Multi-Agent Systems for Real Time Adaptive Resource Management

Petr Skobelev
Smart Solutions, Ltd,
Samara State Aerospace University
skobelev@kg.ru

1.1 Introduction

The 21st century will be the century of complexity compared with the previous century of physics and biology. The growing complexity of the modern real time economy is already widely recognised and associated with the increased uncertainty and dynamics of demand and supply.

The new economy strongly demands adaptive solutions for real time decision making support for resource allocation, scheduling, optimisation, co-ordination and controlling which need to support a high level of adaptability and responsiveness in real time.

However, there is a gap with the existing solutions based on combinatorial methods and tools of scheduling (Leung 2004) where all orders and resources need to be known in advance.

Multi-agent technology is considered as one of innovative and powerful tools for real time scheduling being able to solve the problem "on the fly".

In this chapter, the approach for developing multi-agent solutions for solving real time scheduling problems will be presented as well as an examples of commercial applications which have been running in day-to-day operations for several years and have produced measurable and proven benefits.

Multi-agent technology was critically important for all these applications to provide the required functionality of solutions.

1.2 The Problem and Solution for Adaptive Scheduling

1.2.1 The Modern Vision of Resource Scheduling Problem

The modern vision of the resource scheduling problem assumes that there is an organization with a number of static or GPS-based mobile resources that receives orders in real time as well as a flow of other unpredictable events: order cancellations, unavailability of resources, failures or delays etc.

The plan for resource usage has to be dynamically formed and continuously and adaptively revised taking into consideration individual set of criteria, characteristics, preferences and constraints of orders and resources. The full cycle of resource management must include fast reaction to new events, allocation of orders to resources, scheduling of orders/resources, optimization of orders (if time is available), communication with users, monitoring of plan execution, re-scheduling in case of a growing gap between the plan and reality.

The revision of the schedule must be made by the allocation of operations to open time slots or by solving conflicts between operations that can be shifted to previously allocated resource or re-allocated / swapped to the new resources.

Communication with users means supporting a dialogue with the users via mobile phones or other tools initiated by either side at any time.

1.2.2 Brief Overview of Existing Methods and Tools

The solving of classical problems on resource scheduling (also known as NP-hard complex problems) is originally formulated as a batch process where all orders and resources are given in advance and are not changed in run time.

Traditionally, the Enterprise Resource Planning (ERP) systems and schedulers offered by SAP, Oracle, Manugistic, i2, ILOG, J-Log and others, implement batch versions of linear or dynamic programming, constraint programming and other methods based on combinatory search of options (Shirzadeh Chaleshtari and Shadrokh, 2012).). In case of one well known scheduling system it may take up to 8-12 hours to allocate 300 trucks to 4500 orders. However, it may turn out that only 40% of this schedule is feasible in the real world application.

To reduce the complexity of combinatorial search, new methods consider heuristics and meta-heuristics (Vos 2001), allowing the provision of acceptable decisions in reasonable time and reducing search options. Some examples are “greedy” local search methods, simulated annealing, adaptive memory programming, tabu search and ant optimization.

However, these methods still use batch processing and struggle to take into consideration real life criteria, preferences and constraints.
The search for options remains very time consuming and results are often just not feasible or not comparable with human decisions by nature.

1.2.3 The Multi-Agent Technology for Adaptive Scheduling

The fundamentals of multi-agent technology began to form in the last decades of the 20th century at the edge of artificial intelligence, object-oriented and parallel programming and telecommunications (Wooldridge, 2002).
In contrast with the classical large, centralised, monolithic and sequential programs multi-agent systems (MAS) are built as distributed communities of small autonomous software objects working asynchronously but in a co-ordinated way to get the results.

The key features of MAS can be specified in the following way:

- agents work autonomously, that means agent can not be called as a method but can only be asked to carry out tasks;
- agents can react to events but can also trigger their activities internally and try to proactively achieve their objectives;
- agents can communicate and coordinate decisions with other agents and can change their decisions adaptively.

At present multi-agent technology is considered as a new paradigm for solving complex problems that are difficult or even impossible to solve by classical mathematical methods or algorithms (Shoham and Leyton-Brown, 2009), for example, in scheduling and optimization, pattern recognition, text understanding and some other domains.
Multi-agent technology was initially applied to solve classical optimisation problems with the use of distributed problem solving approaches, for example Distributed Constraint Optimisation Problem (Rolf and Kuchcinski, 2011). Alternatively a number of bio-inspired methods were developed, for example, swarm optimisation, hybrid methods based on artificial immune system (AIS) and particle swarm optimization (PSO) for solving production planning problems and others (Gongfa 2011; Xueni and Lau, 2010).
As a next step a market-based approach to scheduling was developed where order agents and the resource agents participate in continuously running auctions based on contact-net protocols (Pinedo 2008; Allan 2010; Noller et al., 2013).
There are a growing number of first prototypes and industrial solutions based on multi-agent solutions (Pechoucek and Marík 2008; Leitao and Vrba 2011; Florea et al., 2013).

1.2.4 The Concept of Demand-Resource Networks

The developed approach is based on a “holon” concept of PROSA system (Brussel et al., 1998) where specific classes of agents of “orders”, “products”

and “resources” were introduced as well as a “staff” agent which monitors results and advises other agents when required.
To make this approach more flexible and efficient the concept of Demand-Supply Networks (DSN) was introduced where agents of demands and supply are competing and cooperating on Virtual Market (VM). In the concept any agent (holon) of physical or abstract entity can generate “small” demand and supply agents, which follow the specific requirements, for example, truck can demand driver or fuel, route or maintenance. From the other side, truck can be busy for first half of a day only and is interested to find and supply truck for orders for the second part of a day. As a result, the schedule can be formed as a kind of requirement-driven network of operations which can be easily adapted by events in real time (Skobelev and Vittikh, 2003, 2009).
Another example: an order for product assembling can generate demands for specific equipment and a worker as a factory resources but the same equipment can generate a new demand on regular maintenance or special repairmen. The role of demand agent here is to get the best possible time slot of equipment and worker in the factory schedule or a truck in cargo transportation, etc. The role of supply agent is the opposite - to provide full utilisation of resource. Having received proposals from various supply agents, the demand agent can decide which proposal suits the best, and vice versa, because DSN agents have conflicting interests and operate in the VM according their economic reasons. The decision-making rules for agents in the virtual market are determined by the microeconomic model of DSN that define the virtual cost of services, penalties and bonuses, rules for sharing the profits, what taxes should be paid under various conditions, etc. It gives agents an opportunity to accumulate virtual money and use it for getting best possible options. In fact, the virtual money plays the role of energy and agents use it to create new schedules or to adapt fragments of the existing ones. This model can become more and more complex by introducing new agents in DSN that represent interests of various physical or abstract entities and with the increasing number and variety of classes of interaction protocols between agents.
Specific DSN-based methods and tools were developed to design adaptive multi-agent systems for real-time scheduling (Rzevski and Skobelev, 2014).

1.2.5 The Formal Problem Statement

The formalised problem statement is based on searching for a consensus between agents in DSN virtual market and can be formulated as following.

Let’s assume that all agents of demands and supply have their own goals, criteria, preferences and constraints (for example, due date, cost, risk, priority, required equipment type or worker qualification). The importance of each criterion can be represented by weight coefficients in a linear combination of criteria for the given situation in scheduling but can change during the schedule forming or execution.

Let’s introduce the satisfaction function for each agent (Fig. 1a), which will show deviation of the current value of this function from the given ideal value by any of the criteria for the current step of finding scheduling solution for this agent. The activity of agents also depends on bonus/penalty function and current budget allocated on specific accounts for virtual money (Fig. 1b).

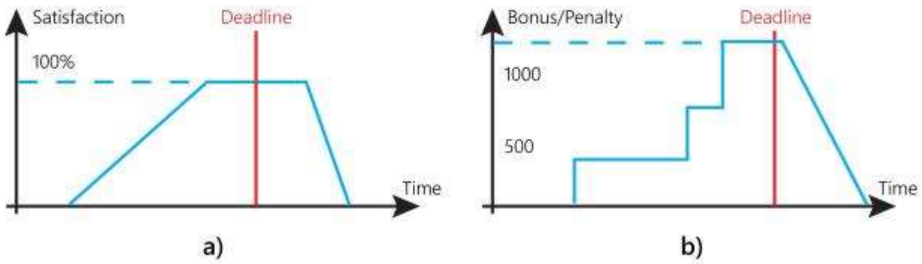


Fig. 1. Example of satisfaction function (a) and bonus/penalty function (b).

Let each demand j has several individual criteria x_i and suggested ideal values x_{ij}^{id}. For each agent of demand j normalised bonus/penalty function is calculated by component i («virtual value»), given for example as a linear function $f_{ij}^{task}(x_i - x_{ij}^{id})$. In the most of cases this function has bell form with maximum in the point of suggested ideal value. As a summary value of the result for each demand, the sum of virtual values for each criterion i with the given weight coefficients α_{ij}^{task} is estimated.

By the proper selection of signs and form of the function, the goal of each agent can be reformulated as maximizing of virtual value y_j^{task} of demand j (upper index *task* means that the values belong to the demand agents):

$$y_j^{task} = \sum_i \alpha_{ij}^{task} \cdot f_{ij}^{task}(x_i - x_{ij}^{id}),$$

where $\forall j$ weight coefficients are normalised: $\sum_i \alpha_{ij}^{task} = 1$.

Similarly the problem of finding the states x_{ij}^{*} of agents of demands j, that maximize the total value of all orders can be formulated:

$$y^{task} = \sum_j \beta_j^{task} y_j^{task} = \sum_j \beta_j^{task} \sum_i \alpha_{ij}^{task} f_{ij}^{task}\left(x_i - x_{ij}^{id}\right) \quad (1)$$

$$y^{task^*} = \max_{x_i}(y^{task})$$

where β_j^{task} is demand weight that allows to set and dynamically change the priorities showing importance of criteria.

Similarly the value function can be given for the supply by criteria z_k, with bonus/penalty function $f_{kl}^{res}(z_k - z_{kl}^{id})$, weight α_{kl}^{res} of criterion k for resource l, and resource value β_l^{res} for the system (which is similar for weight for demand agents function):

$$y^{res} = \sum_l \beta_l^{res} \cdot y_l^{res} = \sum_l \beta_l^{res} \sum_k \alpha_{kl}^{res} \cdot f_{kl}^{res}\left(z_k - x_{kl}^{id}\right) \quad (2)$$

$$y^{res*} = \max_{z_k}(y^{res})$$

$$z_k \in D^K, \ x_i \in D^I \ \ \forall i, k, I = Dim(D^I), K = Dim(D^K) \quad (3)$$

Variables x and z belong to some areas of the space of criteria for demands and supply, I and K are dimensions of the corresponding criteria spaces, upper index *res* means that the values belong to resource agents.
Thus in DRN the optimisation problem is formulated as solving (1)-(3).
In other words, in the suggested bottom-up methodology one global optimiser is replaced by many small local optimizers which are able to negotiate and find trade-offs when they search their local optimums.

1.2.6 The Method of Adaptive Scheduling

The developed method is based on DSN concept where agents of demands and supply operate in the VM and continuously try to improve their individual functions of satisfaction that reflect their given multi-criteria objectives.

The core part of the developed method can be identified as the following:

1. The number of classes of demand and supply agents represents specifics of the problem domain with the required level of granularity.
2. Satisfaction function and function of bonuses / penalties are represented by linear combination of multi-criteria objectives, preferences and constraints of each agent.
3. Protocols are defined which specify how to identify conflicts and find trade-offs with the open slots, shifts and swaps of operations.
4. A schedule formed in the process of DSN agents self-organization is based on decision-making and interaction of agents.
5. Special event procession protocols are triggered when new events occur (for example, arrival of a new demand):
    a. An agent is allocated to a demand as it arrives into the system. The Demand Agent sends a message to all agents assigned to available resources stating that it requires a resource with particular features and it can pay for this resource with a certain amount of virtual money.

    b. All agents representing resources with all or some specified features and with the cost smaller or equal to the specified amount of money, offer them to the Demand Agent.

    c. The Demand Agent selects the most appropriate free resource from those on offer. If no suitable resource is free, the Demand Agent attempts to obtain a resource, which has already been linked to another demand, by offering to that demand some compensation.

    d. The Demand Agent who has been offered some compensation considers the offer. It accepts the offer only if the compensation enables it to obtain a different satisfactory resource and at the same time increase the overall value of the system.

    e. If the Demand Agent accepts the offer, it reorganises the previously established relationship between that demand and resource and search for a new relationship with resource increasing the overall value of the system.

f. The same process is running for Resource agents which are able to generate Supply agents with specific context-based requirements.

6. The above process is repeated until all resources are linked to orders and there is no way for agents to improve their current state or until the time available is exhausted.

To achieve the best possible results agents use the virtual money that regulates their behaviour. The amount of virtual money can be increased by getting bonuses or decreased by penalties depending of their individual cost functions. The key rule of the designed VM is that any agent that is searching for a new better position in the schedule must compensate losses to other agents that change their allocations to resources, and propagation of such wave of changes is limited by virtual money (Skobelev and Vittikh, 2009).

The use of the VM presumes that demands buy the services of the resources that, in their turn, have static or dynamic cost. The dynamic cost of the resource depends on how resource can be shared. For example, a truck has a certain cost, but it is distributed between it's cargoes with some planned profitability given in advance. As it is stated before, the agents can offer each other compensations for shifts and reallocations, the sum of which is defined during negotiations between demand and supply agents. If the cost of functioning is not covered by the income, the resource can decide to switch off.

The main features of the suggested VM microeconomics are:

- Agents have ideal and current values of objective functions, which are used to compute agents "satisfaction" by the current plan;
- Order agents enter the system having virtual money to achieve their objectives, including service level, costs and delivery time;
- Resource agents looking for their maximum utilization but also have their own ideal preferences and constraints and costs for sharing;
- Product agents are interested to minimize time spent in storage;
- There are dynamic values of weight coefficients of the scalarised objective function which are linked to a virtual money bonus or penalty for Orders, Products and Resources, each criteria has its own coefficient of conversion to the virtual money;
- Current virtual budget is used by agents to improve their local allocation of demands and supply in the schedule;
- Agents iteratively improve their criteria to reach locally optimal values compensating the losses of other agents from their virtual budget – in a way that virtual profit is growing.

Such approach gives an opportunity to introduce virtual taxes related to agents' jobs planning and execution, cost of messaging between agents, etc. These taxing mechanisms can be used to control the process of self-organization of the schedule to provide a good quality schedule within limited time.

If necessary, the user can interactively intervene the plan at any time and manually rework the schedule by dragging and dropping the operations.

As a result, the plan will be automatically revised and rescheduled.

1.2.7 The Basic Multi-Agent Solution for Adaptive Scheduling

The basic solution includes a number of agents which are applicable for various domains of real time resource scheduling:

- agent dispatcher which supports agent life cycle, creation and termination of agents, communication protocols;
- agent for supporting messaging services;
- main classes of PROSA agents and supporting agents of DSN;
- event queue agent who is responsible for events processing;
- scene agent for data loading and serving the resulting schedule.

The list of the designed basic classes of agents is shown in Table 1.

Table 1. Main classes of agents.

Agent Class	Specification of agent behaviour, main goals and tasks	Attributes
Order	The goal of the order agent is to complete the order in time, with maximum quality, minimal cost, best delivery time and minimum risk. Tasks include loading of business process, creation of the business process (BP) agent, analysis of their results, changing settings and strategies for the BP agents, triggering proactivity of BP agents.	Service level, real and virtual money for order execution, given specifications for resources, deadline for order execution, risks.
Business process (techno-logical process)	The goal of the BP agent is to coordinate business or technical jobs (tasks or operations) and make sure they are properly scheduled. Tasks include decomposition of process into jobs (operations), creation of the job agent, analysis of their results, changing settings and strategies for the job agents, triggering proactivity of job agents.	Preferred time slots, real and virtual money for order execution, given specifications for resources, jobs interdependencies and deadlines.
Job/Task	The goal of the job agent is to find the best possible resource for executing the job. Tasks include finding the best resources with matching characteristics and getting agreement on allocating job to the free time slot or starting negotiations for solving conflicts with the previously allocated jobs by shifting and moving the jobs between resources, proactive improvement of the job state according to the situation.	Given characteristics of resources, real and virtual money for job execution, time and cost preferences, interconnections between jobs, deadlines.
Person for job execution	The goal of the person agent is to maximise resource workload and utilisation by the best orders and get a bigger salary. The tasks include participating in matching and negotiations of jobs allocation, calculating dynamic price for jobs, sharing costs between jobs, state analysis and proactive search for better jobs, overriding availability constraints when required, calculating salary and bonuses.	Availability (for example, 8-hour working day), key competencies and skills, current load and potential capacity, cost and risks.
Machine or tool for	The goal of machine is to maximise resource workload and utilisation by using the best orders. One person can	Availability, maintenance

job execution	operate with a few machines or one machine can require a few persons. Machine may require regular maintenance or repairmen.	regularity, load productivity, cost and risks, energy consumption.
Product (physical or abstract)	The goal of the product agent is to get the best characteristics to match order specifications and requirements and to be delivered in time and with minimum costs and risks.	Domain-specific product requirements which are specified in order
Organisation (team, department)	The goal is to balance workload of the resources. The tasks include switching the resources on and off, resource workload monitoring, pre-selecting resource for allocation, discovering the "bottlenecks" in organisation and generating recommendations, calculating KPIs for organisation, managing resource strategies.	The list of resources, availability of resources and other preferences and constraints for organisation.
Event	The goal is to manage the events queue. The tasks are input of the event into the system, activation of the required agents, collecting information on event processing, estimation of the result of event processing.	Event type, time of occurrence and time of input of event, time of event processing, value of event.
Resulting schedule	The goal is to fix the resulting schedule for the users. The tasks include monitoring scheduling process and fixing the result in case when it has reached the required level of quality, oscillations of solution reached the "plateau" with given delta-epsilon, the available time was exceeded or the user intervened.	Level of the solution quality, delta-epsilon for oscillation, the available time interval.

The presented list of agents (Tab. 1) can be adjusted during the development process for specific domain of scheduling. As an example, the list of agents and protocols of their negotiations for factory scheduler is given in Fig. 2.

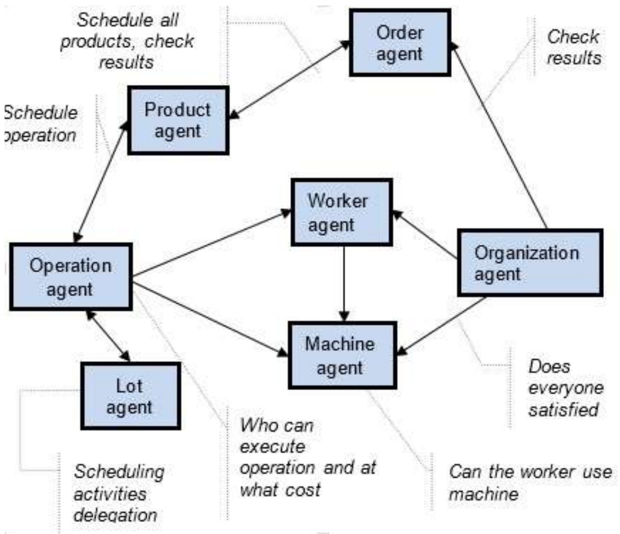


Fig. 2. Main classes and lines of agent communications.

The “scene” is an object model of the forming schedule in which jobs / tasks linked with time slots of resources. Scene is considered to be a “mirror” of the reality. In the new versions of adaptive schedulers the scene is formed as a ontology-based semantic network of key domain objects and relations for example, in factories linking orders and technological operations, equipment and workers, skills and competencies. These links are continuously investigated by agents and help them to narrow search and find reasonable options by analysing the “topology” of the schedule.
The solution can be easily integrated with the existing ERP systems, for example, order management, accounting, etc.

1.2.8 The Multi-Agent Platform for Adaptive Scheduling

Multi-agent platform is designed to automate the developed methodology and increase quality and efficiency of the development process for creating real time resource management systems for different problem domains.
The developed multi-agent platform combines functionality of a basic adaptive scheduler that can be easily modified for new domain with simulation environment which is useful for experiments with the different DSN models, methods and algorithms.
Functionality of the multi-agent platform provides a possibility for end-users to specify initial network of resources, form sequence of events manually or automatically or load it from external files, make individual setting for all demands and resources, run simulations with different parameters and visualise process and results of experiments.
An example of the user interface of the platform representing results of experiments with given flow of orders is shown in Fig 3.
The screen presents a network of resources, Gant chart with the resource schedule, workload of orders and resources, satisfaction of orders and resources, virtual money transfer, log of decisions and some others.
During the simulation mode, a number of useful charts and diagrams can be visualised or exported in Excel files for future investigations:

- graph of network loading shows utilisation of all resources;
- Gantt-chart shows allocation of demands to resources in time;
- communication activity diagram shows how many messages are generated in the platform at any moment of time;
- satisfaction of demand and resources chart demonstrates how the satisfaction level changes during the process of simulations;
- orders execution chart shows status of orders execution;
- resource utilization chart shows how busy resources are at different moments of time;
- message log demonstrates exchange of messages between selected agents;
- decision making log presents results of decision making for a selected agent;

- financial transactions log shows transfers of virtual money between demand and supply agents.

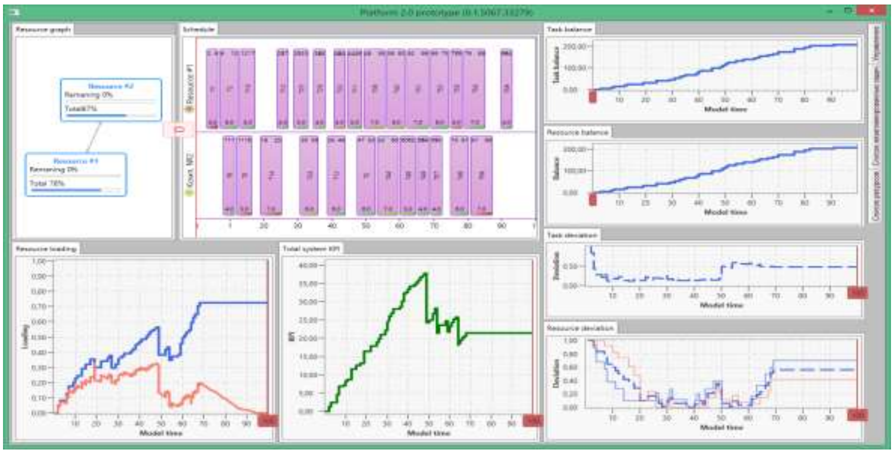


Fig. 3. Screen of multi-agent platform for real time resource management.

The platform architecture includes the following components: initial scene editor, event generator, event queue for main classes of events, multi-agent world built as a virtual market, basic classes of agents and supporting demand and supply agents, visual components for editing agents’ settings and visualization of results, export and import of data, logging and tracking of messages and agent financial transactions and some other specific components. These components can be adjusted for new problem domains and applications.

1.3 Examples of Applications for Industry

1.3.1 Multi-Agent System for Flights and Cargo Scheduling for International Space Station

1.3.1.1 Application Overview

The International Space Station (ISS) is one of the most complex engineering projects in the history of mankind.

Servicing of the ISS requires scheduling of flights with the focus on scheduling of the space crew activities and delivering such cargoes to the space station as fuel, water and food for astronauts, laboratory equipment, materials, tools and other types of objects, as well, as scheduling of cargo returns back to Earth.

The main problem here is the limited capacity of spaceships which requires adaptive event-based re-scheduling of cargo deliveries, for example, when unpredictable demand for an additional cargo arrives, fuel or water volumes or amounts of other resources may need to be recalculated and reduced.

Multi-agent system for the ISS flights and cargo scheduling provides an interactive support for developing a plan of flights and cargo deliveries taking into consideration a number of preferences and constraints: for example, different spaceship types and types of the ISS modules, number of astronauts,

fuel consumption forecast, solar activity and ballistic requirements, minimal period of time between operations of docking and undocking; permanent presence of at least one piloted ship docked to the station and many other specifics (Skobelev, 2011).
The key features of multi-agent world of solution are presented in Fig. 4.

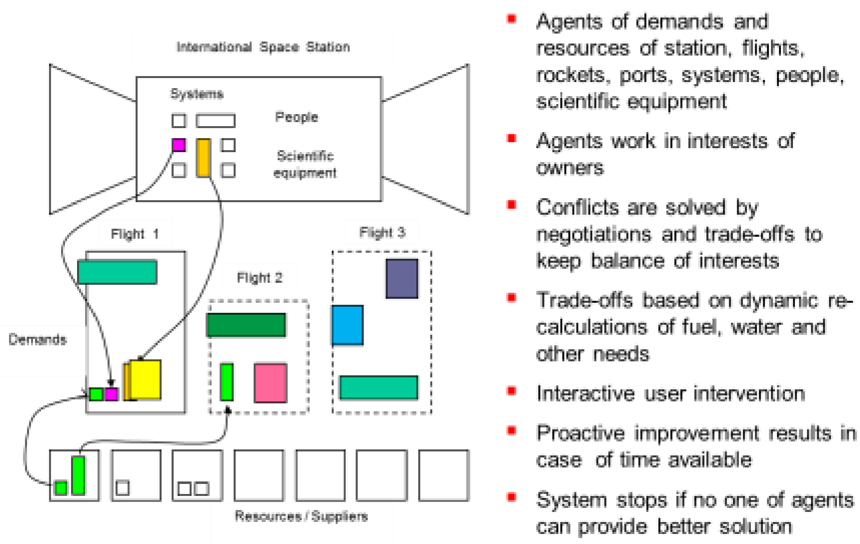


Fig. 4. Key features of multi-agent world of solution.

The examples of the system user interfaces which a detailed explanation of the screens are presented in Fig. 5–7.

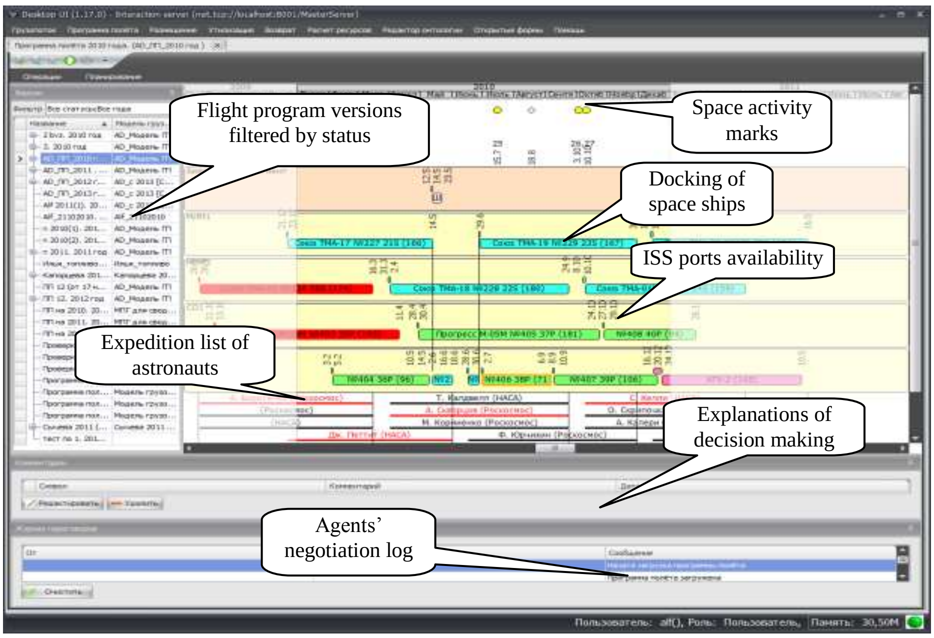


Fig. 5. Interactive flight program editor

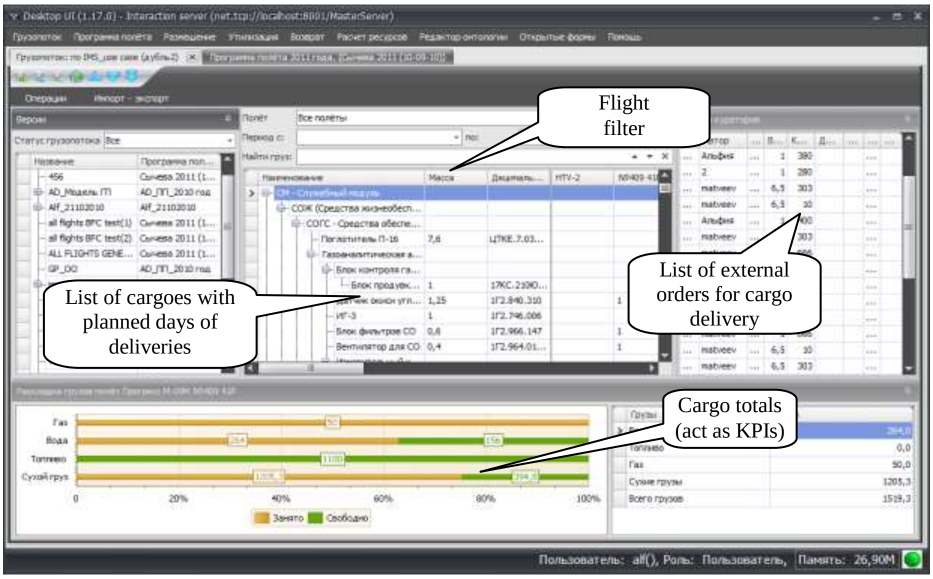


Fig. 6. Cargo flow delivery plan.

Fig. 7. Fuel delivery/spending balance.

Flight program design, scheduling of cargo flow and resources of ISS starts with the creation of a strategic model of cargo flow. There is a need to create a strategic model of cargo flow, which helps to calculate the number of required transportation flights per year on the basis of the number of expected expeditions. The number and times of dockings and un-dockings of space ships to ISS modules is determined at this stage (Fig. 5). On this basis the program of updated cargo flow (fuel, water, other human life items, scientific equipment, etc) is created. Further it is constantly being corrected adaptively (Fig. 6, 7).

The described solution was implemented on .NET platform using MS SQL Server 2005 as a database server and integrated with the ISS on-board inventory management system which provides every day updates on cargo utilization.

1.3.1.2 Benefits and Assessment

The solution was designed and implemented in 2010-2012 for one of the world's biggest Rocket and Space Corporation "Energia".

The system's main functionality allows fast interactive development and comparing options for flight program and plan of cargo deliveries, including adaptive event-driven re-scheduling of cargo in case of unpredictable events. At the moment the system is used by the team of 8 core specialists and 120 users who operate the system on a daily basis and generate schedules for about 3500 types of cargoes.

The system has significantly reduced complexity of resource computations and speeded-up scheduling for new flight/cargo programs development up to 4-5 times, improved transparency and coordination of all operations.

The system provides an opportunity to simulate the worst-case scenarios for risk management which is critically important for success of the space mission.

1.3.2 Multi-Agent System for Scheduling Factory Workshops

1.3.2.1 Application Overview

The multi-agent system "Smart Factory" is designed to increase the factory's productivity and efficiency by adaptive resource allocation, scheduling, optimisation and controlling for machine assembling workshops in real time.

Adaptability means that each event in the workshop can influence the workers' schedule, shift or reallocate the previously scheduled orders and resources and resolve conflicts. Examples of the events that can lead to rescheduling are arrival of a new order, equipment failure, changes in priorities, new urgent tasks, delay in delivery of materials or operations of workers, etc.

The system can be used for factories which can be characterised by on-going innovations, complexity and dynamics of operations, as well as high uncertainty in supply and demand that require a high level of real-time adaptability in reaction to unpredictable events.

The system architecture is represented by three main tiers including Application server component, Client components and Database. Events are usually processed sequentially but some of these events have a higher priority.

The adaptive scheduling solution is a part of the Application server component that contains agents of orders, workers, machines, operations and materials, which take into consideration a relationship between the operations.

Agents are constantly trying to respond to new events but also to proactively improve the operation plan by using free machines or free time slots in the workers' schedule by the chain of moves and rearrangements of the previously scheduled operations or by transferring them to other resources. As a result, the work plan of the workshop is also built here not by a classical combinatorial search, but as a balance between the interests of all mentioned agents.

Examples of the user interfaces of the developed system are shown in Fig. 8-13. Fig. 8 shows all orders for workshops with their current order status including: not started, planned, started, executed, in preparation, stopped, delayed, postponed, etc. Fig. 9 shows the key stages of preparing and loading data for the scheduler including technology processes, time estimates for each operation, competencies of workers, etc.

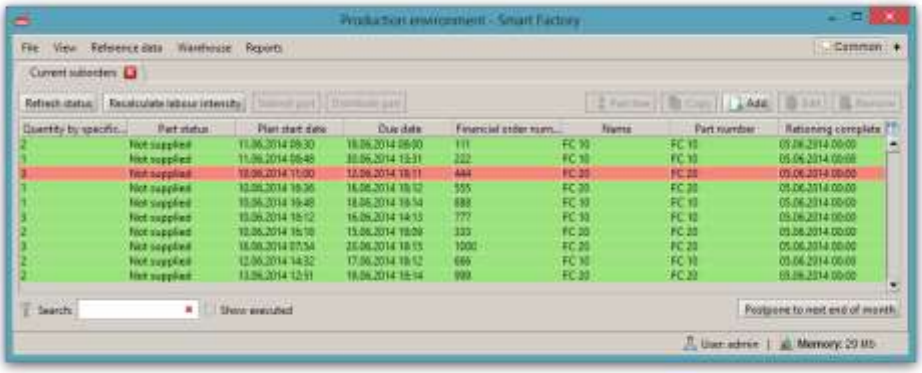


Fig. 8. Screens of Orders Status.

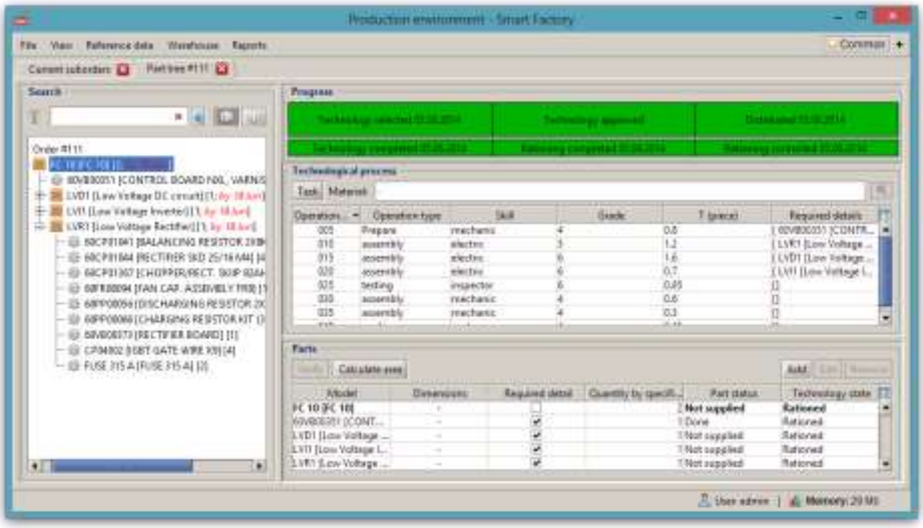


Fig. 9. Screens of Technology Loading.

Events Queue gives a possibility to managers to enter information on new events and start rescheduling as it is shown in Fig. 10, for instance, entering a new order for manufacturing which hierarchy components are visualised on the left part of the screen. In Fig. 11 the combined Gantt and Pert diagrams show interdependencies between the manufacturing operations. The user can select any operation on the screen and "drag and drop" it to another worker as well as merge or split operations and adjust the event plan by triggering an automatic chain of changes in the schedule.

In case a worker does not have enough skills for the operation, the system will highlight this operation in red color and give a warning message to the user (Fig. 12). The list of tasks for workers (Fig.13) can be printed in a traditional form or presented in a kiosk with a touch screen.

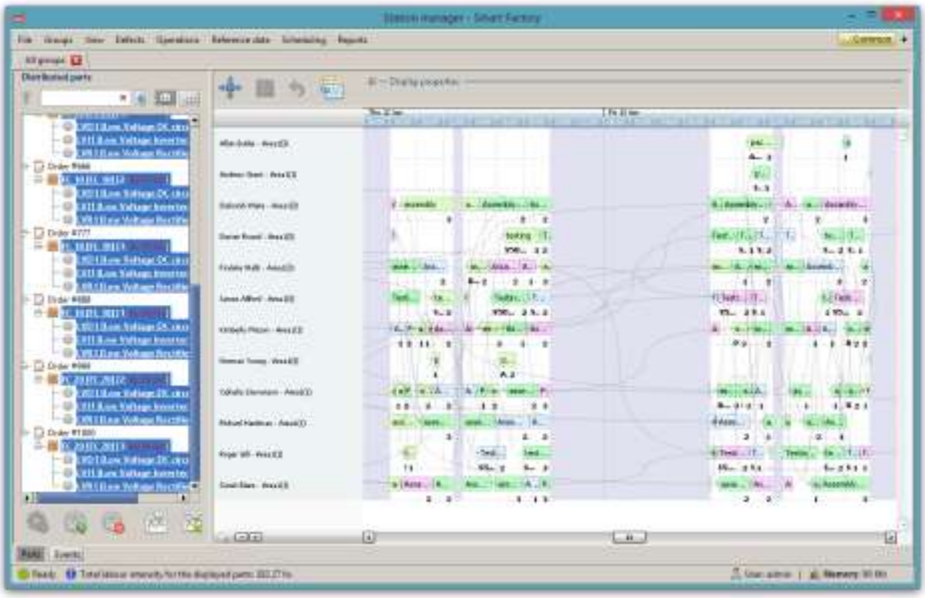


Fig. 10. Queue of Events.

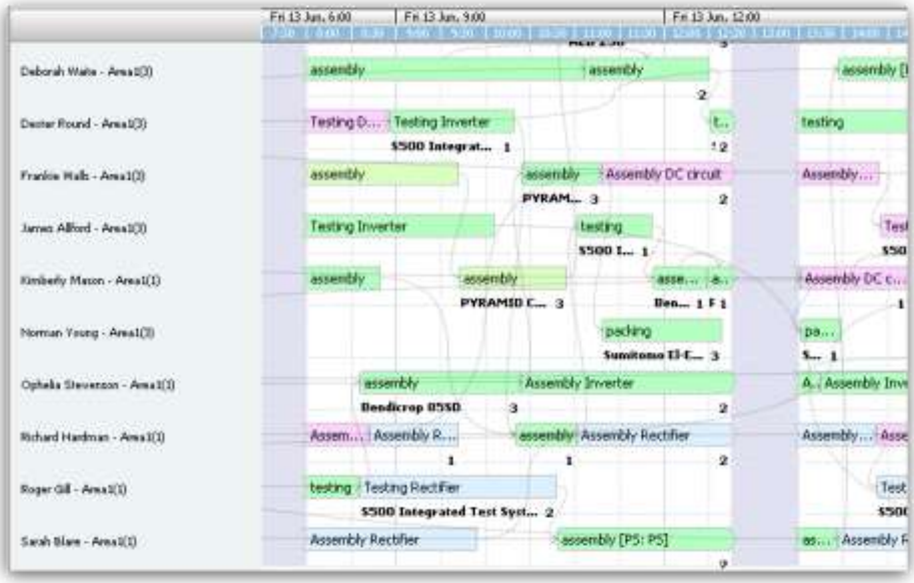


Fig. 11. Schedule of Workers.

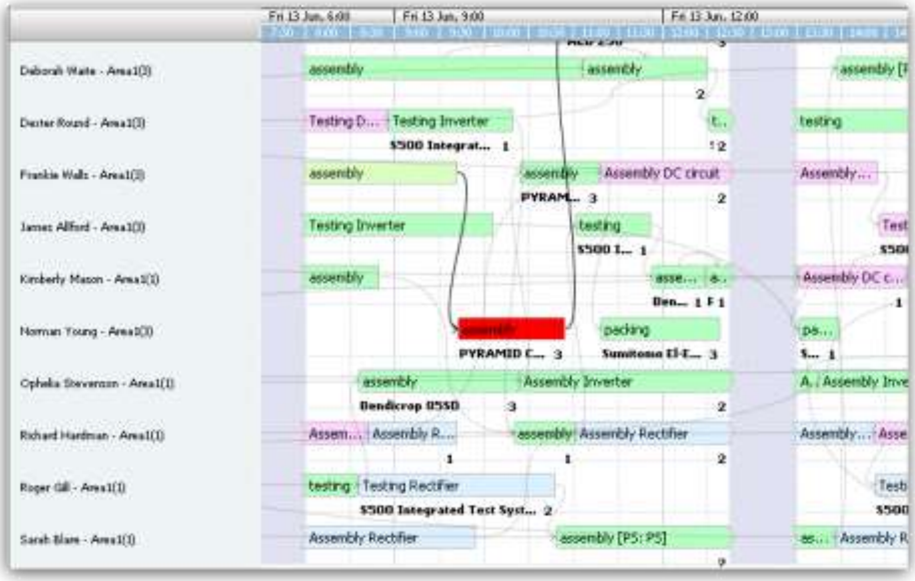
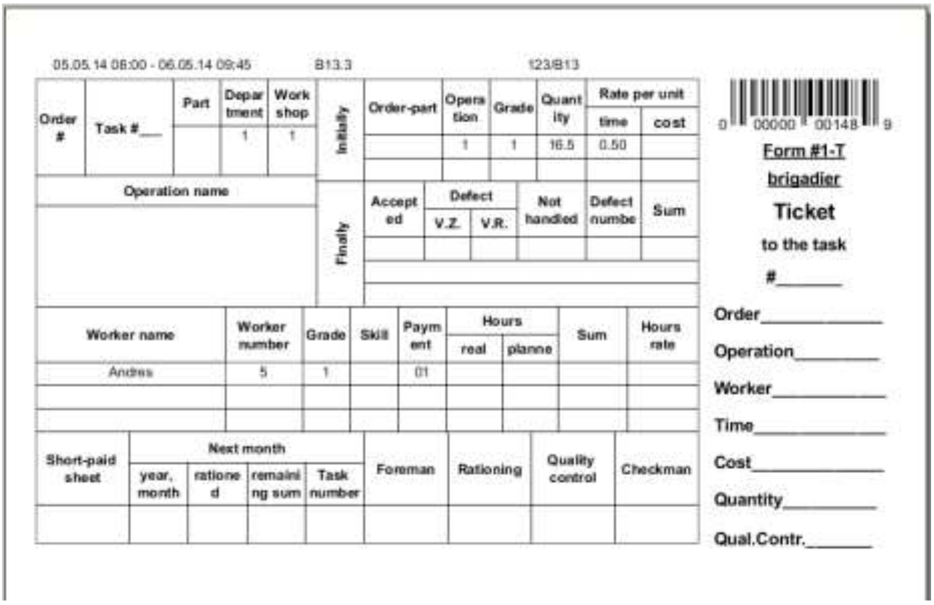


Fig. 12. Screen showing agent of operation signaling about mismatch with worker's skills.



05.05.14 08:00 - 06.05.14 09:45 B13.3 123/B13

Order #	Task #___	Part	Department	Work shop	Initially	Order-part	Operation	Grade	Quantity	Rate per unit: time	Rate per unit: cost
			1	1			1	1	16.5	0.50	

Operation name	Finally	Accepted	Defect: V.Z.	Defect: V.R.	Not handled	Defect number	Sum

Worker name	Worker number	Grade	Skill	Payment	Hours: real	Hours: planned	Sum	Hours rate
Andres	5	1		01				

Short-paid sheet	Next month: year, month	Next month: rationed	Next month: remaining sum	Next month: Task number	Foreman	Rationing	Quality control	Checkman

0 00000 00148 9

Form #1-T
brigadier
Ticket
to the task
#______
Order____________
Operation________
Worker___________
Time_____________
Cost_____________
Quantity_________
Qual.Contr.______

Fig. 13. Screen showing final report of workshop load for a given time period.

The described solution was implemented on J2EE platform using Oracle as a data base solution.

1.3.2.2. Benefits and Assessment

The considered solutions increase efficiency of factory through flexible real time planning of equipment, manpower and materials in real time. It can be applied to any factories, that require individual approach to each order, product or resource, have small production batches, require high workers qualification, have to deal with multiple unexpected events and require high efficiency and flexibility in manufacturing.

The basic “Smart Factory” solution is implemented for JSC “Axion Holding”, JSC “KUZNETSOV” and a few other factories (Skobelev 2011).

The main results of the solution deployment are the following:

- full transparency of day-to-day operations for given time horizon;
- increase of the workshop productivity by 15-20%;
- reduction of the efforts on tasks allocation, scheduling, coordination and monitoring for running orders by 3-4 times;
- increase of the resource efficiency - from 15% and more;
- reduction in response time to unexpected events by 2-3 times;
- increase in percentage of the enterprise orders completed within the given timeframe by 15-30%.

The next R&D step is associated with development of adaptive p2p networks of multi-agent schedulers of workshops which is now under development in the EU integrated project “Adaptive Ramp-Up Management (ARUM)” of FP7 program “Smart Factory” (www.arum-project.eu).

1.3.3 Multi-Agent System for Mobile Fileld Services Scheduling

1.3.3.1 Application Overview

The solution is developed for the Samara regional gas distributor “SVGK” who is operating a large network of gas pipelines and special gas equipment.

Technicians work for the gas pipeline network in small mobile teams on trucks with special equipment servicing gas installations and maintenance, as well as emergency calls from the call center.

The company dispatchers were overloaded with the real-time flow of orders but decisions about team schedules require a lot of specific knowledge about type of technical problem, skills of technicians and workers, required and available equipment, destination point and current position of the team on the map, estimates of delivery time, etc.

Team plans are frequently disrupted by unexpected events such as urgent orders, equipment failure, traffic jam or delay in completing a task, etc.

The client required a real time adaptive scheduler capable of reducing time required to schedule and execute servicing tasks, overall travelling time of service teams and teams utilization during a day.

An adaptive multi-agent solution for scheduling tasks to servicing teams is able to analyze a situation, taking into account orders and teams work load, to select and allocate a preferable team to order, to form a route to destination point and adapt the schedule for the team, to communicate a new plan to the team and then monitor execution of the plan controlling the gap between the plan and reality in real time.

A diagram demonstrating the solution workflow is presented in Fig. 14.

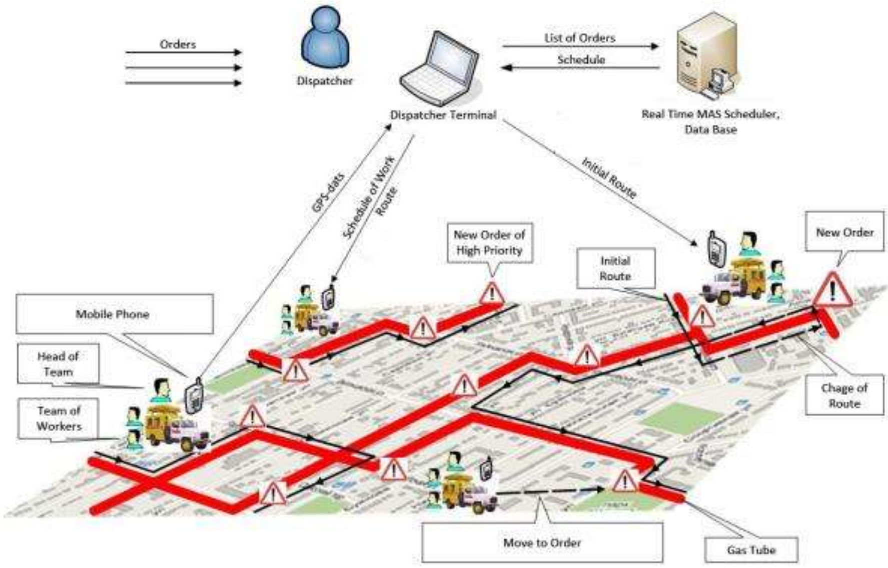


Fig. 14. Workflow of multi-agent solution for mobile teams.

Dispatcher monitors the availability of servicing teams. New orders come to the call center where operators register number of order, address, the type of accident, etc. Then this information enters the scheduler. Priority of order, the urgency and complexity of the required work is automatically determined based on knowledge base and status of available teams. As a result the plan of order execution is formed based on estimates which resources should be involved to solve the problem and what is the best way to reallocate teams.

Examples of multi-agent scheduler screens are given in Fig. 15-17.

List of planned orders and their current statuses is presented on Fig. 15.

Agents of orders and teams take into consideration the priority of orders and try to minimize empty miles between destinations. Schedules are built under control of dispatchers and sent to mobile phone of the servicing teams by texts. Mobile phones are also used for sending progress reports on orders directly to scheduler by text messages.

The screen presented on Fig. 16 is designed to monitor the current location and status of servicing teams on the map. Also it helps to view and check addresses of orders because there are some gaps and other issues on electronic maps which require manual intervention and corrections from dispatchers.

Mobile client is designed to send information about new demands from dispatcher to foreman of servicing team and to send back status of the order execution from foreman to dispatcher. Reports for teams are formed in real time as well as for a required time period (Fig. 17).

The solution was implemented on .NET platform using MS SQL Server 2005 as a database server. But the mobile application is implemented on Java ME platform and can function on mobile devices that support Java/JavaScript and have Internet access via http protocol.

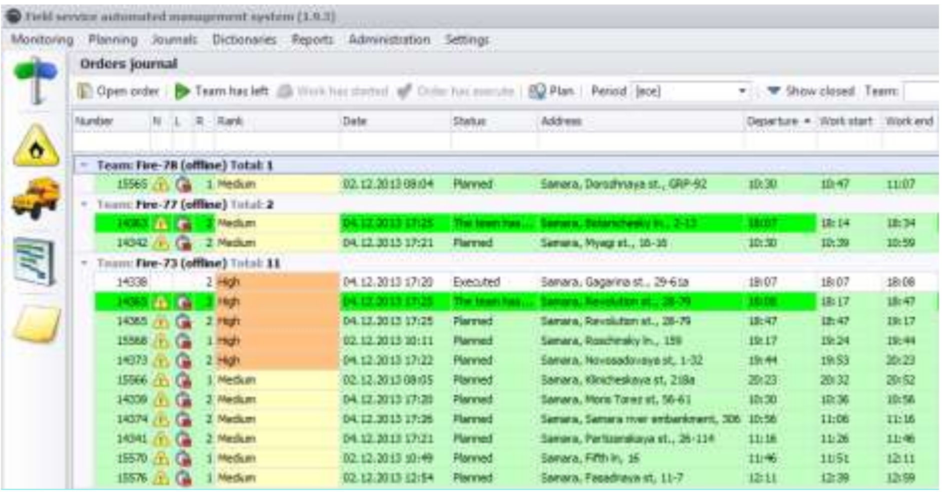


Fig. 15. On-line communications of the system with the teams via low-cost mobile phones: Dispatcher’s screen (late orders are shown in red, planned orders are shown in green).

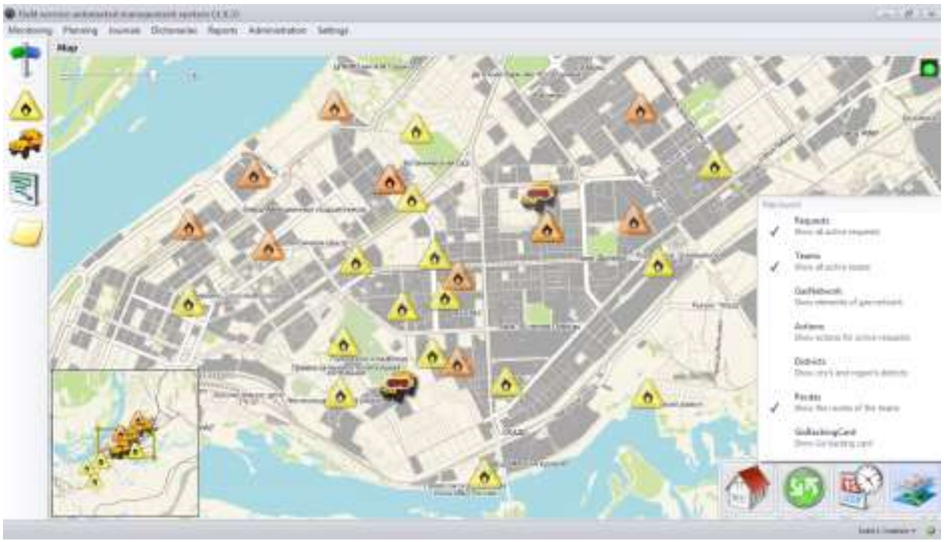


Fig. 16. On-line communications of the system with the teams via low-cost mobile phones: orders and team are shown on the regional map.

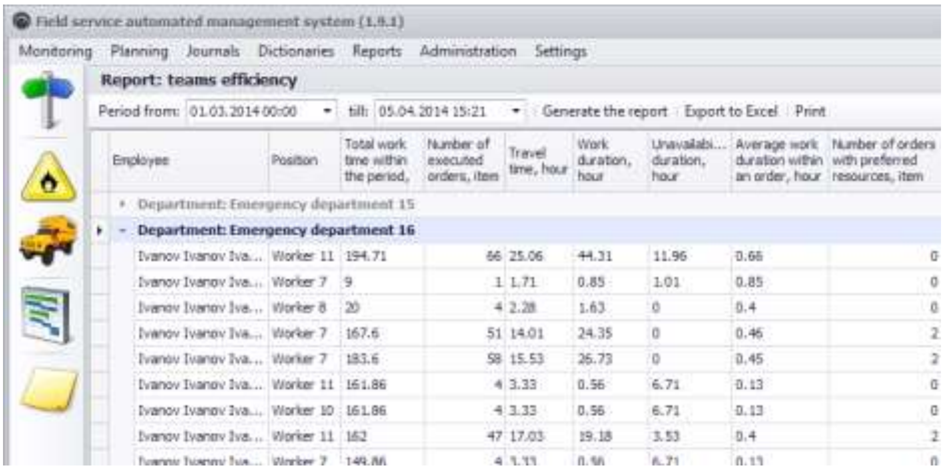


Fig. 17. On-line reports: efficiency of teams.

1.3.3.2 Benefits and Assessment

The adaptive scheduler was implemented in 2011 and used for mission critical functions improving flexibility and efficiency of teams by reducing delivery time, delays and empty miles (Skobelev, 2011).

As a result, the key benefit of the system was 40% increase in mobile teams' productivity. Each team of service engineers has managed to complete on average about 12 tasks a day instead of 7 tasks before system delivery.

In addition, the scheduler enables managers to reduce human factors and have a full transparency of the overall operations of service teams on the map, as well as detailed information on individual team's productivity; the current progress of jobs fulfilment for each team; the number of calls for servicing awaiting to be allocated daily; efficiency of each service engineer and technician, individual costs of every servicing task/operation, etc.

The scheduler has won the "Product of the year" award at the Russian National exhibition "Soft-Tool 2011".

1.4 Discussion

The designed solutions support the shift to real-time economy in corporate resource management by making a very important step from the traditional centralised, monolithic and batch processing systems to real-time adaptive multi-agent systems with an ongoing on-line communication with users.

The industrial applications of the developed multi-agent solutions for adaptive scheduling prove the following benefits for customers:

- allow enterprises to move to real-time economy by analyzing options and taking decisions "on the fly";
- solve complex scheduling problems by replacing combinatorial search with adaptive detecting conflicts and finding trade-offs;
- improve efficiency of resources, quality of service, reduce costs and delivery time, reduce risks and penalties;

- support continuous adaptive re-scheduling in real time with fast reaction to unpredictable events;
- provide an individual approach to every order, operation and resource;
- support coordination by interactions with users in 2-way directions;
- help to reduce the human-factor in the process of decision making;
- enable modeling of "What-if" scenario to optimize decisions;
- create a platform to support the business growth.

The discussed industrial applications of multi-agent systems also help to define future R&D projects to provide more flexible design decisions, better analyse quality and efficiency of real time scheduling, improve performance, etc.

From our point of view, one of most interesting opportunities now is to design adaptive scheduling solutions as a complex adaptive systems in the sense of new theory of complexity based on the concepts of dissipative structures and non-liner thermodynamics of Nobel Laureate Prof. Ilya Prigogine (Prigogine and Stengers, 1984; Nicolis and Prigogine, 1989).

The discovered similarity between self-organised systems in chemistry and developed adaptive schedulers with "unstable equilibriums", chaos and order, oscillations and catastrophes is very inspiring for further R&D works and developing a new generation of adaptive scheduling systems which will potentially provide well-balanced schedules of such a high quality which are equal or even better than schedules created by humans.

The developed approach and tools could be also considered as a basis for designing advanced self-organised systems providing emergent intelligence (Rzevski and Skobelev, 2014).

1.5 Conclusions

This chapter presents results of recent works on development of the industrial multi-agent solutions for the real time adaptive scheduling of resources.

The achieved results prove that multi-agent technology is becoming an efficient industrial solution for real time resource management in those application areas which can be characterised by high uncertainty, complexity and dynamics.

The discussed solutions improve the quality of services for clients and efficiency of resources utilisation, reduce costs and time of delivery.

The experience learnt in the industry also opens up new areas for future R&D works for improving quality and performance of the adaptive real time scheduling solutions.

References

[1] Leung, Y-T, 2004, Handbook of Scheduling: Algorithms, Models and Performance Analysis, CRC Computer and Information Science Series, Chapman & Hall, London.

[2] Shirzadeh Chaleshtari, A, Shadrokh, Sh, 2012, 'A Branch and Bound Algorithm for Resource Constrained Project Scheduling Problem subject to Cumulative Resources', World Academy of Science, Engineering and Technology, vol. 6, pp. 23-28.

[3] Vos, S. 2001, “Meta-heuristics: The State of the Art” in Local Search for Planning and Scheduling, eds A Nareyek, Springer-Verlag, Berlin, pp.1-23.
[4] Wooldridge, M, 2002, An Introduction to Multi-Agent Systems, John Wiley & Sons, London.
[5] Shoham, Y, Leyton-Brown, K, 2009, Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations, Cambridge University Press, New York.
[6] Rolf CR, Kuchcinski, K, 2011, ‘Distributed constraint programming with agents’, in Proceedings of the second international conference on Adaptive and intelligent systems, Springer-Verlag, Berlin, pp. 320-331.
[7] Gongfa, L, 2011, ‘A Hybrid Particle Swarm Optimization Algorithm for Job-Shop Scheduling Problem, International Journal of Advancements in Computing Technology (IJACT), vol.3, no.4, pp. 79-88.
[8] Xueni, Q, Lau, H, 2010, ‘An AIS-based Hybrid Algorithm with PSO for Job Shop Scheduling Problem’ in Proceedings of the tenth IFAC Workshop on Intelligent Manufacturing Systems, Lisbon, pp. 371-376.
[9] Pinedo, ML, 2008, Scheduling: Theory, Algorithms, and System, Springer, Berlin.
[10] Allan, R, 2010, ‘Survey of Agent Based Modeling and Simulation Tools’, Computational Science and Engineering Department, Technical Report DL-TR-2010-007.
[11] Noller, D, Hanis, T, Feldman, M, Asher, J, Bosler, B, 2013, ‘Semantic Models in Manufacturing Execution/Enterprise Systems’, MES – Theory and Practice, vol.6, NGSS, Moscow, pp. 37-74.
[12] Pechoucek, M, Marík, V, 2008, ‘Industrial deployment of multi-agent technologies: review and selected case studies’ Autonomous Agents and Multi-Agent Systems, vol.3, pp. 397-431.
[13] Leitao, P, Vrba, P, 2011, ‘Recent Developments and Future Trends of Industrial Agents’, in Proceedings of the fifth International Conference on Holonic and Multi-Agent systems in Manufacturing, Springer, Berlin, pp. 15-28.
[14] Florea, A, Lauttamus, J, Postelnici, C, Martinez, J, 2013, ‘Agent-Based Control of Operational Conditions for Smart Factories: The Peak Load Management Scenario’, in Proceedings of the sixth International Conference on Industrial Applications of Holonic and Multi-Agent Systems, Springer, Berlin, LNAI 8062, pp. 13-24.
[15] Brussel, HV, Wyns, J, Valckenaers, P, Bongaerts, L, 1998, ‘Reference architecture for holonic manufacturing systems: PROSA, Computer in Industry, vol. 37, no.3, pp. 255-274.
[16] Skobelev, P, Vittikh, V, 2003, ‘Models of Self-organization for Designing Demand-Supply Networks’, Automation and Remote Control, Journal of Russian Academy of Science, vol.1, pp. 177-185.
[17] Skobelev, P, Vittikh, V, 2009, ‘The compensation method of agents interactions for real time resource allocation’, Avtometriya, Journal of Siberian Branch of Russian Academy of Science, vol.2, pp. 78-87.

[18] Rzevski, G, Skobelev, P, 2014, Managing complexity, WIT Press, UK-USA, 198 p.
[19] Skobelev, P, 2011, 'Multi-Agent Systems for Real Time Resource Allocation, Scheduling, Optimization and Controlling: Industrial Applications' in fifth International Conference on Industrial Applications of Holonic and Multi-Agent Systems, Toulouse, France, 2011, Springer, Berlin, vol. 6867, pp. 1-14.
[20] Prigogine, I, Stengers, I, 1984, Order out of Chaos: Man's New Dialogue with Nature. Flamingo, London.
[21] Nicolis, G, Prigogine, I, 1989, Exploring complexity: An introduction. NY: W. H. Freeman.

Published: 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 (Ed.). – Elsevier. – 2015. – pp. 207-230.

https://doi.org/10.1016/B978-0-12-800341-1.00012-7