Vladimír Mařík
Pavel Vrba
Paulo Leitão (Eds.)

Holonic and Multi-Agent Systems for Manufacturing

5th International Conference on Industrial Applications of Holonic and Multi-Agent Systems, HoloMAS 2011
Toulouse, France, August 29-31, 2011
Proceedings

Springer

Multi-Agent System for Scheduling of Flight Program, Cargo Flow and Resources of International Space Station

Anton Ivaschenko[1], Igor Khamits[2], Petr Skobelev[1], and Marina Sychova[2]

[1] SEC "Smart Solutions" Ltd, 95 Sanfirovoy St. Samara, 443080, Russia
ivashenko@smartsolutions-123.ru
[2] RSC "Energy", 4a Lenina St. Korolev, Russia

Abstract. This paper presents the results of application of innovative knowledge-based multi-agent technology for scheduling of flight program, cargo flow, fuel, water, and other resources of International Space Station (ISS). The developed system aims creating solid information space for scientists, responsible for ISS life activity and supply with various resources and provides intelligent scheduling considering volumes of transportation space ships and requirements of ISS life support.

Keywords: International Space Station, ISS, adaptive scheduling, multi-agent systems, distributed decision making, ontology.

1 Introduction

To provide intermediate functioning of International Space Station (ISS) it is necessary to solve lots of interrelated tasks to schedule space flights (including starts, dockings and undockings) of transportation and piloted (manned) space ships considering various requirements, support space crew life activity, deliver laboratory equipment, different material and instruments. Recently, while solving this problem, lots of scientists, engineers and managers had to make millions of iterations of interaction to come to a certain compromise solution to support ISS with all required stuff considering lots of limitations and constraints. To reduce time and increase the effectiveness of such cooperation the experts of Russian Rocket Space Corporation "Energy" together with "Smart Solutions" software engineering company have started in 2009 a project to develop a specialized system for interactive ISS flight program interactive design, and cargo flow and resources intelligent scheduling.

From the very beginning it was decided to develop such system based on modern software development technology to assure intelligence and effectiveness of decision making. As soon as ISS supply chain can be treated as a complex network of continuously running and co-evolving intelligent specialized schedulers, the whole solution has been based on holons paradigm [1] and bio-inspired approach [2], which requires development of new methods and tools for supporting fundamental mechanisms of self-organization and evolution similar to living organisms (colonies of ants, swarms of bees, etc).

V. Mařík, P. Vrba, and P. Leitão (Eds.): HoloMAS 2011, LNAI 6867, pp. 165–174, 2011.

In this paper we would like to present the results of development and deployment of multi-agent system that allows cooperative scheduling of space ships flights and dockings to ISS, cargo flow and utilization, fuel, water, and food support and space crew time management. Agents, often with conflicting interests, dynamically create a schedule though a process of negotiation and distributed decision making. The system was that was developed according to bio-inspired approach and demonstrates effectiveness being applied in such innovative industry as astronautics.

2 Problem Definition and Solution Vision

ISS supply chain scheduling includes the following tasks:

- Flight program design, that results in distribution of space ships' docking to ISS across modules (ISS segments) and time considering various constraints (there should be a certain period of time between operations of docking/undocking, at least one piloted ship should be docked to the station, there are certain preferences to dock ships of one type to certain modules;
- Cargo flow strategic and operational scheduling, that results in distribution of deliveries of units, blocks and systems across flights of transportation and piloted (manned) space ships;
- Fuel deliveries and spending strategic and tactical scheduling based on a forecast of ISS position changes, the Sun activity, operations plan and flight program;
- Water, food and other human life activity support items delivery scheduling based on information about the expeditions and flight program;
- Scheduling of return and utilization;
- Flight crew time scheduling.

Scheduling of flight program, cargo flow and resources of ISS has several stages with different scheduling horizon. First the strategic model of cargo flow is created, that helps calculating a number of required transportation flights per year on the basis of the number of expected expeditions. Than the process of flight program interactive design starts. At this stage the number and times of dockings and undockings of space ships to the modules (segments) of ISS are identified and agreed with all the involved parties considering timeframes of possible starts of space ships, the Sun activity, configuration and expected position of ISS, space crew requirements and etc. Several versions of flight program can be created and examined at this stage until the final plan will be signed-up and published.

After the signed-up flight program is obtained the process of cargo flow, fuel and water scheduling is started in parallel. Cargo volumes are distributed between transportation space flights (and manned flights when special assistance is required) on the basis of data about usual everyday spending. The number of astronauts and information about dates of starts and dockings are taken from the signed-off flight program. Fuel and water deliveries are calculated on the basis of data about the ISS position corrections and spending for docking and other space operations. Cargo flow plan is expanded by the plan of cargoes that need to be taken back to the Earth (returned) and dismissed from the ISS (utilized).

The problem of scheduling here is that all decisions should be made in parallel and coordinated between each other. The volumes of transportation space ships are limited and in case of additional cargo request the fuel or water volumes should be reduced. In turn, if there is a need to increase fuel spending, some cargo delivery should be cancelled or postponed till next flight.

Due to such a nature of concurrent engineering of flight program, cargo flow and resources plans, the automated system of decision support should be designed as an adaptive network of intelligent schedulers. Such schedulers can compete and cooperate, coordinate and adapt their behaviors, aggregate their services to users and take various requirements individually. Each event that occurs here can influence on the whole network and needs a collaborative reaction from all dependable components which take into account personal objectives and constraints of each decision making member.

To provide such cooperation and adaptive nature of decision making each scheduler can be developed based on multi-agent platform [3], that recently has been utilized and demonstrated effectiveness in the area of transportation and production resources scheduling [4, 5]. Multi-agent algorithms can deal with high complexity, function in real time and adaptively react to any change that makes them useful for automated scheduling.

Multi-agent systems (MAS) are based on one of the new information technologies recently arrived on the edge of artificial intelligence, object-oriented programming, parallel processing and telecommunications [6]. MAS are based on the concept of a software agents community – each one is an autonomous object, which can react to events, make decisions and communicate with other agents.

One of the modern approaches of developing multi-agent systems for decision making is a bio-inspired approach, which is based on the concept of holonic systems introduced by Artur Koestler and first implementation of this concept in PROSA system [7]. Holonic Manufacturing System (HMS) paradigm is implemented in ADACOR holonic approach [1] that offers a way of designing adaptive systems with decentralization over distributed and autonomous entities organized in hierarchical structures formed by intermediate stable forms.

This approach allows developing reconfigurable and responsiveness systems by considering such biological inspired techniques as swarm intelligence and self-organization. In addition to an ability to react on sudden and unpredictable changes modern reconfigurable systems should be easy in maintenance and use [8]. The offered architecture is aimed at modular composition of models of multiple agents behavior, capability and self-reflection and enhancing the agent's performance by usage of semantics and knowledge ontology.

3 System Architecture and Multi-Agent World Description

Fig. 1 describes the architecture of multi-agent system. Each scheduler for flight program, cargo flow and resources utilization interactive design and scheduling has special user interface (UI) and organized as a separate engine. Scheduling in each engine is performed by agents of different types.

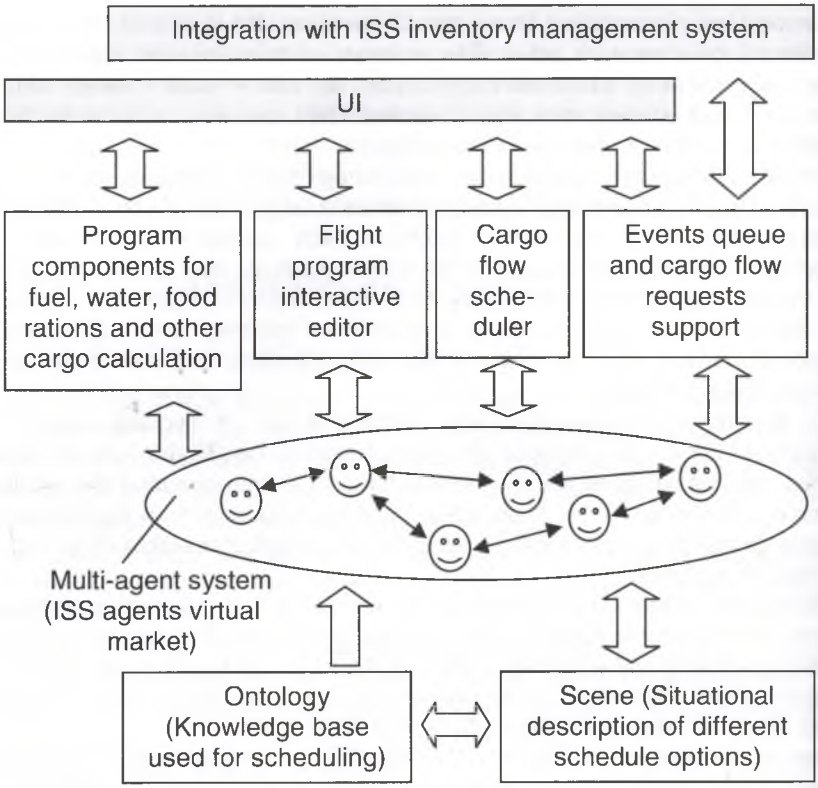

Fig. 1. Flight program and cargo flow scheduling multi-agent system architecture

Each scheduling engine can have agents of own type (for example, flight program is developed by the agents of space ships, expeditions, astronauts, and cargo flow is formed by the agents of cargo and space flight). Still some agents function between 2 or more engines: every two engines should involve agents of at least one common type. This rule helps introducing adaptive interaction between the engines.

For example, the agents of space ship flights are introduced both in flight program design engine and cargo flow scheduling engine. If due to space ship preparation delay in flight program the start of its flight is postponed, the agent of this flight changes its plan, i.e. shifts the dates of start, docking and undocking. As soon as it acts both in flight program and cargo flow scene, it reports about the changes to the agents of cargoes in cargo flow schedule, and cargoes that need to be delivered earlier can "jump" to the flight of ship of another type.

In turn, if some cargoes are reduced (e.g. due to lower than expected onboard utilization) the utilization of a certain flight can become too little, which will be represented in a cargo flow schedule by "unhappy" (that means low loading KPI) flight agent. This agent will try to shift later in flight program to become more competitive and attractive for the agents of cargoes that will try to be scheduled on it.

To provide this type of interaction all the agents are put to solid virtual market and "linked" to one or several decision making engines that form a network of adaptive schedulers. According to this description, there are introduced three types of agents:

- Agents of decision makers, who participate in real world negotiations: cargo managers, engineers and scientists, top management.
- Agents of inanimate objects that can still act as independent entities and have own objectives and constraints: space ships and flights, expeditions, astronauts, cargo items, systems, fuel, water and etc.
- "Group" agents that represent flight program options, cargo flow schedules, options of fuel and water tactical calculations that negotiate on behalf of groups of the agents to produce consistent plans.

All the knowledge including a description of agents' types, preferences and constraints are captured in ontology, which acts as a scheduling knowledge base. The fragment of Ontology is represented on Fig. 2.

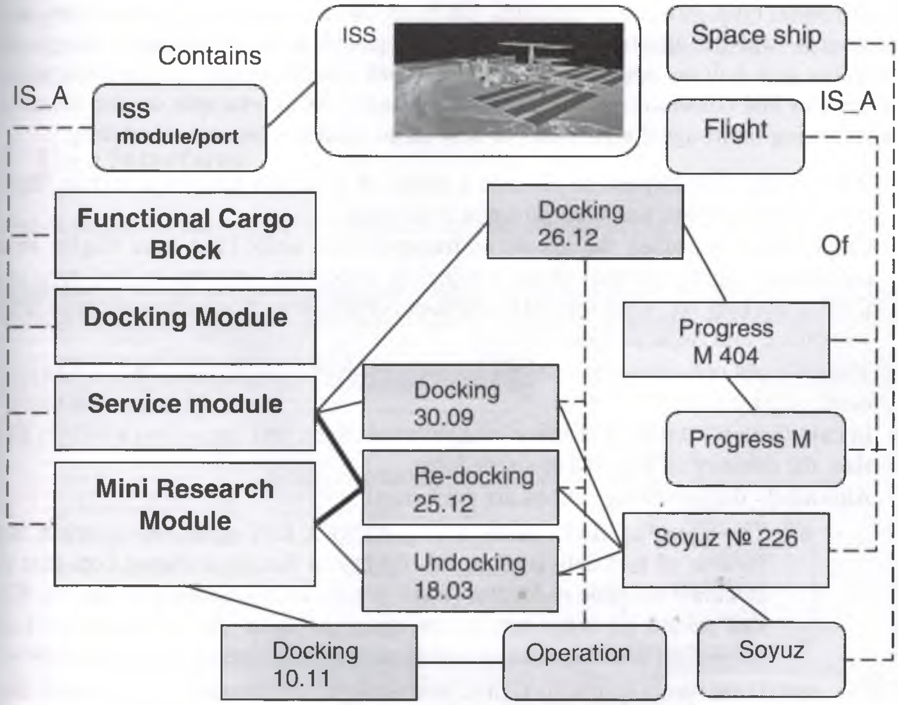

Fig. 2. Flight program Ontology fragment

Ontology is a repository of conceptual knowledge and separates the specific knowledge of ISS domain from the program code. The knowledge is represented in ontology as a semantic network of domain concepts, defined by their attributes and relations. Ontology provides a basis for constructing instantaneous models of domain, known as Scenes.

Special user interface is developed that allows ontology editing without system shutdown. Users can introduce new types of space ships and flights (and specify to what ports they should be docked); new types of ports/modules; new types of cargo and operations, and the UI of flight program and other editors will change correspondingly, providing new capabilities for interactive scheduling.

4 Agents' Interaction and Planning Strategy

To implement the proposed solution the system was divided to several decision support subsystems, each of which was implemented using multi-agent technology. These subsystems include flight program design, cargo flow scheduling, fuel and water calculation, returns and utilization scheduling and flight crew time scheduling. The list of components can be easily expanded in case new decision making support is required.

Each of these subsystems forms a separate community of agents, where the agents of different type can communicate. Each agent has its own preferences and constraints, still the agents' community being created in an autonomous subsystem according to a holonic approach can interact with another agents' community using preferences and constraints of the whole community. As an example we can describe the following multi-agent algorithms of new cargo curator's request scheduling:

- When some new request to allocate a cargo of a certain time to a certain flight comes from curator, a new cargo agent is created;
- Cargo agent examines the agents of transportation units (that have flights in a considered flight program) about a principle allocation opportunity and receives feedback – here top-level messages are sent between the subsystems of cargo flow scheduling and flight program design;
- The obtained options are prioritized according to value and correspondence to new cargo requirements;
- In case there is enough free space on a selected flight, and cargo fits in weight and size, the delivery is included to a schedule.
- Alternately the following actions are performed:
    - The agent of flight sends a request to a fuel agent. As a result the volume of fuel delivered on this flight can decrease. Please note that to calculate possible reduction (there is pre-defined amount of fuel on ISS that should be supported in any case) the agent of fuel addresses fuel subsystem (that has own ontology and types of agents) for recalculation.
    - If the proposed reduction is not enough, water agent is requested and acts in a similar way.
    - In case fuel and water reduction in not enough, the agents of other cargos are pushed to move to other flights if possible.

Two features here are of high importance:

- One cargo can be linked to the others (they can be located to one box or be parts of a single unit). In this case when one cargo moves to another flight, it should consider the interests of the others in gear. So they will move only altogether and only in case the new cargo will bring higher profit to the flight.

- When one or several cargoes move, they free some space that can be greater, than it is required to allocate a new cargo. Such an event initiates rescheduling of water and fuel: in case their agents previously had to reduce their volumes, now they have an opportunity to restore.

Cargo agents make decisions to maximize their key performance indicators (KPIs) that are predefined for each agent individually on the basis of cargoes characteristics and preferred/acceptable delivery time intervals. KPI function is determined considering cargo priority: life activity support items have the highest one.

In such an algorithm it is impossible to predefine any steps' sequence or determinative logic. In this respect it can hardly be even identified as an algorithm – better to treat it as a number of principles. The agents perform repetitive interactions involving other agent's communities supported by specialized subsystems and introduce fluctuating corrections of their schedules.

This behavior absolutely corresponds to a scheduling process in real life, when decision makers, responsible for cargoes, fuel and water argue on suitable allocation using a round table. That's why we treat the proposed logic of agents' interaction as bio-inspired and have in view that its implementation is adequate and helpful for decision makers.

5 User Interfaces

The examples of the system's workspace are represented at Fig. 3 – 6.

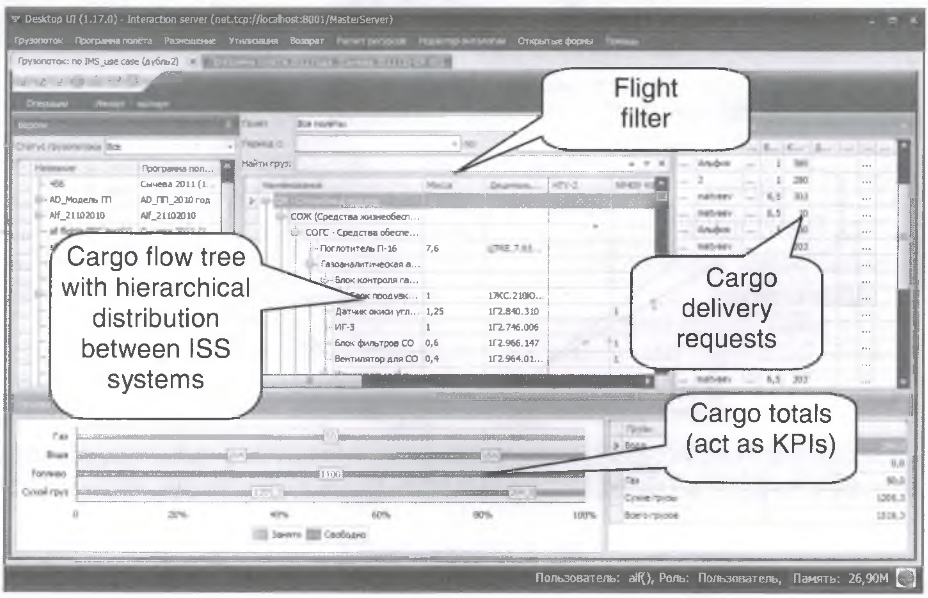

Fig. 3. Cargo flow delivery plan

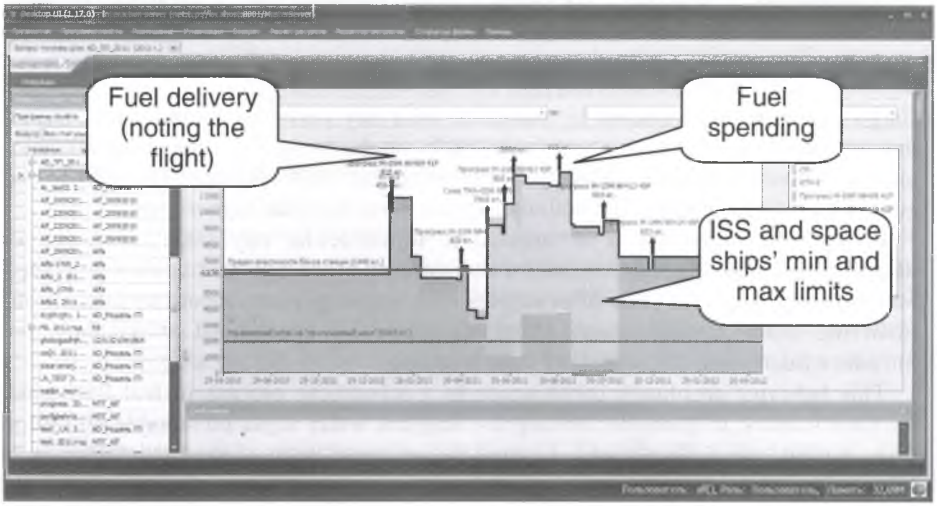

Fig. 4. Fuel delivery/spending tactical balance

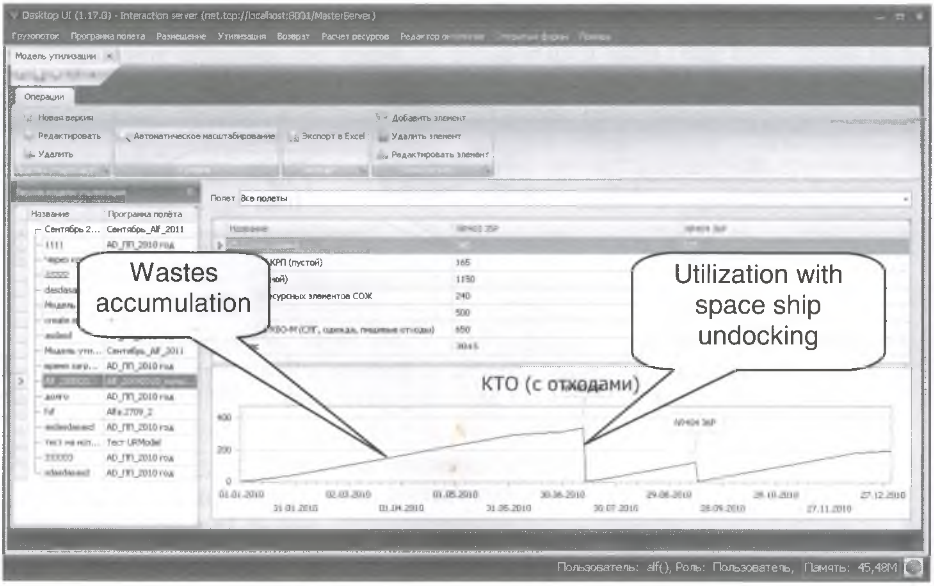

Fig. 5. Utilization balance

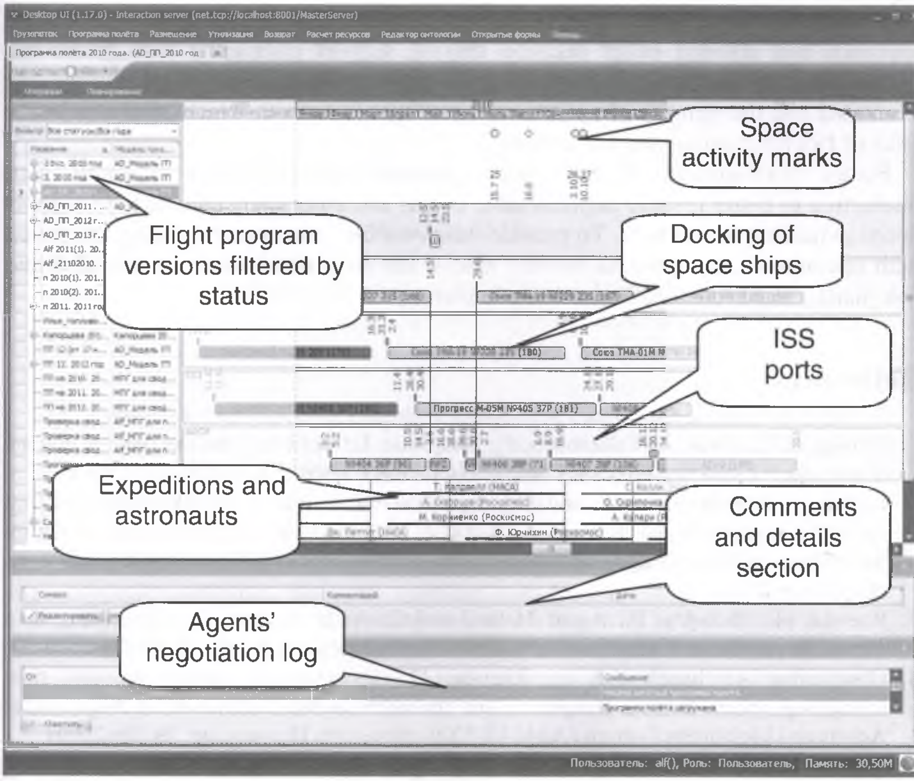

Fig. 6. Flight program interactive design editor

The described above approach was implemented on .NET platform using MS SQL Server 2005 as a database server.

Using the described system several flight programs were designed for a period of 2010 – 2014, cargo flow schedule and resources calculations were made for 2011. The main system's feature is that is allows designing and comparing several options, including possible reaction modeling to unpredictable events that is very important in astronautics problem domain.

The basic effect was detected in a reduction of time spent on scheduling and the resulting ability to simulate different schedule options and support negotiations between the involved departments to find better reaction to external events. This helps minimizing possible risks and better prepare to unpredictable events.

Integration with inventory management system (using import and export files of specific format) allows updating the created schedules with actual data and initiate re-scheduling as a reaction in real time.

6 Conclusion

This paper presents the results of industrial implementation of multi-agent system for flight program, cargo flow and resources interactive design and scheduling of

International Space Station (ISS). The system was built according to bio-inspired approach that allowed better decision making support comparing with traditional systems, supporting hierarchical workflows. As a result the time of scheduling was decreased and the number of options analyzed was increased, which results in lower risks of ISS functioning and life activity.

Future development of this system considers improvements in multi-agent interaction to better react to unpredictable events and emergency situations to support decision making in real time. To provide this possibility the system is being integrated with operational management system, special ontology extensions are provided and risk management functionality is being implemented.

References

1. Brussel, H.V., Wyns, J., Valckenaers, P., Bongaerts, L.: Reference architecture for holonic manufacturing systems: PROSA. Computer in Industry 19998 37(3), 255–274 (2008)
2. Leitão, P.: Holonic rationale and bio-inspiration on design of complex emergent and evolvable systems. In: Hameurlain, A., Küng, J., Wagner, R. (eds.) Transactions on Large-Scale Data- and Knowledge-Centered Systems I. LNCS, vol. 5740, pp. 243–266. Springer, Heidelberg (2009)
3. Rzevski, G., Skobelev, P.: Agent Method and Computer System For Negotiating in a Virtual Environment. Patent No: WO 03/067432 A1 (published August 14, 2003)
4. Glaschenko, A., Ivaschenko, A., Rzevski, G., Skobelev, P.: Multi-agent real time scheduling system for taxi companies. In: 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2009), Budapest, Hungary, pp. 29–36 (2009)
5. Yankov, I., Rzevski, G., Skobelev, P.: A multi-agent scheduler for rent-a-car companies. In: Mařík, V., Strasser, T., Zoitl, A. (eds.) HoloMAS 2009. LNCS, vol. 5696, pp. 305–314. Springer, Heidelberg (2009)
6. Wooldridge, M.: An Introduction to Multi-Agent systems. John Wiley and Sons, Chichester (2002)
7. Leitão, P., Valckenaers, P., Adam, E.: Self-adaptation for robustness and cooperation in holonic multi-agent systems. In: Hameurlain, A., Küng, J., Wagner, R. (eds.) Transactions on Large-Scale Data- and Knowledge-Centered Systems I. LNCS, vol. 5740, pp. 267–288. Springer, Heidelberg (2009)
8. Terzic, I., Zoitl, A., Rooker, M., Strasser, T., Vrba, P., Mařík, V.: Usability of multi-agent based control systems in industrial automation. In: Mařík, V., Strasser, T., Zoitl, A. (eds.) HoloMAS 2009. LNCS, vol. 5696, pp. 25–36. Springer, Heidelberg (2009)