ARCHIVES
of
FOUNDRY ENGINEERING

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

ISSN (1897-3310)
Volume 14
Issue 3/2014

111 – 116

23/3

The Implementation of Computer Platform for Foundries Cooperating in a Supply Chain

D. Wilk-Kołodziejczyk [a, b,] *, S. Kluska-Nawarecka [b], G. Rojek [a], K. Regulski [a]
[a] AGH University of Science and Technology, Mickiewicza 30, Cracow, Poland,
[b] Foundry Research Institute, Zakopianska 73, Cracow, Poland;
*Corresponding author. E-mail address: wilk.kolodziejczyk@gmail.com

Received 01.04.2014; accepted in revised form 15.05.2014

Abstract

This article presents a practical solution in the form of implementation of agent-based platform for the management of contracts in a network of foundries. The described implementation is a continuation of earlier scientific work in the field of design and theoretical system specification for cooperating companies [1]. The implementation addresses key design assumptions - the system is implemented using multi-agent technology, which offers the possibility of decentralisation and distributed processing of specified contracts and tenders. The implemented system enables the joint management of orders for a network of small and medium-sized metallurgical plants, while providing them with greater competitiveness and the ability to carry out large procurements. The article presents the functional aspects of the system - the user interface and the principle of operation of individual agents that represent businesses seeking potential suppliers or recipients of services and products. Additionally, the system is equipped with a bi-directional agent translating standards based on ontologies, which aims to automate the decision-making process during tender specifications as a response to the request.

Keywords: Application of information technology to the foundry industry, Innovative foundry technologies and materials, Multi-agent technology, Production support, Order management

1. Introduction

Currently, it is increasingly difficult for small plants with small capacities to operate in the metal processing market. Such companies have neither means nor infrastructural facilities to undertake large orders or orders for high-volume production. Hence, the aim of the developed computer system is to provide tools that will enable the consolidation of several plants in a production network in such a way as to allow participation of many companies in the implementation of one large contract. The basic idea is to allow potential cooperating companies to seek contractors for outsourced services. This problem can be solved with the use of a platform which is the aim of studies disclosed in this article. The platform was described from a theoretical point of view in a previous publication of the authors [1]. This publication shows the result of the next phase of work on the platform, i.e. the implementation and testing with a view to obtaining the opinions and suggestions related to the possibility of further development of the system aimed at practical implementation in the metal processing industry.

With the developed platform, each plant forming part of the consortium will receive information on the available orders placed by customers and will have the opportunity to send its own offer to the customer. As a result, the customer will receive many offers from which he can select the most equivalent ones, without the need for sending multiple orders to individual manufacturers potentially interested in cooperation. Automation of this process should contribute to reducing costs and accelerating the process of acquiring customers and thus it should increase the competitiveness of the whole industry sector.

2. The multi-agent system

Multi-agent system is the system that consists of a number of interacting software agents. Typically, agents are acting on behalf of users with different goals and intentions. To successfully interact, agents need the ability to cooperate, coordinate and negotiate with other agents. Agent systems are often used in planning (so-called agent-based planning). It is a technique of planning often used in problems with the changing environment, including real-time problems, where the problem parameters may vary during calculations.

Multi-agent systems are also used when there is the need to solve the problems of decentralised or distributed character, such as is the case of the system built for the cooperating companies. The usability of agents in the construction of decentralised systems stems from the basic characteristics of the agent, which is autonomy in action and the ability to cooperate with other agents through communication [1-5].

3. System implementation

The, anticipated in [1], agent-based system architecture provides support in the dispersion of system between computers of numerous offering tenderers and those who need services in the field of metallurgical industry. It has been anticipated that the customer (whether offering services or in need of certain services) will be represented by an agent supporting him in all basic operations associated with the preparation and analysis (including acceptance) of the service. To accelerate and facilitate the creation of such a specific system, the JADE Framework has been applied as the one that allows the use of ready-made classes for agents and provides support for the implementation and analysis of the agent activities. Among the agent activities of essential importance are the operations for sending and receiving messages, on which the transmission of offers between agents representing buyers and agents representing service providers has been based. The use of the JADE Framework involves the use of Java as a programming language, which is most desirable because of the main features of Java, which is independence of the operating system and ability to run code on a remote machine. This enables the use by industrial plants of different operating systems, and so does not force interference in the hardware or software used previously by individual cooperating companies.

4. The start up of the system

To run the system, one must first run the JADE platform, which is a common operating environment for agents responsible for the preparation and analysis of services. To run the platform it is needed to have a Java virtual machine running in the operating system. Technically, launching of the platform consists in loading the Boot class from a JADE package. To facilitate the addition of agents one can use the graphical user interface, which is obtained via the gui option. Full call to start the JADE platform is: java jade.Boot -gui, and then a window as shown in Figure 1 should appear.

Figure 1 shows the JADE platform, on which operate agents running by default. AMS agent provides a service called white pages, i.e. the basic operations of adding agents, removing agents, and closing of platform. DF agent provides a service called yellow pages, which is e.g. searching and registration of agents providing certain services. RMA agent is associated with the aforementioned graphical user interface that allows the performance of basic operations (such as e.g. adding an agent) in a graphical way .

When the JADE platform has been successfully launched, the next step is to run the agents representing the providers and recipients of certain services. The use of the agent mechanisms allows adding many different types of agents.

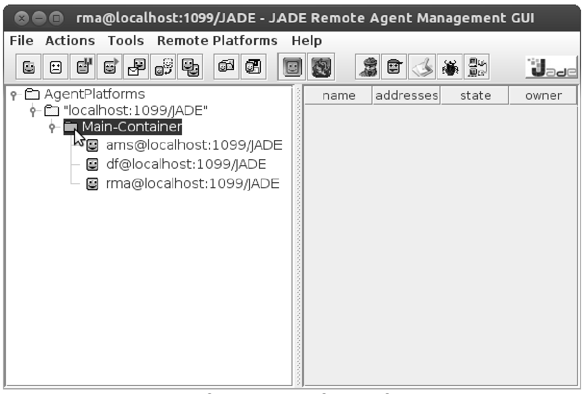

Fig. 1. View of JADE platform after launching

Figure 2 shows a system in which a single agent in search of services (called consumer_1) and two agents who represent producers (named p1 and p2) have been added.

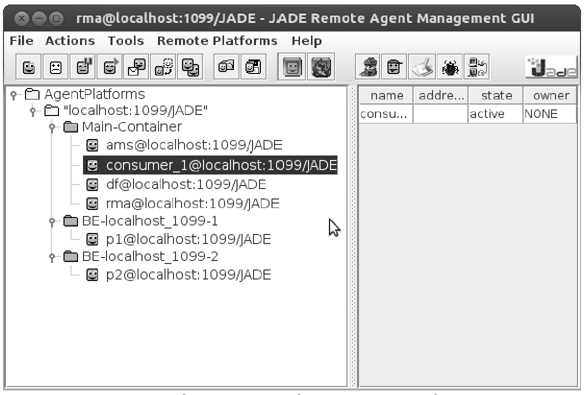

Fig. 2. View of JADE platform with the functioning manufacturer's agents and consumer agent

4.1. Preparation of order

The agent representing the company which aims at finding a contractor for certain services (hereinafter referred to as the consumer agent) has a graphical user interface shown in Figure 3.

Consumer agent allows entering in a graphic way the basic data relating to the order, among which one can mention the number of castings, a description of their shapes (with the possibility of adding a technical drawing in a graphical file format), casting dimensions (length, width, height), the cast material (grey cast iron, ductile iron, or malleable iron) and the required parameters of casting (minimum and maximum values of Rm, the maximum reduction of area, roughness and hardness of the casting). Consumer agent also provides automatic selection of the technology relative to the weight of a single casting, the maximum size of the casting, and volume of the repeated lot of castings produced. This algorithm works by eliminating technologies that do not fit into the technical conditions of the contract. The final choice of technology is made, however, by the contracting party from a list of those who are compatible with the technical specification.

The system and the knowledge base have been written in Polish for a Polish user, and hence the screen shot images of the system are also described in the Polish language.

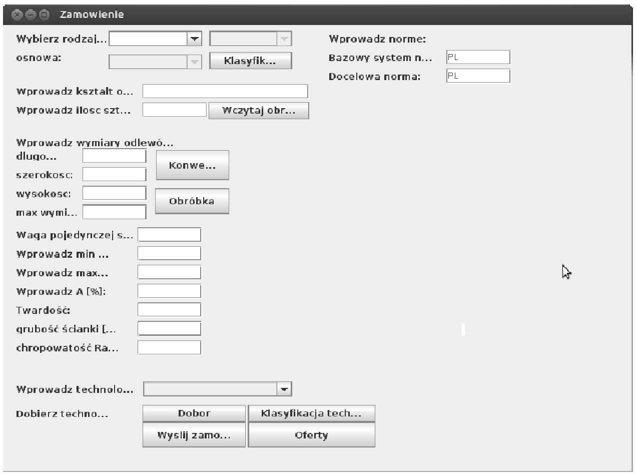

Fig. 3. The graphical interface of the consumer agent written in the Polish language - the language of system implementation

Consumer agent window which is filled with the sample data is shown in Figure 4. It contains a fully specified order, which is ready to send.

After selecting the appropriate parameters of the contract, order must be sent by clicking the "Send order" button. Clicking this button has the following consequences:

- consumer agent searches for all agents who may execute orders, and who operate within the platform,
- consumer agent sends a message to all the agents found with information containing the parameters of the contract,
- consumer agent moves in condition of waiting for a possible confirmation of offers from those who are willing to cooperate.

4.2. Replying to order – preparation of offer

Each agent representing a company that is interested in the adoption and execution of the order (hereinafter referred to as the manufacturer's agent) has a graphical interface that allows viewing, analysing and responding to orders that have been sent by consumer agents. This interface is shown in Figure 5.

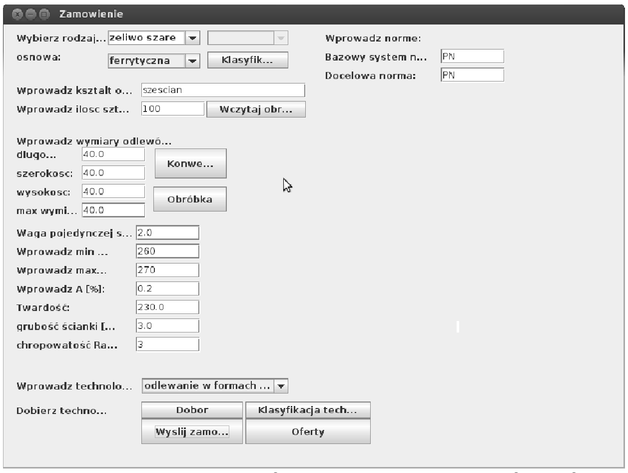

Fig. 4. The contract specified using graphical interface of the consumer agent

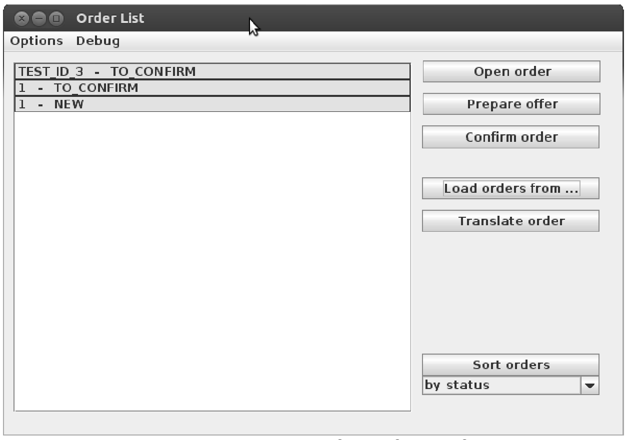

Fig. 5. The main graphical interface of manufacturer's agent

As mentioned in previous part of the study, the system has two manufacturer's agents (Fig. 2). The window shown in Figure 5 is an interface of one of the agents. Each agent has the same window allowing the execution of autonomous operations related with orders.

Orders shown in the window in Figure 5 have different status. The status of the first two orders is "TO_CONFIRM", which means that the manufacturer's agent responded to these orders by submitting his own terms, but has not received final approval of the contracting party. The status of the last order is "NEW", which means that it is a new order, which came to an agent. The answer to the new order should begin with reviewing all contract parameters specified by the consumer agent graphical interface, which is possible by clicking "Open order". Then the offer should be prepared by clicking the "Prepare offer" button, which displays the window shown in Figure 6.

Offer specification as a response to a specific request should include the price of the service, the currency and due date (or possibly additional information in the field called "Description").

After entering the information, click the "Send" button that sends the offer for the execution of order to the consumer agent who has placed the order. After clicking this button the order should change its status to "TO_CONFIRM" (Fig. 7).

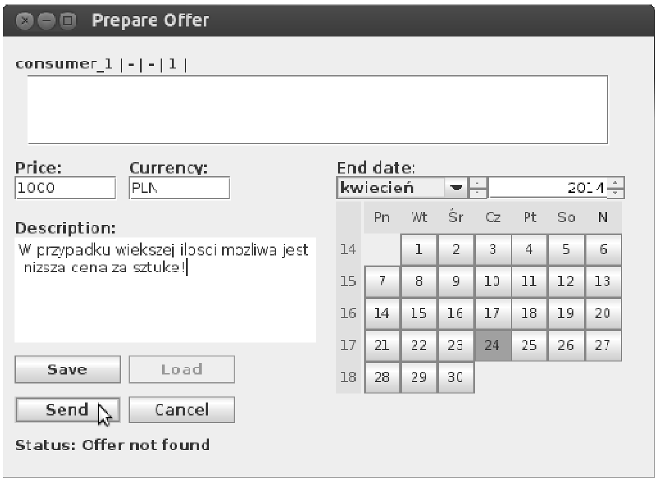

Fig. 6. Preparation of offer for the execution of contract

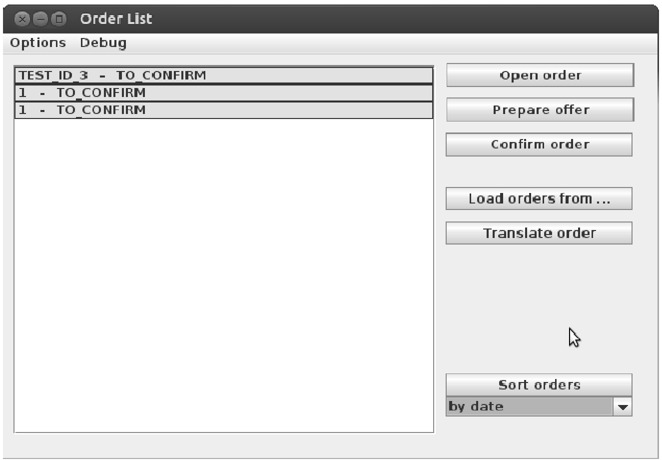

Fig. 7. Changing the status of order

4.3. Acceptance of offer

Acceptance of offer is done on behalf of the consumer agent who prepared the order and received at least one offer which is an answer from the agent representing the manufacturer. After clicking the "Offers" button in the window allowing specification of the order (Fig. 4), a graphical interface that allows viewing individual offers appears. It is shown in Figure 8.

As follows from the content shown in Figure 7, consumer agent has received 3 offers which are replies to the created order. The cheapest offer (1000 PLN) is also offering the shortest time of the order execution (by 24 April 2014), so the choice of this offer from among all the offers submitted by different agents that could provide the service is quite obvious. The selection is done by indicating the offer and clicking the "Accept" button.

Fig. 8. Consumer agent interface for browsing of offers

As a result of acceptance of the offer by the consumer, the contract also changes its status in the manufacturer's graphical interface by getting a value of "CONFIRMED". The "CONFIRMED" status means that the order has been accepted for processing at the plant, which is represented by a manufacturer's agent.

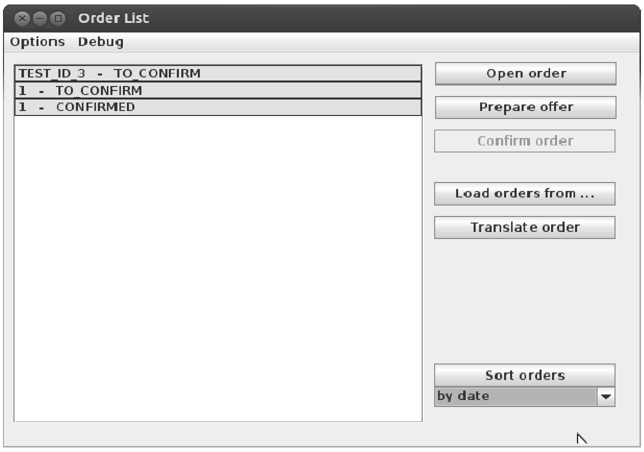

Fig. 9. View of the last order which was accepted for execution

5. Translation of orders

Since orders can be clarified by the standards not owned by a subcontractor, the system provides an agent that helps in processing the application and translates the standard that relates to a specific order to a standard owned by a particular establishment. The problem of the representation of domain knowledge, which is here represented by the standards, in such a way as to make it computer-processable has been discussed by the authors in [6-8].

Agent provides bi-directional translation system of standards and norms of production, and therefore it is hereafter called a translator. The aim of the agent is the translation of parameters in the order into the data valid in the applicable system of standards.

Translator uses in its action ontologies implemented in JADE. In JADE, ontologies, in addition to describing the reality in which the agents work, serve as a "dictionary" for agents to communicate. More detailed information about the ontologies the reader can find in [10, 11]. Normally, in JADE, the additional content of the messages sent by the agents is either a variable of the 'string' type, or a sequence of bits. At the time of transfer of objects one can choose serialisation or the use of ontology. Technically, both cases operate in the same way, but in the case of serialisation, the user is limited to the Java language and the form of the message is not clear for the people watching the operation of the system.

In JADE one can distinguish the following types of content present in the ontology :

- decisions - these are expressions describing the fragment of reality which can be depicted as true or false,
- expressions - these are the expressions that identify entities in the depicted reality to give to the agents using ontologies an idea about them. Additionally, the expressions are divided into [3]:
  - primitives - basic description of existence,
  - concepts - objects of more complex structure than the primitives, which can be defined as part of the expression,
  - actions of agents –special terms which can describe the actions performed by some agents,
  - aggregations – grouping the remaining terms,
  - IRE (Identifying Referential Expressions) - expressions that identify the object or objects for which the predicate assumes the value “true”,
  - variables - expressions used mainly in queries which indicate the object of not known priority.

In addition, all ontologies implemented in JADE need to define the dictionary. In the dictionary is contained the whole explicit description of the ontology. The language of the ontology should be defined as well. For this purpose, FIPA standard recommends to select the language SL, which saves the data in human-readable string.

The main objective of the project is to create a translator of standards that uses the ontology. The programme was supposed to be a universal translator for norms and standards, and therefore standardising the data structure was the main priority in the phase of its creation.

Thinking of what the standard really is, one can say that it is a set of parameters describing the properties of a thing. Each of these parameters is described with a class code which is a specific name of a given parameter. Each standard has also its own code and the name. Based on the above reasoning, in the project, each system consists of a list of standards and each item in the list consists of a description and specification. The creation of such relationships allows easy and fast access to data structures and creates a diagram of the appearance of classes in a programme.

The “translate” function searches for the base standard and the target standard, and then validates the standards and compares the values given in the order with the values given in the target standard. Then it prepares the comment of a string type that contains either information about the error, or ranges of values in the target standard.

In the first part of the operation, the function takes arguments temp, content and reply, then based on the data from temp it sets the base and target system. Systems that appear in the translator as a list of standards are assigned to the temporary list_base and list_target variables. The function checks if the assigned systems exist in the database, if not - an error is returned, then the function checks whether there is a standard in the base in which the order has been placed. With no existence of the base standard, an error is returned immediately. The situation is different in the case of the search for target standards.

At the beginning, the target standard is searched from a list of related standards. If searches to find the standard in the list fail, the target system is additionally searched using “searching by tags” function. In case of failure, also this time, the function returns an error. When the base and the target standard is located, Part 2 of the function follows, i.e. the translation.

In the second part, the specification is taken from the standard, from the specification are, in turn, retrieved UTS fields and hardness. They are sequentially checked to see if the values of these fields match the values specified in the order. If the two values match, a comment is being prepared containing information about the ranges of values in the target standard. The comment is placed in the reply variable and returned.

If none of the values matches, the reply is returned with the contents "Missing norm".

Below an example is given of the results obtained during testing of the programme. The values of all variables were determined at random. The case assumes that the standards are linked to each other in the form of a list of related standards. Additionally, data for the UTS are given in the ksi units and are to be compared to the data given in MPa.

Table 1.
Values of variables adopted in the test

Name	Value
Base system	ASTM
Code of standard	A01
Class of standard	A01-2
Target system	EN
UTS	16 ksi
Brinell hardness	150 HB

The data from Table 1 are sent to the translator by the test agent (as presented onto Fig. 10).

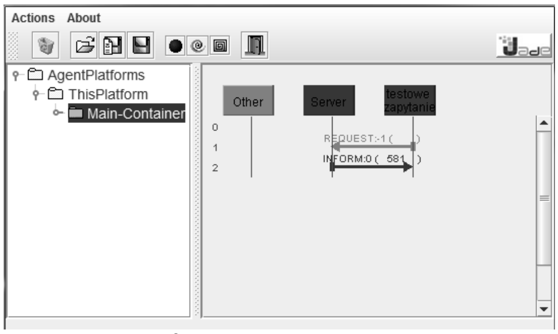

Fig. 10. Preview of agents communication, where Server is a translator and test question is a test agent

Communication in this case was successful because translator in response to a REQUEST message sent by the test agent answered INFORM and type of the message was set to “translated”. Additionally, the content sent by the translator satisfied the assumptions regarding the content of the comment that the test agent was to receive.

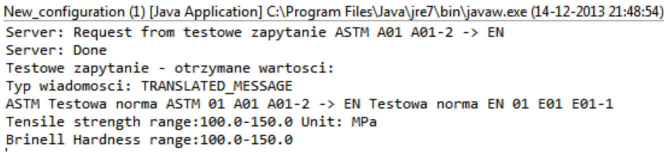

Fig. 11. View of the console after completed agents communication

Agent has been implemented in the way that it refers to defined standards. As a result, future development of the translator is possible just by expanding it with new standards. The translator went through the tests with positive result and returns all results as it was provided for in the phase of the project creation (Fig. 11). To ensure a high standard of the translator action, it must be provided with the current standards, but this comes at a cost.

6. Summary

Operations of agents presented in Chapter 4 correspond, in the authors’ opinion, to the most frequent case of use, which is associated with the search and selection of suppliers of services or products. In special cases, the course of described events may be supplemented by the optional use of an agent-translator which assists in the assessment of order and allows making translation between ASTM and EN standards. The intentions of the authors are to make a system that would also provide support for the economic analysis of the orders received, but this is a functionality that requires completion and refinement. The next phase of development of the disclosed research is in fact fine-tuning of the system in a way that will ensure its usability, which can occur only as a result of collecting and analysing the views of stakeholders from industry and science. Computer systems for defect detecting and quality control are known and well described [i.e. 9,12], also are computer aided simulations and optimization [13], but this system, implemented by the Authors, uses rather innovative methodology, and because of its decentralised architecture gives possibility for further development.

Acknowledgements

Financial support of the The National Centre for Research and Development (LIDER/028/593/L-4/12/NCBR/2013) is gratefully acknowledged.

References

[1] Kluska-Nawarecka, S., Regulski, K. & Rojek, G. (2013). The framework of agent-based negotiation platform for foundries cooperating in supply chain. *Archives of Foundry Engineering*. 13(4), 73-78.

[2] Wooldridge, M. (2001). *An Introduction to MultiAgent Systems*. New York: John Wiley & Sons, Inc.

[3] Bellifemine, F., Giovanni, C., Greenwood, D. (2007). *Developing multi-agent systems with JADE*, Chichester: John Wiley & Sons, Inc.

[4] Nawarecki, E. & Koźlak, J. (2010). Building multi-agent models applied to supply chain management. *Control and Cybernetics*. 39(1), 149-176.

[5] Macioł, A., Wrona, R. & Stawowy, A. (2010). An application of advanced information technology in foundry engineering. *Archives of Foundry Engineering*. 10(2), 83-88.

[6] Nawarecki, E., Kluska-Nawarecka, S., Regulski, K. (2012). Multi-aspect character of the man-computer relationship in a diagnostic-advisory system. In Z.S. Hippe, J.L. Kulikowski, T. Mroczek (Eds.), *Human – computer systems interaction: backgrounds and applications 2,* 85-102. Berlin Heidelberg: Springer-Verlag.

[7] Górny, Z., Kluska-Nawarecka, S., Wilk-Kołodziejczyk, D. & Regulski K. (2010). Diagnosis of casting defects using uncertain and incomplete knowledge. *Archives of Metallurgy and Materials*. 55(3), 827-836.

[8] Górny, Z., Kluska-Nawarecka, S. & Wilk-Kołodziejczyk, D. (2010). Attribute-Based knowledge representation in the process of defect diagnosis. *Archives Of Metallurgy And Materials*. 55(3), 819-826.

[9] David, J., Jancikova, Z., Frischer, R. & Vrozina, M. (2013). Crystallizer's desks surface diagnostics with usage of robotic system. *Archives of Metallurgy and Materials.* 58(3), 907-910.

[10] Regulski, K., Kluska-Nawarecka, S. (2012). *Knowledge integration computer tools and algorithms in the improvement of the production processes of cast-steel castings.* Kraków: Foundry Research Institute.

[11] Kluska-Nawarecka, S., Mrzygłód, B., Durak, J. & Regulski, K. (2010). Analysis of the applicability of domain ontology in copper alloys processing. *Archives of Foundry Engineering.* 10(2), 69-74.

[12] Pietrowski, S., Pisarek, B. & Gumienny, G. (2008). Computer-aided control of high-quality cast iron. *Archives of Foundry Engineering*. 8(1), 101-108.

[13] Kaczorowski, R., Just, P. & Pacyniak, T. (2013). Test bench for analyzing the Lost Foam Process. *Archives of Foundry Engineering*. 13(1), 57-62.