Designing Distributed Multi-Agent System for Aggregate and Final Assembly of Complex Technical Objects on Ramp-up Stage

Petr Skobelev[2,3], Valery Eliseev[1], Igor Mayorov[1,3], Vitaly Travin[2],
Alexey Zhilyaev[1,2] and Elena Simonova[1,2]
[1]*Smart Solutions, Ltd, Samara, Russia*
[2]*Samara National Research University, Samara, Russia*
[3]*Samara State Technical University, Samara, Russia*

Keywords: Assembly Workshops, Planning and Scheduling, Distributed Problem Solving, Multi-Agent Technology, Multi-Agent System, Adaptive Scheduling, Knowledge Base, Decision Making, Real-Time.

Abstract: The paper covers the problem of aggregate and final assembly of complex technical objects at the ramp-up stage. New models, methods and tools for distributed scheduling are proposed, including modified version of virtual market with new classes of agents. The new feature of multi-agent scheduler considers knowledge base technology which helps to specify each operation in more detailed and individual way. The paper describes first system prototype for adaptive planning at the ramp-up stage and the main directions of future system development.

1 INTRODUCTION

The problem of production scheduling for workshops of aggregate and final assembly of complex technical objects is now often solved with the help of classical models, methods and means of production planning, such as SAP, BAAN, etc. (Cox, 2000, Herrmann, 2006).

However, during development of new versions of complex technical objects on ramp-up stage a number of special problems can arise (Klocke, 2016):

- composition of the product and production technology are finalized only during assembly;
- failures in the work of suppliers (non-compliance, non-delivery of parts or delay in delivery);
- workers do not yet have the required competencies;
- reassessment of labor productivity;
- unforeseen time spent on decision-making;
- urgent additional orders, etc.

As a result, significant delays occur in the plant's operation, which are especially evident in aggregate assembly workshops and at the final assembly line.

In this connection, this paper considers an approach to designing a distributed multi-agent system (MAS) for workshops of aggregate and final assembly lines. This system is capable of flexibly rearranging the work of assembly shops in case of unforeseen events.

The first part of this paper covers the problem of ramp-up production scheduling for complex technical objects assembly. The second part substantiates ways of solving the problem using the knowledge base and multi-agent technology. The third part provides a formalized formulation of the planning task, based on maximizing the overall satisfaction of agents in auction-like negotiations. The fourth part describes the basic method of adaptive agent planning and considers architecture of the distributed multi-agent scheduling system. The fifth part describes the architecture of the multi-agent system and its components. The sixth part demonstrates the possibility of using knowledge base for adaptive planning. The conclusions propose further steps for development of the system.

This project develops and advances the results of the ARUM project under the EU Program "Factories of the Future", during which the first prototype of this kind of system was created (Leitão, 2013).

Skobelev, P., Eliseev, V., Mayorov, I., Travin, V., Zhilyaev, A. and Simonova, E.
Designing Distributed Multi-Agent System for Aggregate and Final Assembly of Complex Technical Objects on Ramp-up Stage.
DOI: 10.5220/0006636202500257
In *Proceedings of the 10th International Conference on Agents and Artificial Intelligence (ICAART 2018) - Volume 1*, pages 250-257
ISBN: 978-989-758-275-2

2 DESCRIPTION OF THE PROBLEM AND THE PROPOSED APPROACH TO ITS SOLUTION

The problem of forming schedules of the aggregate and final assembly lines includes:

- formation of production schedules using various criteria, preferences and constraints;
- forecast of the possibility of performing schedules with available resources;
- adaptive re-scheduling of assembly in case of unforeseen events in real time;
- identification of bottlenecks and resource re-allocation between workshops;
- optimization of production schedule by comparing options and initial conditions;
- control of implementation of production schedules, etc.

Complexity of this processes is caused by NP-hard nature of combinatorial search, nonlinearity of decision making space, interdependences of operations in technological processes, specific individual features of matching rules for assigning operations to resources, including competencies of workers, and a number of other features.

Ramp-up stage is bringing additional complexity by high dynamics of production, when various events are constantly taking place: new orders are emerging, composition of products is changing, technological processes are being refined, supply terms are broken, work centers fail or defects are detected.

This complexity and high dynamics of scheduling process, which is event-driven in this case by definition, leads to the fact that traditional, centralized, hierarchically organized, sequential methods and algorithms of combinatorial type cannot effectively solve this problem with acceptable quality and within the required time for practical applications in the workshops.

For practical solution of the problem the paper proposes new distributed solution based on models, methods and algorithms for adaptive scheduling (Rzevski, 2014).

First of all, in this solution, instead of one "large" central scheduler for aggregate and final assembly shops, a distributed "system of systems" is proposed. It is built as a multi-level network of "small" stand-alone operational planners for individual workshops with a plan horizon of up to a month, working in coordination with the end-to-end scheduler of aggregate and final assembly workshops with a horizon of up to 3-6 months.

Secondly, to solve the problem of ramp-up management, it is suggested to use multi-agent technology which utilizes the concept of an "agent" – an autonomous software object capable of perceiving the situation, making decisions and interacting with others (Skobelev, 2014). Solution of any complex task in the multi-agent system is formed by self-organization of agents through interaction of dozens and thousands of agents of demand-and-resource network (DR-network), continuously competing and cooperating with each other (Vittikh, 2003).

The schedule of workshops is created as a self-organizing network of orders and resources, adaptively changing depending on events in real time. In the process of self-organization, software agents of orders and resources search for each other, at first choosing the best free options, and then resolving conflicts until the system is balanced and none of the options can improve the overall performance of the system. Then the calculation process stops, and the solution is given to the user.

This process more naturally describes the way experienced managers and dispatchers usually form schedules, finding a balance of interests among all concerned parties. The transition from combinatorial search for the optimal schedule to the search for an "acceptable" (reasonable) schedule corresponding to the current situation (reflecting the balance of interests of orders and resources at the given moment) allows to create quality schedules.

Thirdly, vitally important knowledge is need to be accumulated when such systems are implemented and used. However, this knowledge is often difficult to formalize. It can be used to improve the quality of planning, for example, information on the possibility of parallelizing part of technological operations or reinforcing them by adding more workers in order to accelerate the work, or knowledge about the competencies of workers, their compatibility in shifts, etc.

For this purpose, the paper proposes using the Knowledge Base (KB) for accumulation, formalization and use of these parts of knowledge, as storage of such knowledge is currently not provided by any of the corporate systems. Such a KB can be based on domain ontology as a semantic network of classes of concepts and relations – a technology that is actively developed within the Semantic Web to describe the content of Internet resources (Skobelev, 2012).

Application of KB will make the developed system open to users and allow them to customize the resource matching and scheduling rules within much wider limits than in traditional systems, to adapt the system during its use, to reduce labor intensity and the cost of operating the system.

3 FORMALIZED PROBLEM STATEMENT OF SCHEDULING BASED ON MULTI-AGENT TECHNOLOGY

In the formal problem statement of managing the production schedule, it is assumed that each of the orders and resources has its own criteria, preferences and limitations, and their importance can change during execution of tasks, and the purpose of the schedule is to find the maximum possible agreement between these agents, taking into account peculiarities of their current situation.

For this purpose, it is proposed to use multi-agent technology to automate the processes of identifying and resolving conflicts between agents of orders placed in the common pool of resources.

Within the proposed multi-agent technology, the problem of searching for a consensus can be formulated through the notion of "satisfaction" of agents of orders and resources, showing how much the observed criteria for selecting solution options differ from the ideal ones, given as wishes or based on experience of practitioners (Mayorov, 2015).

The following list of agents is proposed for building production schedules (Table 1).

Thus, the model of a manufacturing enterprise is specified through a network of linked demands and resources (DR-network), in particular, orders and resources (relating to execution of work on technological processes – operations and transitions): resources are represented by assembly stations, work centers, and workers of specific areas of expertise. Orders at this level are production assignments to workers for assembly of technological assembly units (TAU) and assembly units (AU). Further, larger structural subdivisions are considered – sections, production shops with sections and orders at the level of aggregate assembly shops with corresponding inter-shop links. At the top level, there is final assembly in the final assembly shop with cooperation in adjacent production and supply.

Table 1: The main classes of MAS agents and their functions.

Agent class	Agent functions
Order agent	Selects the technological process (from alternative ones), monitors the plan and the fact
Technological process agent	Plans tasks, monitors timeframes and results obtained within the process
Division agent	Manages the load of employees within divisions (shops, stations, workers)
Task agent	Searches for the best worker according to the list of competencies and conforms the terms of task fulfillment, monitors its execution
Resource agent	Plans tasks suitable for the employee or equipment, monitors the load and results
Product agent	Evaluates the results, does not allow for idling
Enterprise agent	Finds and corrects bottlenecks in plans and analyzes risks

In the proposed approach, each department of assembly production at the level *h* (assembly line, workshop, enterprise a s a whole) is given specific types of agents of resources, tasks, operations, technical processes, products (TAU and AU), the state of which is described through functions of satisfaction $u^{res\ h}{}_{j}$ by indicators (criteria functions) *i* from the set $\{x_i^h\}$ with the weight $\alpha^{res\ h}{}_{ij}$, characterizing how much their values for the given indicators deviate from the desired values of $x_{ij}{}^{id\ h}$ for resource *j* in subdivision *h*. The indicators are brought together in an additive way into a unified satisfaction function. In this model, the target function of resource agents *(res)* in subdivision at level *h* of enterprise structure on satisfaction from range [0,1] is piecewise linearly dependent on indicators x_i^h, on their values at the previous level of organization *h-1* and the values of satisfaction of resource agents. Similarly, functions with the weight $\beta^{task\ h}{}_{mn}$ of satisfaction of tasks $u^{task\ h}{}_{n}$ at the level *h* can be given, where the set $\{y_n^h\}$ and $\{z_p{}^h\}$ is considered as indicators for agents of products u^{prod}. For each of the agents of resources, tasks, products, and enterprise agent, importance (priorities) $\{w_j{}^{res\ h}\}$, $\{w_n{}^{task\ h}\}$, $\{w_p{}^{prod\ h}\}$, respectively, can be introduced. Superposition of satisfaction functions is related to functions of bonus-penalties.

The enterprise agent is considered the only one at each level *h* of enterprise structure. The task of building an enterprise plan is to maximize satisfaction of agents for level of enterprise structure *h=1,..., H*, including agents of resources, tasks and products (1):

$$u^{res\,h} = \sum_j w_j^{res\,h} u_j^{res\,h} = \sum_j w_j^{res\,h} \sum_i \alpha_{ij}^{res\,h} f_{ij}^{res\,h}\left(x_i^h, x_{ij}^{id}, x_i^{h-1}, f_{ij}^{res\,h-1}\right)$$

$$u^{task\,h} = \sum_n w_n^{task\,h} u_n^{task\,h} = \sum_{\text{T}} w_n^{task\,h} \sum_m \beta_{mn}^{task\,h} f_{mn}^{task\,h}\left(y_m^h, y_{mn}^{id}, y_n^{h-1}, f_{mn}^{task\,h-1}\right)$$

$$u^{prod\,h} = \sum_p w_p^{prod\,h} u_p^{prod\,h} = \sum_p w_p^{prod\,h} \sum_k \gamma_{kp}^{prod\,h} f_{kp}^{prod\,h}\left(z_k^h, z_{kp}^{id}, z_k^{h-1}, f_{kp}^{prod\,h-1}\right)$$

$$u^{dep\,h} = \sum_l \delta_l^{dep\,h} f_l^{dep\,h}\left(s_l^h, s_l^{id}, s_l^{h-1}, f_l^{dep\,h-1}\right)$$

$$x^{res\,h^*} = \max_{x_i^h}(u^{res\,h}) \qquad y^{task\,h^*} = \max_{y_m^h}(u^{task\,h})$$

$$z^{prod\,h^*} = \max_{z_k^h}(u^{prod\,h}) \qquad s^{dep\,h^*} = \max_{s_l^h}(u^{dep\,h})$$

where $x^{res\ h*}$, $y^{task\ h*}$, $z^{prod\ h*}$, $s^{dep\ h*}$ are the "optimal" (for this situation) values of variable resources, tasks, products and enterprise for level h. For the lower level of enterprise structure the functions of satisfaction components $f_{ij}{}^{h=1}$ do not depend on the previous values,

$x_i \in D^I$, $y_m \in D^M, z_k \in D^K, s_l \in D^L\ \ \forall i, m, k, l.$ The variables x, y, z, s lie in the range of the following indicators: D^I resources, D^M work on technological processes, D^K products, and D^L enterprise agent, where I, M, K, L are dimensions of the corresponding spaces.

Evaluation of their states is performed by agents using the functions of satisfaction and bonus-penalty multi-component functions.

Evaluation of the quality of planning in operational schedulers is carried out using the main criteria:

- minimization of production end time and compliance with delivery dates of each object;
- maximization of the number of produced object for the given planning horizon;
- minimization of time spent on production of each TAU and AU;
- minimization of delays and storage time for assembled aggregates, TAU and AU between operations;
- maximum concurrency of TAU production, allowed by technical processes;
- ensuring uniformity of equipment load in each workshop and reducing downtime;
- uniformity of work load of employees and minimization of additional shifts;
- decreasing the amount of work in progress at the end of the period.

Thus, the planning task is formulated for agents as the task of maximizing total satisfaction (1), provided that decisions taken by the agents are consistent. The recursiveness of the task by department level and the non-linearity with respect to the solutions at the previous level allow for an iterative solution with the help of "nested" network multi-agent schedulers of aggregate and final assembly shops.

4 THE BASIC METHOD OF ADAPTIVE PLANNING

At the heart of the developed multi-agent planning technology lies the concept of "virtual market" of a DR-network of any department, production line or shop, where order agents can buy resource services at a virtual price.

Within the virtual market of the system, orders receive a certain stock of virtual currency for purchases of finished products (components) or resource services, and resources sell their services on a time-based or other basis. Order agents can enter into virtual contracts with resource agents, but then, if necessary, they can reconsider their decisions, compensating each other for possible losses from the break of contacts. In the long term, virtual money can be directly linked to real money and micro-contracts during order execution to reflect the state of production in real time.

The basic method of adaptive planning based on the example of initial planning of TAU in the operational planner of the workshop is as follows:

1. The order agent for production of a new TAU loads its execution technology from the KB and creates an agent of technological process (tasks).

2. The technological process agent creates agents of child tasks (operations within the technological process), setting tasks for them.

3. Agents of tasks begin searching for free (or busy, if they are not available) resources in the scene of the shop, which will cost the least amount of virtual money, providing the maximum profit due to the bonus part of the penalty function.

4. If the resource is already occupied by other tasks, the conflict is fixed and its resolution begins.

5. In case of a conflict, the task already placed on the resource is asked to give way, but if the new placement is worse, then the required compensation is calculated according to the function of bonuses and fines.

6. If the required compensation is too high, the incoming task itself starts to look for a new placement. And if the compensation is acceptable, it pays for the departure of the previous task.

7. This process continues until there is a dynamic stop, when no agent can improve its condition (satisfaction function).

8. At this point, the Enterprise Agent identifies the agent with the worst criteria for the system as a whole.

9. The selected worst agent is given a command to break ties and be dis-scheduled from related orders / resources.

10. The selected agent receives an increased importance coefficient for the worst criterion in its satisfaction function, which will cause selection of other options when it is rescheduled.

11. The agent tries to be rescheduled – if successfully, the process proceeds to the next criterion and the corresponding agent.

12. If not successfully, the agent reports the sum of virtual currency, which he needs in order to reach a new criterion value.

13. The enterprise agent assesses availability of this sum in the system and, if necessary, adds the virtual currency to the agent.

14. As a result, the worst agents iteratively "pull up" their criteria to new values, compensating for the losses of other agents at the expense of the virtual budget.

15. The process ends when all the criteria of the system as a whole fall into the comfort zone or when they can no longer change.

16. Otherwise, control is transferred to the user for further manual improvement.

Improvement of the current state of agents is performed through proactive interaction between order agents, task agents and resource agents within the proactivity protocol (Figure 1).

Division agent, having received a message from the order agent, ranks the order list according to the delay fine in the descending order. Then, during evaluation of the schedule, the division agent updates all of its indicators, the total profit, and normalized criteria values. The list is dynamically rebuilt as a result of the algorithm performance.

The tasks search for a more profitable position in the resource schedule, while each resource plans more profitable tasks in the oncoming process.

Changes in the positions of task agents are made on the basis of the compensation method. In this method, permutations occur in those cases where compensation to the displaced agent from the shifting one exceeds the decrease of its objective function.

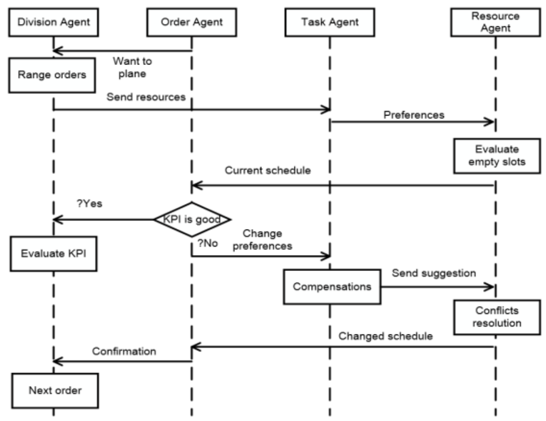

Figure 1: The part of simplified protocol proactive improvement of schedule.

The conditions for completion of the event scheduling algorithm are as follows:

- the stop occurs when the system reaches the balance of satisfaction between conflicting agents (orders and resources, as well as others), when none of the proposals can increase the value of indicators;
- changes in agents' indicators and agent satisfaction associated with them does not exceed the specified threshold (set in the settings);
- termination of negotiations between conflicting agents in case of refusal by the parties to further negotiate due to lack of options or achievement of the state of full "nirvana" (specified special zone of satisfaction);
- the allotted number of iterations or the specified time for proactive improvement has been exceeded;
- failure to reach the specified by the user indicators, with a message to the user about the need for manual intervention to resolve conflicts.

As a result, the created schedule is built in the process of self-organization of shop agents, which allows for flexible and quick adaptation of the schedule depending on events.

5 ARCHITECTURE OF THE BASIC MULTI-AGENT SYSTEM

The architecture of the multi-agent scheduler is presented in Figure 2.

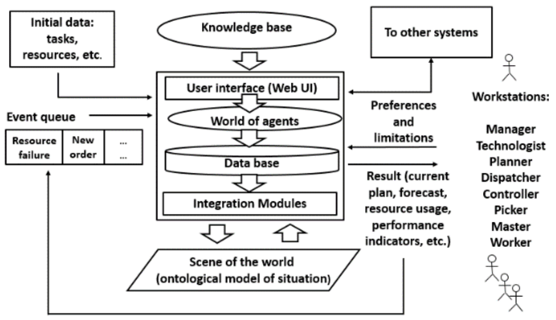

Figure 2: The architecture of the multi-agent scheduler.

The following most important components are found in the architecture of the basic multi-agent scheduler:

- World of agents consists of instances of agent classes to be executed. Agents are asynchronous software objects (programs) operating as a state machines.
- Event queue provides accumulation of events coming from the real world, as well as their processing.
- The World Scene is a semantic model of the current situation, built by the agents of the system on the basis of ontology.
- Application components make it possible to perform additional functions for third-party systems and user interfaces.

The distributed multi-agent system contains two-level "system of systems". This system is based on the network-centric interaction of planning modules (schedulers) used at the inter-shop (strategic) and workshop (operational) levels (Figure 3).

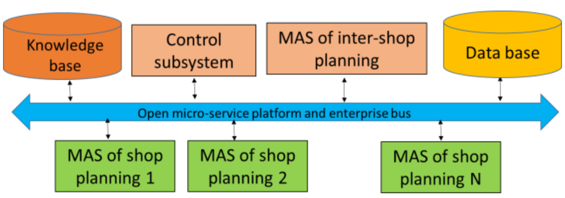

Figure 3: Network-centric system architecture.

The module of end-to-end inter-shop planning forms the work schedule for a group of aggregate and final assembly shops with a time horizon of one quarter. Information on the capacity of equipment and the total number of employees of a certain category is taken into account as limitations for orders execution.

At the shop level, there are operational planners, each of which is responsible for scheduling a certain workshop. The operational planning module has more detailed information about the resources of its workshop, including peculiarities of technological processes, the composition and characteristics of equipment and personnel. The technological processes considered at this level can have a greater degree of granularity and additional requirements for characteristics of equipment and employees.

6 USING THE KNOWLEDGE BASE FOR ADAPTIVE PLANNING

The main purpose of the Knowledge Base (KB) is formalization of the problem domain in order to provide all the necessary information to systems that solve applied problems, in particular, planning and managing resources, by aggregating knowledge from various information sources and providing the user with the most complete information on the object and the current situation in production.

The purposes of using KB are the following:

- customization of MAS scheduler for the specific features of each workshop;
- extension of matching rules for assigning tasks to resources (add criteria "on the fly");
- ensuring openness of the system and reducing the costs of its operation.

The formalized model of knowledge in KB is represented as classes of concepts and relations that form semantic network of the domain. The knowledge base technology is used in this case in order to more accurately specify the requirements that should be taken into account for individual planning of each operation within the technological process and to separate this knowledge from the source code of the system in order to enable users to independently expand or modify this information.

For ease of use, three levels of domain description are distinguished: "ontology", "ontological model", and "situation" ("scene").

The key element within KB is ontology, which at the basic level should, first of all, provide a vocabulary of concepts for representation and exchange of knowledge, as well as the multiple relationships (relations) established between the concepts in this vocabulary. Attributes are introduced for concepts and relationships, in order to expand the possibility of describing subject areas by including in ontology not only objects, but also their qualitative and quantitative characteristics.

The ontological model describes real workshop as a set of equipment, workers, technological processes and operations, etc.

The scene describes instances of concepts and relations with specific attribute values at the given point in time (a set of facts).

The ontology consists of the following sections:

- organizational structure of the enterprise, describing the list of departments, employees, their professions, positions, competencies;
- infrastructure model of the enterprise, describing the main technical means (equipment, tools, materials, production sites, facilities, characteristics of material objects, physical and technical principles of operation, etc.);
- technological processes of assembly, describing the processes of the main activities (production technology);
- manufactured products and provided services, key indicators of quality and efficiency of service delivery;
- professional specialization and competence of workers;
- other knowledge required to plan operations of technological processes.

The main part of the knowledge base for planning is the concept "Task" and its instances for performing specific technological operations (Figure 4).

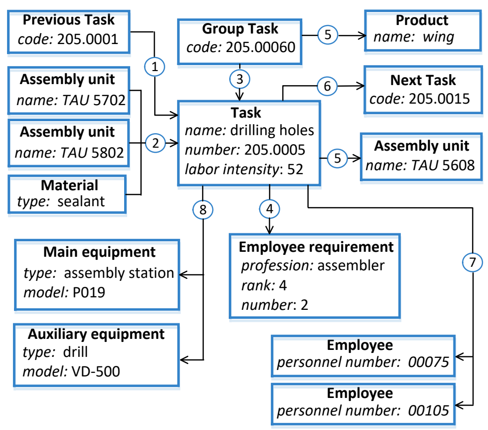

Figure 4: Relations in the concept "Task".

The presented relations in the concept "Task" are used by agents in resource management system in the following way:

- "Previous" (1) and "Next" (6) – allow the agent to find the previous task with the request to move earlier or reschedule, to find the next task, and to send its agent a delay message;
- "Input objects" (2) and "Output objects" (5) – show which agents should be in the scene to start the task execution, and also what will be the output as the result of task execution;
- "Part-Whole" (3) – shows that the task is a part of a compound problem, the agent of which receives parameters of the plan and the fact of the task;
- "Employee Requirements " (4) – defines the requirements for the performer of the task;
- "Who did it last time?" (7) – finds employees who have already performed such tasks;
- "Required resource" (8) – specifies a resource, predetermined by the manufacturing process.

The developed approach to MAS prototype makes it possible to add new rules of matching "on the fly", taking into account the complexity factors revealed during planning for the workshops.

7 CONCLUSIONS

Further developments will be aimed at implementation of schedulers of workshops and their coordinated interaction in distributed system of system for factory scheduling.

The transition from one centralized "global" scheduler to a distributed MAS solution based on a network of MAS schedulers of workshops will ensure further expansion of the system, better efficiency, flexibility and performance, productivity, scalability and reliability of the system.

Further research will be focused on measuring adaptability, quality and efficiency of scheduling.

ACKNOWLEDGEMENTS

This paper was prepared with the financial support of the Ministry of Education and Science of the Russian Federation – contract №14.578.21.0230, project unique ID is RFMEFI57817X0230.

REFERENCES

Cox, J., Spencer, M., 2000. *The Constraints Management Handbook*. SRC PRESS. London.

Herrmann, J. (ed.), 2006. *Handbook of Production Scheduling*. Springer. Switzerland.

Klocke, F., Stauder, J., Mattfeld, P., 2016. Modeling of Manufacturing Technologies During Ramp-up. *Procedia CIRP*, vol. 51, pp. 122-127. Elsevier.

Leitão, P., Barbosa, J., Vrba, P., Skobelev, P., Tsarev, A., Kazanskaia, D., 2013. Multi-agent System Approach for the Strategic Planning in Ramp-up Production of Small Lots. In *IEEE SMC 2013, the IEEE International Conference on Systems, Man and Cybernetics.* IEEE Computer Society, Washington, DC, USA, pp. 4743-4748.

Rzevski, G., Skobelev P., 2014. *Managing complexity*. WIT Press. Boston.

Skobelev, P., 2014. Multi-Agent Systems for Real Time Adaptive Resource Management. In *Industrial Agents: Emerging Applications of Software Agents in Industry*. Elsevier.

Vittikh, V., Skobelev, P., 2003. Multiagent Interaction Models for Constructing the Demand-Resource Networks in Open Systems. *Automation and Remote Control*, vol. 64, issue 1, pp. 162-169.

Skobelev, P., 2012. Activity ontology for situational management of enterprises in real-time. *Ontology of Designing*, vol. 1, issue 3, pp.6-38.

Mayorov, I., Skobelev, P., 2015. Towards thermodynamics of real timescheduling. *International Journal of Design & Nature and Ecodynamics*, vol. 10, issue 3, pp. 213-223. WIT Press.