

MANAGING AIRCRAFT LIFECYCLE COMPLEXITY

G. RZEVSKI¹, J. KNEZEVIC², P. SKOBELEV³, N. BORGEST⁴ & O. LAKHIN³

¹The Open University, UK.

²MIRCE Akademy, UK.

³Smart Solutions Ltd, Russia.

⁴Samara Aerospace University, Russia.

ABSTRACT

A thorough analysis of data on aircraft lifecycle revealed inadequacy of current lifecycle management methods in the face of increased complexity of the Internet-based global market. A new method for managing lifecycle has been developed by authors and their teams using concepts and principles of the emerging complexity science with the aim of reducing lead times and costs. Centralised control has been replaced with distributed decision-making empowering all lifecycle stakeholders. The solution described in this paper is the first of its kind and it represents a genuine advance in knowledge, which leads to considerable reduction in design/production lead times and decrease in the lifecycle cost. The method has been validated in a variety of applications.

Keywords: aircraft, complex adaptive systems, complexity, lifecycle management, real-time spare parts management.

1 INTRODUCTION

The lifecycle of a large aerospace product such as an airliner or a military aircraft consists of a very large number of interconnected activities, which are normally partitioned into several phases such as: *Design, Production, Operation* and *Decommissioning*. Each phase of the lifecycle consists of activities as outlined below.

Design consists of all activities needed to produce, operate and recycle a product, including: market research, requirements specification, conception, design management, overall design (including design for X, where X is performance, safety, reliability, operability, manufacturing, assembly, maintainability and recycling) and design of components.

Production includes the selection and development of production technologies, production planning and management, supply of materials and components, manufacturing, assembly, testing and quality assurance.

Operation includes the scheduling and management of the use of aircraft and its crew within a fleet; the maintenance and repair tasks and resources; and the logistics support tasks and resources.

Decommissioning includes safe disposal and recycling of components and/or materials.

The current practice is to consider each phase of the lifecycle separately managed by a different team of specialists, often from different organisations, with a limited interaction between them. As a rule, there is no central lifecycle management team charged with coordinating all activities from the creation of the idea of the future product to its disposal and recycling. Moreover, each phase of the lifecycle is normally supported by stand-alone computer systems, such as computer-aided design, computer-aided engineering, computer-aided manufacturing, enterprise resource planning and various production and logistic schedulers, which are rarely fully compatible with each other and never integrated into a single lifecycle system.

Partitioning of lifecycle activities into individual phases and managing each phase by different management systems supported by stand-alone computer systems weakens important connections between key lifecycle activities and thus delays useful flow of information, particularly feedbacks

from manufacturing, assembly, operation, maintenance and support to design. As a consequence, designers have no real-time access to data on failure analysis, which would help them to better understand the relationship between failures and design decisions.

The safety record of aircraft industry is excellent. Our aim is to identify areas where lifecycle costs could be significantly reduced by improving management of lifecycle complexity while maintaining or improving current levels of safety.

2 LIFECYCLE OF LARGE AIRCRAFT

Failure analysis conducted by one of the authors (JK) in support of this project [1, 2] has established several surprising features.

Firstly, each aircraft has a unique individual failure pattern, which depends on how it is used, how it is maintained and how it was supported. Operational conditions play a significant role: aircraft that is often scheduled to fly to Middle East will be subject to failures due to the presence of sand and high ambient temperatures whilst aircraft regularly scheduled to fly to the North will suffer from different types of failures due to low temperatures, snow and ice.

Secondly, aircraft failures are interdependent; a failure of one component may cause failures of others depending primarily on proximity of locations and also on functional links. A typical case is locating all operational and redundant components of a hydraulic system next to each other, which occasionally cause all of them to fail as a result of a single external event.

And finally, the timings of component failures are unpredictable but not random; patterns of failures can be discovered from failure data; these patterns are not permanent – they change as operating conditions of aircraft change and with the age of aircraft.

These findings clearly show that current reliability assessment methods can be significantly improved. Consider the following:

1. If each individual aircraft has a unique life, is it correct to assess reliability and organize identical maintenance for all aircraft manufactured to the same design?
2. If failures are interdependent, is it correct to calculate failure rates ignoring spatial interdependence?
3. If a pattern of failures change in time, does it make sense to prescribe maintenance procedures for the whole life of aircraft?

The situation as described above is plainly wrong, ineffective and wasteful but it is accepted due to high *complexity* of the lifecycle of large aerospace products as, currently, no individual or a team is capable of comprehending let alone managing the interactions between the multitudes of diverse elements of the lifecycle process.

Let us illustrate the complexity of aircraft lifecycle taking as an example airliner B747, one of the largest aircraft in the world, which began operation in 1970 and was since then sold to 170 airline companies. A B747 has 6 million components of which 3 million are rivets. Aircraft has 5,000 to 6,000 safety, reliability and maintenance significant modules, which need to be regularly checked, tested, calibrated, cleaned, repaired or replaced. To safely deliver one flying hour, it requires 12 maintenance man-hours (which compares with Concord that required 137 man-hours for each hour of flying).

The following data are taken from the maintenance logbook of the first Boeing 747, which started in-service life in 1970 as the Pan Am asset with a tail number N747PA. During the 22 years of in-service life, the list of major operation and maintenance statistics collected is shown below:

- Consumed more than 271 million gallons of fuel.
- Flown 37 million miles.
- Carried 4 million passengers.
- Been airborne 80,000 hours.
- Made 40,000 take-offs and landings.
- Gone through 2,100 tyres.
- Used 350 brake systems.
- Been fitted with more than 125 engines.
- Had 806,000 man-hours of maintenance.
- Had the metal skin on its superstructure, wings and belly replaced five times.
- Used 9,980 individual X-ray frames of film for structural inspection for metal fatigue and corrosion.
- Had the passenger compartment and lavatories replaced four times.
- Needed 50,000 of man-hours of work for each major overhaul.

Newer aircraft have smaller number of constituent components mainly due to the use of composite materials, and longer intervals between major services. Recently designed aircraft have the so-called heavy maintenance visit (HMV) every 10 years instead of 5 years for aircraft designed in the 1970s. Nevertheless, major services still cost in the region of \$1 million and aircraft is out of service for 1–2 months. Our research shows that there is a considerable opportunity for further improvements in lifecycle costs.

It is important to note that the complexity of aircraft lifecycle increases with time.

Factors driving the increase in lifecycle complexity include:

1. The recent sharp increase in complexity of the Internet-based global market, which makes the forecasting of demand and supply of air transport unreliable.
2. Strong competition between aircraft and engine manufacturers, which creates pressure to innovate with a view to producing improved fuel-efficient aircraft offering more comfortable and more cost-effective transportation service and at the same time reducing concept-to-market lead times.

Shorter lead times put a considerable pressure on designers and manufacturers because of the resulting demand for increased speed of decision-making, which combined with volatility of demand and supply makes a strong case for developing much improved decision support systems.

3 THE CONCEPT OF COMPLEXITY

It is now time to reconsider the lifecycle management of large aircraft and look at it through the lenses of Complexity Science [3–7], which, although very new, has already reached sufficient maturity to provide much needed help.

Complexity is defined as a property of an open system that consists of a large number of diverse, partially autonomous, richly interconnected components, often called Agents, has no centralised control, and whose global behaviour emerges from the intricate interaction of agents and is therefore uncertain without being random.

Complex systems can be distinguished from other systems by the seven features: connectivity, autonomy, emergence, nonequilibrium, nonlinearity, self-organisation and co-evolution.

Connectivity. Agents are interconnected. Complexity of the system increases with the number of links that connect agents to each other. Complexity also depends on the strengths of links. The

weaker the links between agents, the easier is to break them and form new ones, which increases system complexity.

Autonomy. Agents have certain freedom of behaviour (autonomy), which is always limited by norms, rules, regulations, and/or laws. The increase in autonomy of agents increases complexity and if all constraints on agent behaviour are removed, the system switches from complex to random behaviour. Inversely, if autonomy of agents is reduced (by tightening of laws and regulations), the system complexity will decrease, and in the extreme, the system will become deterministic. Complex systems have no central control.

Emergence. Behaviour of complex systems emerges from the interactions of agents and is not predictable and yet it is not random. Uncertainty about the outcome of agent interactions is always between 0 and 1. Emergence, in general, denotes a property of a system that is evident in the system as a whole but it is not present in any of its components.

Nonequilibrium. Complex systems are subjected to perpetual change experienced either as a succession of discrete disruptive events or as a slow, imperceptible drift into failure. Frequency of disruptive events varies with complexity. In systems of high complexity disruptive events occur so frequently that the system has no time to return to stable equilibrium before the next disruption occurs. When the complexity levels are very high, the system is said to be at the edge of chaos because the uncertainty of behaviour is close to 1.

Nonlinearity. Relations between agents are nonlinear. Nonlinearity may amplify a small, insignificant disruptive event and cause a catastrophic outcome (an extreme event), the property called butterfly effect. Butterfly effect increases with complexity. In complex systems, outcomes are, as a rule, consequences of numerous interacting causes, and therefore the cause–effect analysis is inappropriate.

Self-organisation. Complex systems have a propensity to react to disruptive events by autonomously self-organising with the aim of eliminating or, at least, reducing the consequences of the disruption, the property called adaptation. Self-organisation may also be caused autonomously by a propensity to improve own performance, the property called creativity or innovation. To initiate and perform adaptive and creative activities, the system must be intelligent. Intelligence, adaptation and creativity are emergent properties exclusive to complex systems and their levels increase with complexity.

Co-evolution. With time, complex systems change as their environments change and, in turn, they change their environments. Co-evolution is irreversible.

The above definition of complexity perfectly fits an aerospace product lifecycle, where thousands of design and production engineers and managers and tens of thousands of operation, maintenance and support staff, with different skills and different autonomy, interact with each other without overall centralised control and where millions of components, characterised by different lifecycles and subject to different conditions of failure, have to be predicted first and then manufactured, supplied, assembled and tested in order to be used under different operating conditions and maintained in different geographical locations by thousands of engineers and technicians each with different skills and many from different working cultures. In addition, all of these tasks and activities are regulated and communicated in English language, which is not a mother tongue for over 80% of the world population.

Complexity of the design process, in which thousands of designers with hundreds of diverse skills cooperate on projects involving structures consisting of millions of components, is also very high. Under such conditions, it is very difficult to trace late design changes and design errors consequences of which can be very serious if not rapidly identified and corrected.

4 A NEW SOLUTION

A new solution to the lifecycle management problem, which has been developed by authors and their teams, is based on the assertion that a complex system can be managed only by another complex system [8].

Therefore, the new Product Lifecycle Management (PLM) system has been designed as an *adaptive network of computational services* based on multiagent technology rather than currently used monolithic computer programs. Nodes of this network are elementary lifecycle services, which are being developed independently, one by one, and then configured into lifecycle applications. Each application is a unique adaptive network of services aimed at delivering a lifecycle management function.

A system or service, which manages a complex system, must have *requisite granularity and requisite complexity* [8].

Requisite granularity for PLM Service is granularity, which enables the service to identify, store and process every single constituent component (millions of them) as well as all design, production and operation resources. The service of this size cannot possibly be designed as a single integrated unit. The trick is to discover processes that are reasonably self-contained to form elementary services that are implemented using multiagent technology, and to ensure that these services, which are effectively swarms of agents, cooperate and/or compete with each other through a process of negotiations [8].

Requisite complexity for PLM Service is complexity, which enables the management service to adapt to every change of state in product lifecycle. This requirement contradicts traditional view that complex systems can be simplified by partitioning them into self-contained constituent subsystems, which can then be considered separately, one by one, the approach known as reductionism. Reductionism is of course valid for deterministic systems and may work for systems of very low complexity, but for levels of complexity that businesses, including aerospace business, currently experience, the approach is not valid. Instead of simplifying the reality, we must build complexity into management systems to match the complexity of their physical reality and corresponding economic environment.

Complexity has been designed into PLM Service by giving partial autonomy to constituent agents with a mandate to negotiate among them how to go about solving every task at hand, arranging for agents to consult Lifecycle Knowledge Base before undertaking any action and by ensuring that links among agents are easily modified when required. Methods used for designing complexity into PLM Service are described in detail in [8].

A PLM service of sufficient granularity and complexity can be implemented using ontology-driven multiagent technology, as depicted in Fig. 1.

Lifecycle Knowledge Base contains all conceptual knowledge (ontology) and data required to manage the whole lifecycle.

Virtual World consists of services, which are real-time multiagent swarms each dedicated to a particular lifecycle management function (strategic function, design, production and operation), which is a node in an adaptive network of services. Each of these services consists of interconnected elementary services that are designed to communicate with each other and access Lifecycle Knowledge Base via Lifecycle Network. Elementary services can be located at different geographical locations.

In principle, there is a fundamental difference between the design phase of a product lifecycle and all the other phases (production, operation and decommissioning). Design is about creating a product and its in-service life, which does not yet exist. At the beginning of the design phase, we have no

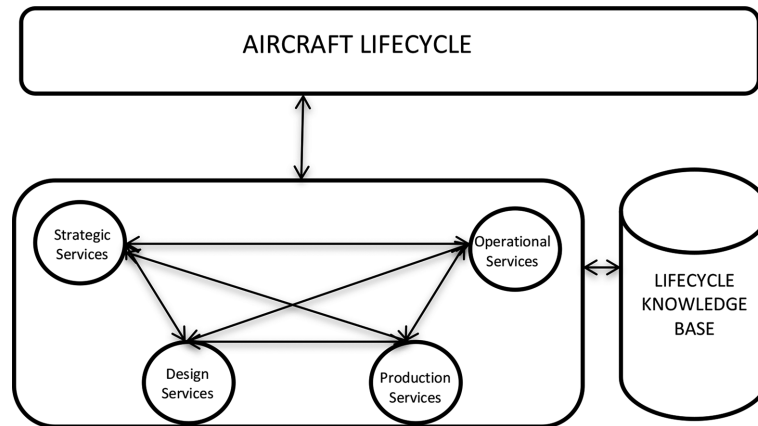


Figure 1: Architecture of an adaptive aircraft product lifecycle management service.

components, no configuration and no ontology. What we have is an incomplete set of candidate product elements (components, configurations and ontologies), which may be used during the design phase.

Design activities can then be divided into two categories:

1. Creating new product elements.
2. Selecting existing product elements.

Whilst the creation of new elements is currently best done by human designers, the selection of existing aircraft elements can benefit from the automation.

We should not forget the importance of design evaluation. Here again we can distinguish two categories:

1. Evaluating how design meets lifecycle specification and aesthetic criteria.
2. Evaluation of design consistency (not least checking for possible clashes when design is done by a large number of different designers).

And again, the first category requires input of human design evaluators whilst the second category is more efficiently performed by multiagent software.

Furthermore, designers must have rapid access to knowledge how past designs worked in-service and therefore we have developed an advanced Adaptive Design Knowledge Discovery Service.

The development of a PLM service is evolutionary. The initial Knowledge Base can be rudimentary and can be expanded if and when required. Services can be added one by one thus enabling the system to be used and tested in real-life application from the very beginning of its development, and to grow as required.

The following PLM services have been specified to form the main nodes of the PLM Service (see Fig. 1):

- Strategy services
 - Adaptive forecasting service
 - Adaptive long-term planning service

Design services

- Adaptive component selection service
- Adaptive design consistency service
- Adaptive design knowledge discovery service

Production management services

- Adaptive supply chain scheduling service
- Adaptive production scheduling service

Operation management services

- Adaptive failure forecasting service
- Adaptive aircraft maintenance and repair scheduling service
- Adaptive personnel maintenance and repair scheduling service
- Adaptive aircraft support and resource scheduling services
- Adaptive spare parts storage and scheduling service

This list is not exhaustive. It includes services that are most likely to improve productivity and reduce costs of aerospace product lifecycles. Decommissioning phase will be considered at a later stage.

The above PLM services are, in turn, composed from the following elementary services:

- Adaptive human resources scheduling service
- Adaptive physical resources scheduling service
- Adaptive transportation scheduling service
- Adaptive loading service
- Adaptive routing service
- Adaptive knowledge discovery service
- Adaptive simulation service
- Adaptive components selection service
- Adaptive design consistency service

Each constituent service of the proposed PLM service must be ‘complex adaptive’, i.e. it must be able to self-organise in reaction to any disruptive event aiming to eliminate or, at least, reduce the consequences of the disruption before the next disruptive event occurs. Since the task of services is to allocate resources to demands and disruptive events are normally all related to unexpected changes in demands or in availability of resources, the self-organisation will always result in re-scheduling.

5 TECHNOLOGY FOR DESIGNING COMPLEX ADAPTIVE SERVICES

At present, the only technology available for designing complex adaptive schedulers is multiagent technology. However, to realise the full potential of this approach, it is essential to pair it with *knowledge technology*, which is generally less developed. Complex adaptive software must have a knowledge base where domain knowledge is stored and where this knowledge may be updated as conditions under which software works change. In addition, it is necessary to have a *knowledge discovery* system capable of dynamically extracting useful patterns of system behaviour from ever changing data.

Agents should be designed as shells without content which, when triggered into existence, consult *life-cycle knowledge base* to learn what is expected from them to do and what *options* they may pursue in order to fulfil their tasks. Having options from which they can choose their actions gives agents a certain decision-making autonomy. For example, resource agents may decide to passively

wait for messages from demand agents offering them opportunity to utilise the resource for which they are responsible or to be proactive and seek appropriate demands to offer them their services. Note that a switch from passive to proactive mode always improves the utilisation of the resource for which the agent is responsible but may, or may not, increase the overall enterprise value and therefore needs to be carefully assessed before the switch is made.

In addition, complexity of the software has to be finely tuned by adjusting the agent *connectivity*. In systems where every agent is connected to every other agent, complexity is excessive, which may lead to undesirable propagation of oscillations caused by repeated attempts by agents to improve matches between resources and demands. The number of active connections between agents can be kept under control by dynamically preselecting agents that would benefit by being involved in the current negotiation. It is also necessary to keep strict control over the length of agent negotiations by allocating to agents appropriate amount of *virtual money*, which they will use to pay for participating in negotiations. Then, changing the allocated amount of virtual money will control the length of negotiations.

To ensure that agents have propensity to keep improving system performance when they are not engaged by disruptive events, we have introduced the concept of *agent satisfaction*. A satisfied agent is the agent, which has managed to negotiate the best possible deal for the resource/demand that the agent represents (e.g. negotiating the full asking price for the transport of spare parts to the location where the aircraft is being serviced). After completing negotiations, the satisfied agents can remain idle. A partially satisfied agent will not be allowed to idle – it will continue proactively to attempt to improve previously agreed deals until it achieves complete satisfaction or runs out of time (e.g. the next disruptive event forces it to start new negotiations).

Results of an agent negotiation to allocate spare parts cargos to trucks, which show agents with different levels of satisfaction, is depicted in Fig. 2. Note that although some individual agents are less satisfied with the results, the Company Agent is very pleased.

From the above discussions, it should be clear that the design of complex adaptive software is an art as much as science. To achieve desired adaptability, complexity of software must be *tuned* and tuning heuristics is derived and polished through experimentation and experience.

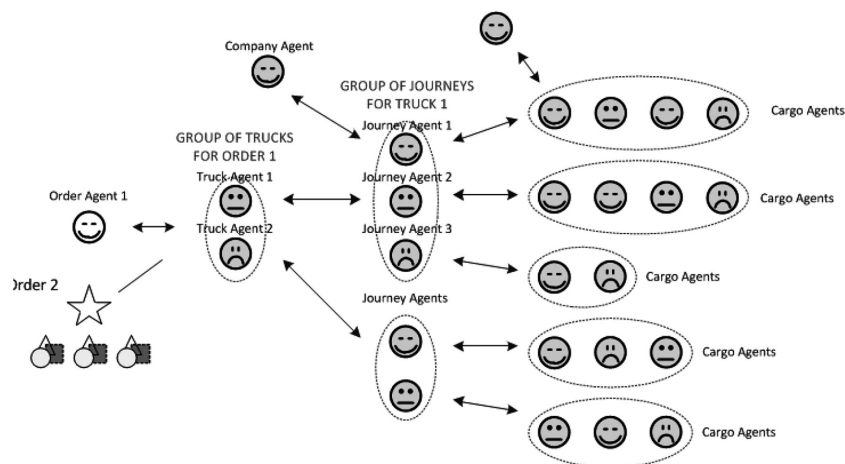


Figure 2: The allocation of spare part cargos to trucks shows the best achievable solution under circumstances with different agents achieving different levels of satisfaction.

6 RESULTS

Elementary services listed in Section 4 of this paper have been thoroughly tested in various commercial applications, many of which are in continuous use, as exemplified by: Real-Time Scheduling of Cargo Delivery to International Space Station [8]; Real-Time Evaluation of Design Consistency [8]; Real-Time Scheduling of Supply Chains for LEGO [9]; Real-Time Scheduling of 2,000 Taxis in London [10]; Real-Time Scheduling of Car Rentals [11]; Real-Time Scheduling of Road Transportation [12, 13]; Real-Time Scheduling of Production [14, 15]; Real-Time Scheduling of Servicing Personnel [16]; Real-Time Scheduling of Mainline Railways [17] and Modelling of Aircraft Functionability [18].

7 CONCLUSION

The new solution to the existing problem of aerospace products lifecycle management is based on proven methods of complexity science, which guarantee considerable economic gains when used in complex environments.

The advantages offered by the proposed solution are summarised below:

1. Every human or physical resource, as well as the enterprise as a whole, is considered as a stakeholder and represented by a personal software agent. Schedules and forecasts are created by negotiation among agents until a consensus is achieved. This ensures distributed decision-making and empowerment of stakeholders.
2. Computational searches for optimal solutions, which work in batch mode and require considerable time (typically 8–10 h) are replaced by a rapid exchange of messages between agents, which speeds up the resolution of conflicts. Schedules and forecasts are therefore created orders of magnitude faster than by traditional optimisers.
3. Agents consult the knowledge base before acting and consequently services are driven by knowledge rather than data, exhibiting emergent intelligence. Agents work 24 h a day either creating solutions or improving previously created solutions.
4. Scheduling services work in real time, which means that each is capable of rescheduling allocation of resources with minimal changes to the previously agreed schedules whenever a disruptive event occurs (e.g. arrival of an unexpected demand or nonarrival of an expected demand). It follows that the lifecycle management system *adapts* to changes in the lifecycle that it manages.
5. Forecasting services (e.g. forecasting of demand and forecasting of failures) are dynamic, which means that they are capable of adjusting forecasts in real time whenever a disruptive event occurs that invalidates the previously agreed forecast.
6. All software services are capable of (a) interacting with each other, (b) accessing the same knowledge base and (c) delivering results through the same service interface.
7. Clients pay a small customisation fee and subscribe to services of their choice paying monthly fee. There is no capital outlay.
8. The development and implementation of software services is evolutionary. Starting with a single service, clients can add additional services if and when required.

8 FURTHER WORK

To achieve the greatest impact on lifecycle management improvement, we have selected to focus on the development and implementation of the Adaptive Spare Parts Storage and Scheduling Service, which provides opportunity for increasing despatch reliability of scheduled flights while reducing maintenance and repair time and costs. Real-time spare parts schedulers for aircraft maintenance and

repair differ from standard supply chain schedulers because they have to cope with the frequently changing aircraft routes – a factor which materially affects decisions on where to store spare parts.

REFERENCES

- [1] Knezevic, J., *Reliability, Maintainability and Supportability: A probabilistic Approach*, McGraw-Hill, 1993.
- [2] Knezevic, J., *Systems Maintainability: Analysis, Engineering and Management*, Springer, 1997.
- [3] Prigogine, I., *The End of Certainty: Time, Chaos and the New Laws of Nature*, Free Press, 1997.
- [4] Prigogine, I., *Is Future Given?*, World Scientific, 2003.
- [5] Kaufman, S., *At Home in the Universe: The Search for the Laws of Self-Organization and Complexity*, Oxford Press, 1995.
- [6] Holland, J.H., *Hidden Order: How Adaptation Builds Complexity*, Addison Wesley, 1995.
- [7] Holland, J., *Emergence: From Chaos to Order*, Oxford University Press, 1998, ISBN 0-19-850409-8.
- [8] Rzevski, G. & Skobelev, P., *Managing Complexity*, WIT Press, 2014.
- [9] Madsen, B., Skobelev, P., Rzevski, G. & Tsarev, A., Real-time multi-agent forecasting and replenishment solution for Lego branded retail outlets. *International Journal of Software Innovation*, **1(2)**, pp. 28–39, 2013, ISSN: 2166-7100.
- [10] Glaschenko, A., Ivaschenko, A., Rzevski, G. & Skobelev, P., Multi-agent real time scheduling system for taxi companies, *Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009)*, Decker, Sichman, Sierra and Castelfranchi (eds.), May, 10–15, 2009, Budapest, Hungary, pp. 29–35. ISBN: 978-0-9817381-6-1.
- [11] Andreev, S., Rzevski, G., Shveykin, P., Skobelev, P. & Yankov, I., *Multi-Agent Scheduler for Rent-A-Car Companies*, Lecture Notes in Computer Science, Vol. 5696, Holonic and Multi-Agent Systems for Manufacturing: Forth International Conference on Industrial Applications of Holonic and Multi-Agent Systems, HoloMAS 2009, Linz, Austria. Springer, ISBN 978-3-540-74478-8, pp. 305–314.
- [12] Himoff, J., Rzevski, G. & Skobelev, P., Magenta technology: multi-agent logistics i-Scheduler for road transportation. *International Conference on Autonomous Agents: Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multi-Agent Systems*, Hakodate, Japan, 2006, pp. 1541–1521. ISBN 1-59593-303-4. doi: <http://dx.doi.org/10.1145/1160633.1160927>
- [13] Granichin, O., Skobelev, P., Lada, A., Mayorov, I. & Tsarev, A., Cargo transportation models analysis using multi-agent adaptive real-time truck scheduling system. *Proceedings of the 5th International Conference on Agents and Artificial Intelligence (ICAART'2013)*, February 15–18, 2013, Barcelona, Spain. SciTePress: Portugal, 2013, Vol. 2, pp. 244–249. ISBN: 978-989-8565-39-6.
- [14] Tyrin, I., Vylegzhanin, A., Kolbova, E., Kuznetsov, O., Skobelev, P., Tsarev, A. & Shepilov, Y., Multi-agent system ‘smart factory’ for real-time workshop management: results of design & implementation for Izhevsk Axion-Holding Factory. *Proceedings of the 2012 IEEE 17th International Conference on Emerging Technologies & Factory Automation (ETFA 2012)*, September 17–22, 2012, Krakow, Poland, 4 pp. <http://www.etfa2012.org/>
- [15] Shpilevoy, V., Shishov, A., Skobelev, P., Kolbova, E., Kazanskaia, D., Shepilov, Ya. & Tsarev, A., Multi-agent system ‘Smart Factory’ for real-time workshop management in aircraft jet engines production, *Proceedings of the 11th IFAC Workshop on Intelligent Manufacturing Systems (IMS'13)*, May 22–24, 2013, São Paulo, Brazil, 2013.

- [16] Blinov, S., Serduk, V., Onischenko, G., Laruchin, V., Ochkov, D., Skobelev, P. & Tsarev, A., Multi-agent system for managing mobile teams for Samara Region gas distributor, *Proc. of International Conference on Complex Systems*, Samara, Russia, 2013, pp. 64–69.
- [17] Shabunin, A., Kuznetsov, N., Skobelev, P., Babanin, I., Stepanov, M. & Simonova, E., Multi-agent system for railways resource management. *Mechatronics, Automation, Control*. **1**, pp. 23–29, 2013.
- [18] Rzevski, G., Managing machine functionability using methods of complexity science, *2013 Annals of Mirce Mechanics*, Mirce Academy: Exeter, UK, January 2014, pp. 68–76.