos. Meta-heuristics: The state of the Art. // Local Search for Planning and Scheduling. Edited by A. Nareyek // *ECAI 2000 Workshop*, Germany, August 21, 2000 // Springer-Verlag, Germany, 2001.

[51] Himoff J., Skobelev P., Wooldridge M. Magenta Technology: Multi-Agent Systems for Ocean Logistics – *Proc. of 4-th Intern. Conf. on Autonomous Agents and Multi Agent Systems (AAMAS 2005)*. – Holland, July 2005.

[52] Glaschenko Andrey, Ivaschenko Anton, Rzevski George, Skobelev Petr Multi-Agent Real Time Scheduling System for Taxi Companies // *Proceedings of AAMAS 2009 8th International Conference on Autonomous Agents and Multiagent Systems* 10–15 May, 2009, Budapest, Hungary – pp. 29 – 36.

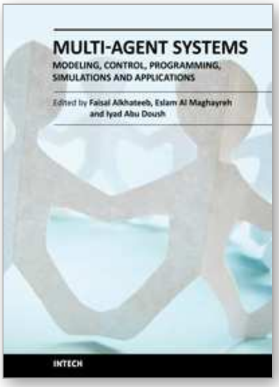
[53] Vyacheslav Andreev, Andrey Glashchenko, Anton Ivashchenko, Sergey Inozemtsev, George Rzevski, Petr Skobelev and Petr Shveykin Magenta Multi-agent Systems for Dynamic Scheduling // *Proceeding of the First International Conference on Agents and Artificial Intelligence ICAART 2009* (Porto, Portugal 19-21 January, 2009) – pp. 489 – 496.

[54] Himoff J., Rzevski G, Skobelev P. Magenta Technology: Multi-Agent Logistics i-Scheduler for Road Transportation – *Proceedings of 5-th International Conference on Autonomous Agents and Multi Agent Systems (AAMAS 2006)*. ¬– Japan, May 2006.

[55] Skobelev P., Glashenko A., Grachev I., Inozemtsev S.. MAGENTA Technology Case Studies of Magenta i-Scheduler for Road Transportation - *Proceedings of 7-th International Conference on Autonomous Agents and Multi Agent Systems AAMAS 2007* - US, Hawai, May 2007.

[56] Yankov I.A., Skobelev P.O, Shibanov S.V., Shashkov B.D., Multiagent approach for planning and resource allocation in dynamic heterogeneous networks // Dynamics of heterogeneous systems: conference proceedings / Penza, State University, 2009. Issue 1. pp. 22-37.

[57] Andeev S., Rzevski G.A., Shveikin P., Skobelev P.O., Yankov I.A. A Multi-Agent Scheduler for Rent-a-Car Companies Applications – *Proc. of Intern. Conf. of HoloMAS 2009* – Linz, Austria, September 2009. p. 305-314.



Multi-Agent Systems - Modeling, Control, Programming, Simulations and Applications

Edited by Dr. Faisal Alkhateeb

ISBN 978-953-307-174-9
Hard cover, 522 pages
Publisher InTech
Published online 01, April, 2011
Published in print edition April, 2011

A multi-agent system (MAS) is a system composed of multiple interacting intelligent agents. Multi-agent systems can be used to solve problems which are difficult or impossible for an individual agent or monolithic system to solve. Agent systems are open and extensible systems that allow for the deployment of autonomous and proactive software components. Multi-agent systems have been brought up and used in several application domains.