ARCHIVES of FOUNDRY ENGINEERING

Published quarterly as the organ of the Foundry Commission of the Polish Academy of Sciences

ISSN (1897-3310)
Volume 15
Special Issue
2/2015

17 – 22

4/2

Multi-agent Approach to Production Scheduling in Foundries

J. Duda *, A. Stawowy
AGH University of Science and Technology, Faculty of Management, Gramatyka 10, 30-067 Krakow, Poland
*Corresponding author. E-mail address: jduda@zarz.agh.edu.pl

Received 06.05.2015; accepted in revised form 29.05.2015

Abstract

The paper describes a complete framework that uses a multi-agent approach for production scheduling in a foundry. Different types of autonomous agents have been designed, each playing a different role in the supply chain management of the fodundry, along with their responsibility and behavior in the system. In order to generate a proper and reliable schedule the agents negotiate with each other by sending messages compliant to FIPA ACL standard. A prototype of the system has been implemented in JADE and allowed to conduct preliminary simulation of the system. Although some agents some agents have been omitted in the prototype, it was possible to verify the main assumptions of the system, as well as, to indicate and discuss potential problems with its implementation into production practice.

Keywords: Application of information technology to the foundry industry, Production planning, Scheduling, Multi-agent approach

1. Introduction

Production scheduling in foundries is still an open question. Classical approach to mid-term production planning, performed on the basis of balancing the resources, is well explored and described in the literature. In contrast to this, short-term planning (scheduling), due to interconnections between different management levels and interdependence of the decisions made at the various time horizon, is very difficult optimization problem, and new solving ideas and methods still are developed. Scheduling is carried out on a daily or weekly basis to determine the assignment and sequencing of tasks (clients' orders) to production units.

There are various decision-making tools employed in the management and production logistics. Kobbacy et al. [8] divide them into two major classes: Operational Research (OR) and Artificial Intelligence (AI). OR techniques include mathematical programming, network analysis, regression, queuing theory, simulation, and maintenance models. The main AI techniques are: logic and theorem proving, uncertainty management, case-based reasoning, data mining and symbolic learning, neural networks, heuristic searching methods, and intelligent agents.

Complex scheduling models are no longer based solely on traditional mathematical programming, but also on constrained programming, business rules and other concepts require more sophisticated methods to solve them. It is necessary to divide planning and scheduling problems into subproblems, simultaneously ensuring that all relations are preserved. Two approaches are typically used: conventional hierarchical approach, and, more recently, agent based approach.

Multi-agent systems are a distributed intelligent approach that is very well suited for modular, distributed, week structured, and complex environment. Multi-agent systems endorse integration, heterogeneity, co-operation and co-ordination, robustness and reactivity, flexibility, and adaptability to rapid changes [11].

In agent based systems the solution of planning and scheduling problem is divided into set of agents which cooperates in order to provide final solution. Particular agents are responsible e.g. for demand forecasting, plan generation, detailed daily

schedule generation and for controlling that all constraints and business rules are satisfied. Application of heterogeneous agents in such systems enables any planning and scheduling subproblem to have different solution technique [10]. It is also worth to underline that agent based solvers are parallel by their nature and they can be also easily scaled, so such approach usually does require any additional computer power to be bought by the enterprises. They can use their own computer resources or the computational power of remote computers by applying the idea of distributed computing or even cloud computing.

The aim of this paper is to present multi-agent system (MAS) for production scheduling that can be applied to foundries production management. In the field of MAS application to planning and scheduling in foundries, the literature is not extensive. We could find only a few papers dealing with this problem that are discussed in Section 2. The details of proposed approach are given in Section 3. The computational experiments are described in Section 4. The conclusions are drawn in Section 5.

2. Related research

In the field of multi-agent system application for production management, the literature is vast. Comprehensive reviews of agent-based systems in the manufacturing can be found in Pechoucek & Marik [12] and Andreadis et al [1].

He and Babayan [7] have introduced the general framework of the agent-based approach for manufacturing system scheduling in an agile manufacturing environment. Each agent is assigned a separate individual job to be scheduled in the system according to the precedence relationships of other agents' jobs. If any changes are made by users in component designs, or assembly structures the product designs, rescheduling can be done easily by corresponding agents.

Dangelmaier et al. [3] have presented the prototype of the multi-agent system for cooperative production planning (MASCOPP) which consists of an editor and runtime environment to model and initialize the agents. The goal is a synchronization of decentralized production plans of autonomous entities.

Lima et al. [10] have presented Production Planning and Control system that can be dynamically adaptable to local and distributed utilization of production resources and materials. The multi-agent system is based on three main agents: Client, Resource, and Manager. These agents negotiate the final product, and the correspondent components, requested by the client.

More recently, Li et al. [9] have proposed an agent-based approach to facilitate the integration and optimization of planning and scheduling processes. The framework consist of three agents and several databases. The job agents and machine agents are used to represent jobs and machines. The optimization agent is used to optimize the alternative process plans and scheduling plans. With the consideration of the scheduling requirements and availability of manufacturing resources, these agents negotiate with each other to establish the actual process plan of every job and the scheduling plans for all jobs. The experiments show that the proposed approach is very effective for the integrated planning and scheduling problems.

In spite of the large number of theoretical works reported on production planning and scheduling, there are very little industrial applications of MAS in this field. One of the examples is multi-agent system to the scheduling problem in a ceramic tile factory proposed by Giret et al. [6] in which the constituent agents cooperate to find a feasible schedule taking into account on-line orders, factory layout and capacity, time constraints, anticipated demands and constraints imposed by the master plan.

Cowling et al. [2] have reported the use of multi-agents system for integrated dynamic scheduling of steel milling and casting. Each agent contains its own scheduling model and realizes its local predictive-reactive schedule taking into account local objectives, real-time information and information obtained from other agents. Agents collaborate to find a globally good (not necessary optimal) schedule, which is able to effectively react to unexpected technical and organizational events occurring in real-time environment.

Ouelhadj et al. [11] have presented an inter-agent cooperation protocol for integrated optimization and dynamic scheduling of the continuous caster and the hot strip mill. Local autonomy allows the planning agents to generate local departments' schedules, using tabu-search heuristic, and to respond locally to disruptive events using the proper rescheduling method. The cooperation protocol allows the panning agents to cooperate and coordinate their local schedules in order to generate global schedules. The cooperation protocol is a three-phase procedure, which involves announcing, bidding, and contracting. To deal with the real-time events, a decommitment mechanism is included in the negotiation protocol that allows the planning agents to propose an alternative schedule to the previously established contract. The experimental tests on predictive and reactive scheduling have showed that local autonomy and cooperation capabilities of multi-agent systems lead to a significant increase in schedule quality and system robustness.

More recently, Fazel et al. [15] have proposed a much more complex solution for the cooperative planning and scheduling in the steel industry. Their system is based on multi-agent architecture and adaptive neuro-fuzzy networks. The system consists of six agents. Agents are built around several modules: user interface, communication interface, reasoning module that uses knowledge base and learning module. Customer agent is responsible transforming customers' orders into production orders after negotiations with a user agent. The User agent deals with customers' orders and communicates with other agents to process them. The role of the Ingot Casting (IC) agent is to supply ingots produced in ingot moulds to the User agent. The Vacuum Degassing (VD) agent is responsible for determining parameters of degassing process. The Ladle Furnace (LF) agent is responsible for determining parameters of molten steel refining process according to the customer's order. Finally, the Electric Arc Furnace (EAF) agent supplies molten steel to the LF agent.

The User agent has to assess the order send from the Customer agent wheatear it is feasible taking into account cost of necessary additives and cost of delivery date differences. If the order receives infeasible status the User and the Customer agents may negotiate different delivery date and/or required properties. The IC agent determines the amount of molten steel that is necessary to produce the ingots ordered by the customer and provides the estimation of total casting processing time for the

order. Both VD and LF agents have to deal with uncertain parameters of the production process. Vacuum Degassing agent must determine such parameters as vacuum pressure, the amount of neutral gas and the processing time. For this purpose, an adaptive neuro-fuzzy inference system is used.

We found only two papers dealing with multi-agent system dedicated to foundry industry.

Duda and Stawowy have proposed a hierarchical multi-agents system that uses PSLX/PPS ontology and communication protocols [4]. There are many technological constraints in foundry processes, which the planner must deal with, but usually only few of them are crucial for effective planning. The operations for remaining processes must be scheduled regarding the superior "interest" of the schedule created for those most important processes, which are usually bottlenecks of the system and/or major cost generators. The schema of the proposed system architecture is presented in Figure 1.

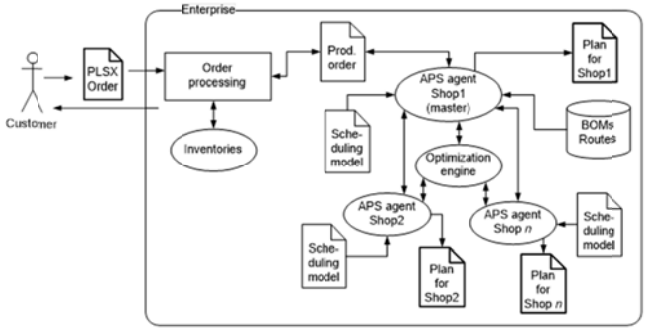

Fig. 1. A structure diagram of the multi-agent system

The key element in the system is an optimization engine, which can be used by any of the APS agents (including the ones outside the enterprise). The agent has to contain its own optimization model (problem description).

Wilk-Kołodziejczyk et al. [14] have presented a practical solution in the form of implementation of agent-based platform for the management of contracts in a network of foundries. The proposed system allows for the joint management of the orders in the network of small and medium-sized metallurgical, providing them with greater competitiveness and the ability to deliver large orders. The paper shows the operations of individual agents representing companies looking for potential suppliers or recipients of services and products. An important element of the system is bi-directional agent used for translation standards based on ontology, whose task is to automate the decision-making process in preparing tender documents.

3. Multi-agent scheduling system

In our example we propose a mutli-agent system that contains several types of autonomous and heterogeneous agents. Each agent plays a different role in a supply chain management, i.e. collecting customers' orders, negotiating due-dates, managing inventory data, taking into account cooperators' restrictions, monitoring the production process and finally providing a proper and reliable schedule. There is also a dedicated agent responsible only for performing optimization tasks.

3.1. System architecture

A structure diagram of the entire system is shown in Figure 2.

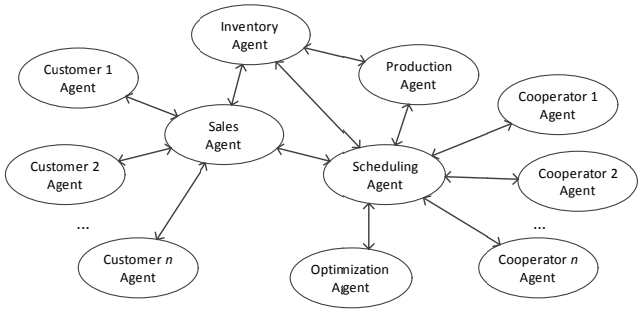

Fig. 2. A structure diagram of the multi-agent system

- Sales Agent is responsible for receiving customers' orders, sending them to the Scheduling Agent and negotiating due dates with the Customer Agents.
- Scheduling Agent plays a key role in the system and is responsible for production scheduling and rescheduling over a given planning horizon. It communicates with the Inventory Agent to check what castings have been already manufactured. It also communicates with the Cooperator Agents to gain the constraints that must be included in the scheduling problem. It may also communicate with the Sales Agent if something goes wrong (e.g. a machine breakdown occurs) to negotiate postponing of the due dates. Finally, every time a schedule must be updated it sends a request to the Optimization Agent to compute the scheduling problem.
- Optimization Agent in our system plays currently a role of optimization engine and uses genetic algorithm to produce within few minutes a feasible schedule, which is not necessarily optimal, but usually 1-5% distant from the optimal one.
- The role of a Production Agent is to update the inventory data with the manufactured quantities and to report every failure or disruption in the production process that may affect the production schedule, and possibly requires its updating.
- Inventory Agent is responsible for storing the data on the inventory levels of the already manufactured castings and passing the inventory levels on the request by the Sales Agent and the Scheduling Agent.
- Cooperator Agents are used to negotiate the feasibility of the schedule and provide the constraints to the production problem that must be fulfilled.

In theory all the agents may be implemented as modules in a production planning and scheduling system and all the communication between them may be carried out almost automatically. This, however, would require that all customers and cooperators have a computer system that is able to send and

receive messages from the proposed system (possibly compatible with FIPA ACL [5] specification) and to interpret them in a proper way. This in turn would require a common ontology that ensures that concepts like production order, produced item, manufacturing operation, machine capacity etc. are interpreted in the same way. In [4] we described a PSLX language that includes such ontology that might be used in this case.

In a real production practice implementation of the necessary agents in all supply chain participants may be very costly and therefore uneconomic, especially that fully automatic negotiations are impossible in many cases, so the human factor cannot be eliminated anyway. So in the following discussion we will focus on the agents that can be implemented within the foundry in a separate computer system. The communication between the Sales Agent and the Customers Agent will be done in the conventional manner, i.e. by phone and e-mail. The same will concern the Production Agent, which in our case will be the operator of the system registering production transactions (ERP). The task of the operator will be to update the inventory levels after the production of some batch of castings has been finished and to send the message to the Scheduling Agent if there is a problem with the production process together with the expected date when this problem will be fixed and its impact on the production capacity (e.g. one of the furnaces will not operate for 2 consecutive hours).

3.2. Negotiation protocols

The agents communicate with each other by sending and receiving messages written in the FIPA ACL format that is supported by the majority of multi-agent frameworks. Although more 20 different types of messages were define in the ACL standard our system uses only five types: REQUEST, PROPOSE, AGREE, REFUSE and INFORM.

We will show the application of the particular types of messages on the example of negotiating a due-date for a customer's order that has been sent by the Customer Agent.

After receiving a new order (to simplify the negotiation process we assume that the technology for the ordered castings is already known) the Sales Agent sends a message to the Scheduling Agent in order to designate a due date for this order in the following form:

Sender: Sales agent
Receiver: Scheduling agent
Message Type: REQUEST
Message Action: Designate a due-date for an order
Deadline: Time for scheduling agent to respond
Content: Order parameters

The Scheduling Agent responds with PROPOSE message in which the earliest possible delivery date is proposed:

Sender: Scheduling agent
Receiver: Sales agent
Message Type: PROPOSE
Message Action: Confirm due-date
Deadline: Time for customer to respond
Content: Proposed due-date for a customer order

The Sales Agent after consulting the Customer agent may send either an AGREE or a REFUSE message. In the latter case it may send another REQUEST message to the Scheduling Agent requiring the confirmation of a due-date given by the customer.

Sender: Sales agent
Receiver: Scheduling agent
Message Type: REQUEST
Message Action: Confirm a due-date for an order
Deadline: Time for scheduling agent to respond
Content: Order parameters and requested due date

In this case the Scheduling Agent may respond with either an AGREE or in the most cases with an INFORM message, informing the Scheduling Agent which orders have to be postponed in order to produce the ordered castings on the requested date.

Sender: Scheduling agent
Receiver: Sales agent
Message Type: INFORM
Message Action: Postponed orders
Deadline: Time for sales agent to respond
Content: List of orders to be postponed

The sales agent may respond with either an AGREE message or a REFUSE message and usually will issue another REQUEST message with the proposal of another due date.

4. Computational experiments

A prototype of the system has been implemented in JAVA Agent DEvelopment (JADE) framework. Until now we have defined 4 agents: *InventoryAgent*, *SalesAgent*, *SchedulingAgent* and *OptimizationAgent*. The agents enumerated in JADE Remote Agent Management GUI are shown in Figure 3.

Fig. 3. Agents in the scheduling system.

InventoryAgent maintains an array that stores the levels of castings and returns those levels on the *SchedulingAgent* requests. The inventory level is updated by the *SchedulingAgent*, which

simply sends the data about the production lots scheduled for a given period and update this data if it performs a rescheduling operation. As we mentioned earlier, in real scheduling system there would be a *ProdcutionAgent* monitoring actual production and update the inventory data on the basis of the amount of actually manufactured castings.

SalesAgent sends request to the *SchedulingAgent* in two forms: a) asking for a projected due-date for a new order, b) enforcing particular due-date to be preserved. Each day the *SalesAgent* should send a request to the *InventoryAgent* to decrease amount of castings with a due-date equal to the current date (this function has not been yet implemented).

We have shown in the previous section that, dependent on the *SalesAgent* request, the *SchedulingAgent* may respond with: a) due-date or b) due-date with the orders for those due-dates have to be changed. In order to calculate a due-date for an order the *SchedulingAgent* first sends a request to the *InvenotryAgent* to gain the actual amount of castings stored in finished product warehouse, then sends a request to the *OptimizationAgent* to solve optimization problem using adequate model. After receiving confirmation from the *SalesAgent*, the *SchedulingAgent* sends information of scheduled castings to the *InvenotryAgent*.

OptimizationAgent uses the genetic algorithm described in [13], as it can provide relatively good results within 3 minutes, which for our purpose is satisfactory. For the general request about a possible due-date for a new order, the model presented in [13] (denoted here as *model1*) is used. This model seeks for such schedule that minimizes the sum of the costs of delayed production, storage costs of finished goods and the setup cost, if the alloy is changed during furnace loads (1).

$$\text{Minimize} \quad \sum_{i=1}^{I}\sum_{t=1}^{T}(h_{it}^{-}I_{it}^{-} + h_{it}^{+}I_{it}^{+}) + \sum_{k=1}^{K}\sum_{n=1}^{N}(st_k z_n^k) \tag{1}$$

where: I_{it}^{-}, I_{it}^{+} - number of castings i delayed (–) and stored (+) at the end of day t; $z_n^k = 1$, if there is a setup (resulting from a change) of alloy k in sub-period n, otherwise 0; st_k - setup cost for alloy k; C - loading capacity of the furnace; h_{it}^{-}, h_{it}^{+} - penalty for delaying (–) and storing (+) production of item i in day t.

However, for the request from the *SalesAgent* demanding a particular due-date to be preserved, a slightly different model is used (denoted as *model2*). This time we seek rather for a good schedule that simultaneously minimizes the number of due-dates that have to be postponed in order to maintain a particular due-date for a given order. The objective function looks as follows:

$$\text{Minimize} \quad \sum_{i=1}^{I}\sum_{t=1}^{T}(h_{it}^{-}I_{it}^{-} + h_{it}^{+}I_{it}^{+}) + P_d\sum_{i=1}^{I}\iota_i^{-} \tag{2}$$

where: P_d is a penalty for each single due-date that has to be postponed, $\iota_i^{-} = 1$, if the original due-date for order i has to be postponed, otherwise 0.

We have also introduced an additional constrains ensuring that the orders with high value of penalty for delaying h_{it}^{-} (with the highest priority) cannot be postponed.

$$\iota_i^{-}h_{it}^{-} \le H, \quad i = 1,...,I, t = 1,...,T \tag{3}$$

where H is a boundary value for penalties of delaying.

An exemplary negotiating session between the *SalesAgent* and the *SchedulingAgent* is shown in Figure 4.

```
SchedulingAgent: REQUEST received from SalesAgent
Action is "Designate due-date for order #10"
Sending REQUEST to Optimization Agent. Model is "model1"
==> Answer is "Proposed due-date: 8"

SchedulingAgent: REQUEST received from SalesAgent
Action is "Refuse"
==> Answer is "Order has not been scheduled!"

SchedulingAgent: REQUEST received from SalesAgent
Action is "Confirm due-date 5 for order #10"
Sending REQUEST to Optimization Agent. Model is "model2"
==> Answer is "Possible due-date: 4. Orders to be changed: 5, 18"

SchedulingAgent: REQUEST received from SalesAgent
Action is "Agree"
==> Answer is "Order #10 has been scheduled for day 4"

InventoryAgent: INFORM received from SalesAgent
Action is "Inventory change"
==> Answer is "Inventory has been updated"
```

Fig. 4. Negotiating session between SalesAgent and ScheduligAgent

In our simulation the *SalesAgent* used a uniform distribution random number to accept or reject of the due-date achieved by solving *model1* (the probability of acceptance was set to 0.7). If the due-date was not accepted *the SalesAgent* sent another request with a proposed firm due date, also randomly generated, but before the date achieved by *model1*. In that case, the *SchedulingAgent* ordered the *OptimizationAgent* to use *model2* to determine the number of orders that due dates have to be postponed. The *SalesAgents* accepted it with a probability of 90%. After the schedule has been accepted the *SchedulingAgent* sent an information to the *InventoryAgent* to update its inventory array.

In our experiment we assumed that simulation would last for 5 consecutive hypothetical days, starting from a set of orders for 20 castings (with 5 different alloy grades) with the parameters generated in the same way as described in [13]. Each day the *SalesAgent* would generate 3–5 new orders. Each time the *SchedulingAgent* used a 5-day planning horizon (weekend days were not considered, so the last day was 10). We were able to simulate 5 days in less than 2 hours.

Although the simulation run without problems we realize that still there are problems that need to be solved before such systems can be successfully implemented in practice. First of all, the interface of the *SalesAgent* has to be properly designed. In the simulation we assumed that this agent will operate automatically, but in a real production system such agent can work at most in a semi-automatic mode. Decisions whether to accept a due-date proposed by the *SchedulingAgent* must be made by a human decision maker who must consult these decisions with the customers. One of the most important problems to be solved at this stage is how to set a deadline for an answer from the *SalesAgent*, as the *SchedulingAgent* must know the answer in order to properly schedule the remaining orders.

Another demanding, and possibly the most costly task, is to connect the *ProductionAgent* and the *InvetoryAgent* to the transactional system that exists in the foundry and registers the actual production of castings (ERP or dedicated one like GUSS-INFO). Also a dedicated tool must be built for registering all the failures and other problems that occur in the production process and may affect the scheduling process. This tool must prepare messages to the *SchedulingAgent* and it must be operated by a human.

Finally, the *CooperatorAgents* must be designed in a realistic way. These agents have been completely omitted in our simulation, however in the real production system planners often need to consult cooperators, e.g. performing outside processing for the foundry, about the constraints that must be taken into account while building a production schedule. It is doubtful that this process can be done automatically.

5. Conclusions and future work

In the paper we have presented a complete framework for building a multi-agent system, able to generate proper and reliable production schedules for a foundry. The simulation conducted using a prototype of the system confirmed its potential usefulness. However, contrary to the simulation environment, in real production system some decisions cannot be taken automatically, so appropriate tools for human interaction have to be designed. This rises an important problem: how to set deadlines for the answers from such tools, as they block a natural flow of the messages in the multi-agent system. One of the possible ways to overcome this problem is to incorporate a knowledge system that contemporary is usually implemented in the form of rules management system. This will allow for more decisions to be taken automatically or semi-automatically, which means more fluent communication between the agents.

References

[1] Andreadis, G., Bouzakis, K.-D., Klazoglou, P. & Niwtaki, K. (2014). Review of agent-based systems in the manufacturing section. *Universal Journal of Mechanical Engineering*. 2(2), 55-59. DOI: 10.13189/ujme.2014.020204.

[2] Cowling, P., Ouelhadj, D., & Petrovic, S. (2004). Dynamic scheduling of steel casting and milling using multi-agents. *Production Planning and Control*. 15(2), 178-188. DOI:10.1080/09537280410001662466,

[3] Dangelmaier, W., Heidenreich, J. & Pape, U. (2005). Supply chain management: a multi-agent system for collaborative production planning. In The 2005 IEEE International Conference: e-Technology, e-Commerce and e-Service, 29 March - 1 April 2005 (pp. 309-314). DOI: 10.1109/ EEE.2005.128.

[4] Duda, J. & Stawowy, A. (2007). A model for efficient production planning and reliable order promising in supply chain. In J. Phelan (Ed.), *IMC 24: manufacturing: focus on the future* (pp. 553-560). Waterford Institute of Technology.

[5] FIPA ACL message structure specification, Foundation for Intelligent Physical Agents. http://www.fipa.org/specs/ fipa00061/SC00061G.pdf.

[6] Giret, A., Argente, E., Valero, S., Gómez, P. & Julian, V. (2005). Applying multi-agent system modelling to the scheduling problem in a ceramic tile factory. In Mass Customization Concepts-Tools Realization IMCM'05 (pp. 151-162). Berlin.

[7] He, D. & Babyan, A. (2003). Agent-based agile manufacturing system scheduling. In S. D'Amours & A. Guinet (Eds.), *Intelligent Agent-Based Operations Management* (pp. 87-108). Butterworth-Heinemann.

[8] Kobbacy, K.A.H., Vadera, S. & Rasmy, M.H. (2007). AI and OR in management of operations: history and trends. *Journal of the Operational Research Society*. 58(1), 10-28. DOI: 10.1057/palgrave.jors.2602132.

[9] Li, X., Zhang, C., Gao, L., Li, W. & Shao, X. (2010). An agent-based approach for integrated process planning and scheduling. *Expert Systems with Applications*. 37, 1256-1264. DOI: 10.1016/j.eswa.2009.06.014.

[10] Lima, R.M., Sousa, R.M. & Martins, P.J. (2006). Distributed production planning and control agent-based system. *International Journal of Production Research*. 44(18/19), 3693-3709. DOI:10.1080/00207540600788992.

[11] Ouelhadj, D., Petrovic, S., Cowling, P.I. & Meisels, A. (2004). Inter-Agent Cooperation and Communication for Agent-based Robust Dynamic Scheduling in Steel Production. *Advanced Engineering Informatics*. 18(3), 161-172. DOI: 10.1016/j.aei.2004.10.003.

[12] Pechoucek, M. & Marik, V. (2008). Industrial deployment of multi-agent technologies: review and selected case studies. *Autonomous Agents and Multi-Agent Systems*. 17(3), 397-431. DOI: 10.1007/s10458-008-9050-0.

[13] Stawowy, A. & Duda, J. (2013). Production scheduling for the furnace - casting line system. *Archives of Foundry Engineering*. *13*(3), 84-87. DOI: 10.2478/afe-2013-0065.

[14] Wilk-Kołodziejczyk, D., Kluska-Nawarecka, S., Rojek, G. & Regulski, K. (2014). The Implementation of Computer Platform for Foundries Cooperating in a Supply Chain. *Archives of Foundry Engineering*. 14(3), 111-116. DOI: 10.2478/afe-2014-0073.

[15] Zarandi, F.M.H. & Kashani, A.F. (2013). A type 2 fuzzy multi agent based system for scheduling of steel production. In 2013 Joint IFSA World Congress and NAFIPS Annual Meeting, 24 -28 June 2013 (pp. 992-996). Edmonton, Alberta, Canada.