www.czasopisma.pan.pl PAN www.journals.pan.pl

ARCHIVES
of
FOUNDRY ENGINEERING

DOI: 10.1515/afe-2017-0055

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

DE GRUYTER OPEN

ISSN (2299-2944)
Volume 17
Issue 2/2017

79 – 84

Computer Catalog and Semantic Search of Data in the Domain of Cast Iron Processing

G. Rojek [a], K. Regulski [a], D. Wilk-Kołodziejczyk [b, *], S. Kluska-Nawarecka [b], T. Wawrzaszek [a]

[a] AGH University of Science and Technology, Faculty of Foundry Engineering,
Reymonta 23, 30-059 Kraków, Poland

[b] Foundry Research Institute, ul. Zakopiańska 73 30-418 Krakow, Poland

*Corresponding author. E-mail address: wilk.kolodziejczyk@gmail.com

Received 01.08.2016; accepted in revised form 28.09.2016

Abstract

The aim of this study is to design and implement a computer system, which will allow the semantic cataloging and data retrieval in the field of cast iron processing. The intention is to let the system architecture allow for consideration of data on various processing techniques based on the information available or searched by a potential user. This is achieved by separating the system code from the knowledge of the processing operations or from the chemical composition of the material being processed. This is made possible by the creation and subsequent use of formal knowledge representation in the form of ontology. So, any use of the system is associated with the use of ontologies, either as an aid for the cataloging of new data, or as an indication of restrictions imposed on the data which draw user attention. The use of formal knowledge representation also allows consideration of semantic meaning, a consequence of which may be, for example, returning all elements in subclasses of the searched process class or material grade.

Keywords: Application of information technologies in the field of foundry, Semantic search, Ontology, Cast iron processing

1. Introduction

The development and dissemination of computer technologies leads to their wide utilization also in studies of the cast iron processing [1-8]. The most common situation is storing the data resulting from various experiments in the form of text files, spreadsheets, workbooks or even database files [9]. In the most common case, knowledge of the contents of the file (type of material processed or type of the process) remains unwritten in any form, which leads to problems when after a long time the user searches for data in terms of its importance, that is, in terms of the knowledge that it contains [10].

The above presented problem can be eliminated through the use of an additional tool, which on the one hand, will not disrupt the current process of materials research, while on the other hand, will record a systematic knowledge comprised in the contents of each file. This tool should allow for:

1) cataloging any data sources in the form of files placed in any location,
2) searching for cataloged data taking into account the criteria oriented at semantic meaning.

It is further assumed that the tool should be universal by matching the specific nature of research and results. So, there should not be any restrictions on processes, materials, and file types. This can be achieved by separating the application, which is search engine, from knowledge relating to the areas within which this search engine operates. In this way, each time, the designed application uses ontologies that are formal representations of knowledge concerning processing operations, materials, or location of hardware resources

The use of formal knowledge representation is also crucial because it allows taking into account the importance of information in the process of searching cataloged files [11]. Semantics should be based on ontologies that describe the material and the process. In respect of fields that do not contain any hierarchy, a simple search may be used based on a comparison of the entered values. For hierarchical fields, the search should return results containing not only the desired term, but also records described by its subterms. For example, a user wants to search for all records describing the processes performed on the gray cast iron. In this case, the search engine will return not only all records containing the desired term, but also records that contain terms such as ordinary gray iron, ductile iron, inoculated cast iron and cast iron with vermicular graphite. Similar behavior should be observed in the case of search taking into account the criterion relating to the process.

The user should also have the ability to search for records based on the composition of particular material. In this case, the search engine must return the elements specified in the request and the range of the searched concentrations. The application should return results based on the conjunction of conditions given in the search specification.

2. System architecture

As shown in the Introduction, system architecture [12] should ensure the independence from system mechanisms operating in the area of knowledge to which the information cataloged and searched by the system refers. Generally, it is assumed that the system uses information concerning:

1) processing of cast iron,
2) the material to be processed (separately before and after treatment),
3) the location of computer resources (files).

In this way, formal knowledge representation is incorporated in the scope as stated above, which is illustrated in Figure 1.

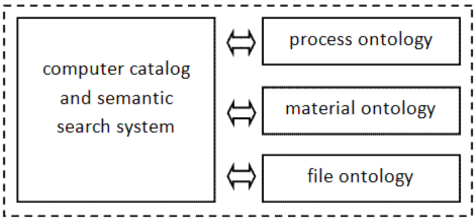

Fig. 1. System architecture

It was decided that the system would use three formal knowledge representations, which were named: process ontology, material ontology and file ontology [13]. This type of architecture of the data cataloging and retrieval [14-15] makes the system completely free from the specifics of the process, material, or the technique of saving the data in files [16].

The specific of the designed computer system assumes that the sources of data are cataloged in the native format as computer files. It means that no database is used nor created during the operation of the designed system. All data is represented by objects [17] in the computer language that is used.

2.1. Process ontology

The process ontology is a formal representation of knowledge [18-21] in the field of cast iron processing [22-23]. This ontology allows for grouping of processing operations, e.g. there is a general process group named "casting", which contains more detailed processes like "sand casting" or "die casting".

As presented in Figure 2, the ontological class “Process” contains one field of text type (String), which is intended to store the name of the process. The dependence "hasSpeciefiedProcess” corresponds to the grouping of processes. In this relationship, the range and the domain is the class of the process. So that, as it follows into dependency, it is possible to reach more detailed processes.

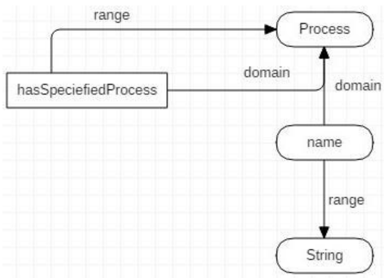

Fig. 2. The structure of the process ontology

2.2. Material ontology

The material ontology also provides the mechanism of grouping – it is important that the materials can be grouped and subgrouped [24]. An example of this might be the creation of a hierarchy for the material called "cast iron" containing less general subgroups such as "gray cast iron" and "white cast iron." Of course, nothing stands in the way of creating more accurately defined materials. This allows taking into account the more specific subgroups when searching for the overall group of materials or processes [25-26]. It is also possible to take into account data on the composition of materials, which enables searching for data on materials with specific composition, e.g. containing certain elements.

The structure of the material ontology is presented in Figure 3. The main class “Material” has four dependences:

- “hasSpecifiedMaterial” that is used to determine the more specific material in the same subgroup; this dependence enables the above discussed grouping mechanism,
- “isCreatedBy” that enables to handle the process (the ontology class Process), by which this material is made,
- “hasElements” that allows to handle the knowledge related to the chemical elements of the material (including the name of the element and its minimal and maximal concentration); this dependence has two subtypes: “hasAdditionalElements” that is related to the elements added during the defined

process and "hasAlloyingElements" that is related to the elements existed before the defined process,

- "usedResource" that enables to handle the knowledge according resources, which can be materials defined within the computer system; this dependence has also two subtypes: "receivedResource" that determines the resource obtained as a result of the process and "usedResourceAsBase" that determines the resource required for carrying out the process.

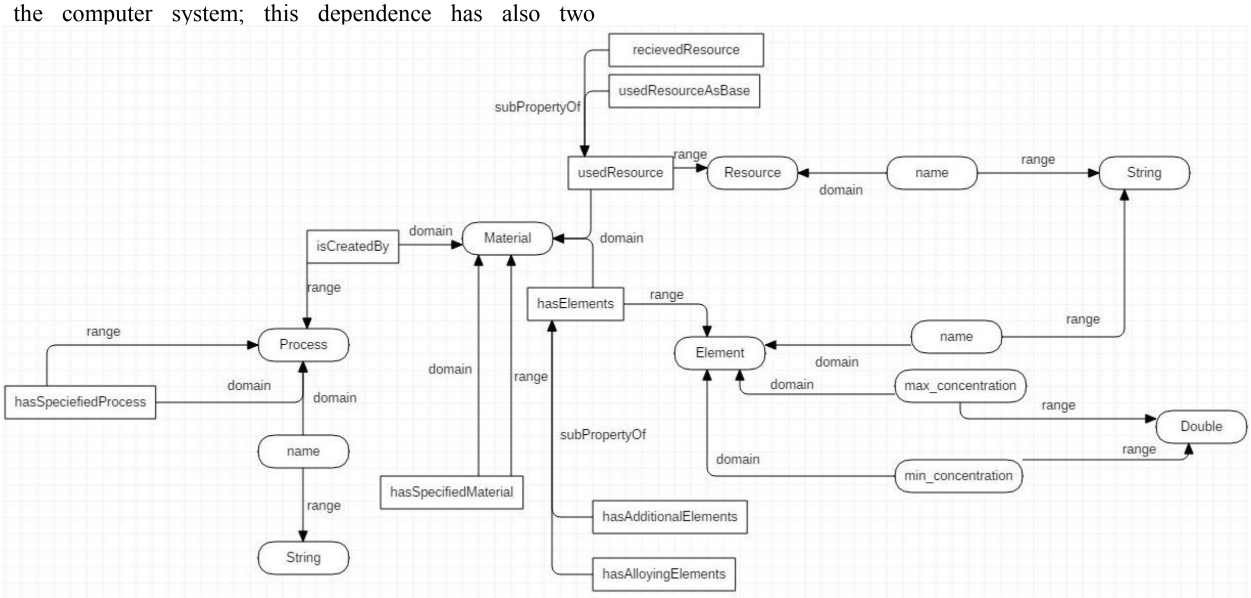

Fig. 3. The structure of the material ontology

2.3. File ontology

The file ontology represents knowledge about the location of computer resources, wherein the data is physically located. A computer resource is a document that is usually a file and is represented in the file ontology by the ontological class named "Record". As presented in Figure 4, this class enables to handle the knowledge according:

- author of the document,
- title of the document,
- document identification number,
- location of the document that is a file path,
- additional information.

The design of the file ontology makes the designed system free from some specific types of files or file saving locations.

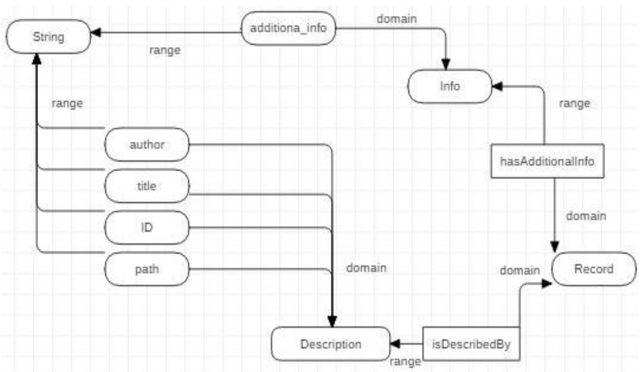

Fig. 4. The structure of the file ontology

3. Search algorithm

The designed system allows searching for data, which is included in its catalog. The catalog consists of documents that are individuals of the ontological class "Record". From a general point of view, each document represents the knowledge of:

- file: title, author, id (in accordance with the formal representation of knowledge contained in the file ontology),
- input material (in accordance with the formal representation of knowledge contained in the material ontology),
- output material (in accordance with the formal representation of knowledge contained in the material ontology),
- cast iron processing (in accordance with the formal representation of knowledge contained in the process ontology),
- alloying elements inherently present in the material,
- elements added in the course of processing,
- additional data.

The user has the option of specifying queries, which are aimed at finding knowledge that meets the criteria in respect of any point in the list above, or a conjunction of these criteria. The search algorithm analyzes any currently existing document according to the following scheme:

- If the "title" \ "author" \ "id" criterion is not selected, set the flag for the title \ author \ id to true. Otherwise, check whether the object title field contains the searched for term. If it does, set the flag for the title \ author \ id to true, if it does not, set the flag to false.
- If the "input material" \ "output material" criterion is not selected, set the flag for the input material \ output material

to true. Otherwise, check whether the object input material \ output material field is identical or if it is located in its sub-hierarchy. If it is, set the flag for the input material \ output material to true, if it is not, set the flag to false.

- If the "process" criterion is not selected, set the flag for the process to true. Otherwise, check whether the object process field is identical or if it is located in its sub-hierarchy. If it is, set the flag for the process to true, if it is not, set the flag to false.
- If the "alloying elements" \ "elements added" criterion is not selected, set the flag for the alloying elements to true. Otherwise, check whether the searched alloying elements \ elements added are present in the object and whether the ranges of their concentration overlap. If they do, set the flag for the alloying elements \ elements added to true, if they do not, set the flag to false.
- If the "additional data" criterion is not selected, set the flag for the additional data to true. Otherwise, check if the object additional information field contains the term searched for the title. If it does, set the flag to true, if it does not, set the flag to false.

If all the flags of a record are true, it means that the record meets the criteria specified by the user. In this case, the record is added to the list of solutions.

The searching process proceeds every document existing in the system with the algorithm that is presented above. After analyzing each record, a list of solutions is presented to the user as a list of records that fulfill the desirable criteria. The searching process enables to take into account the semantic meaning of information e.g. for the "process" criterion the hierarchy of processes is considered. This is consisted with the overall goal of the research presented in the Introduction.

4. System implementation

The system for cataloging and semantic search of data has been implemented using the Java language [27]. This makes the use of the system independent from the operating system, which features a user.

The main functionalities of the system include cataloging the files with the data on the methods of cast iron processing and searching this data taking into account the semantic meaning, which is done with the use of ontologies previously described. Implementation of major functionalities has been separated and included in the engine package. Figure 5 contains an UML class diagram [28-29] for the engine package. The following packages were also created in the project:

- item package, which contains classes that implement the basic concepts used mainly to describe the field of ontology,
- model package that implements models used by the GUI and models used to create a hierarchy of ontology,
- ui package, which is responsible for all contacts with the user. It contains the class that is the main called window, and the auxiliary windows and panels responsible for the appearance of the entire graphical user interface [30-31].

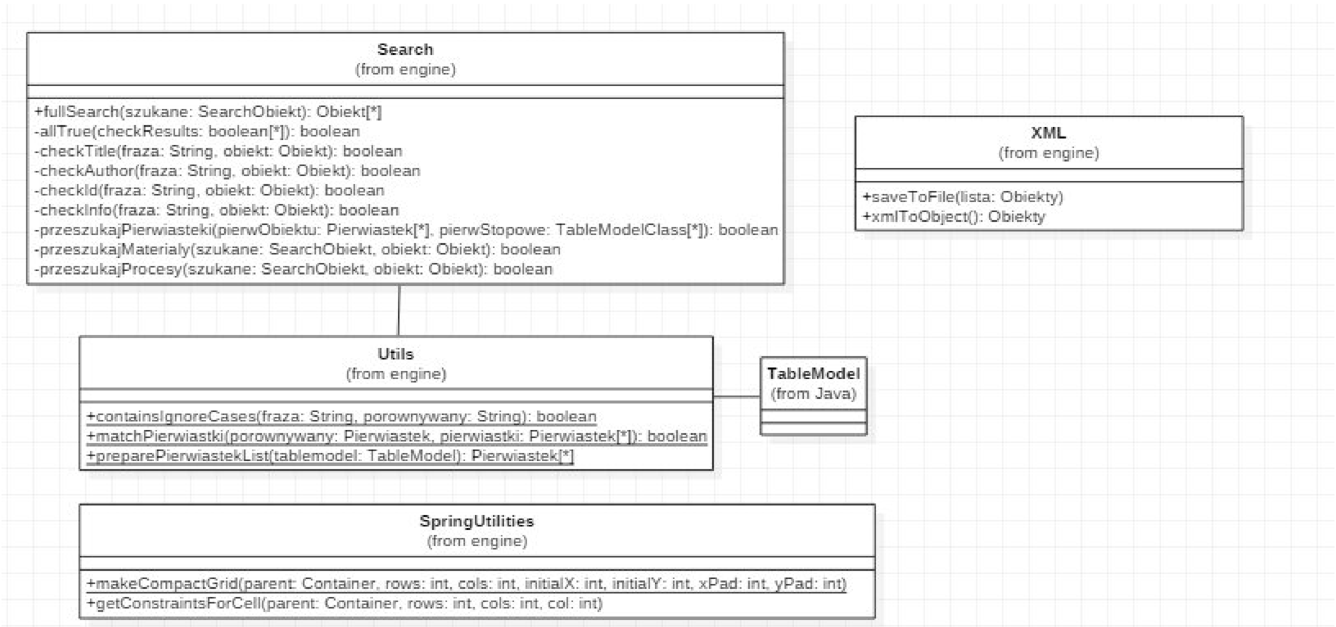

Fig. 5. UML class diagram of the engine package

5. Use of the system

After launching the application, the main window appears where the user can choose one of the following options:

- review an existing document,
- add a new document to the computer system,
- view and edit the set of documents,
- search for documents in the computer system.

As part of the studies performed using the implemented computer system, a series of tests on the above mentioned options has been made. One of the tests consisted in an attempt to find information on meeting all of the criteria listed below (as shown in Figure 6):

- the concentration of carbon in alloying elements in the range of 3 to 4 percent,
- the process of machining.

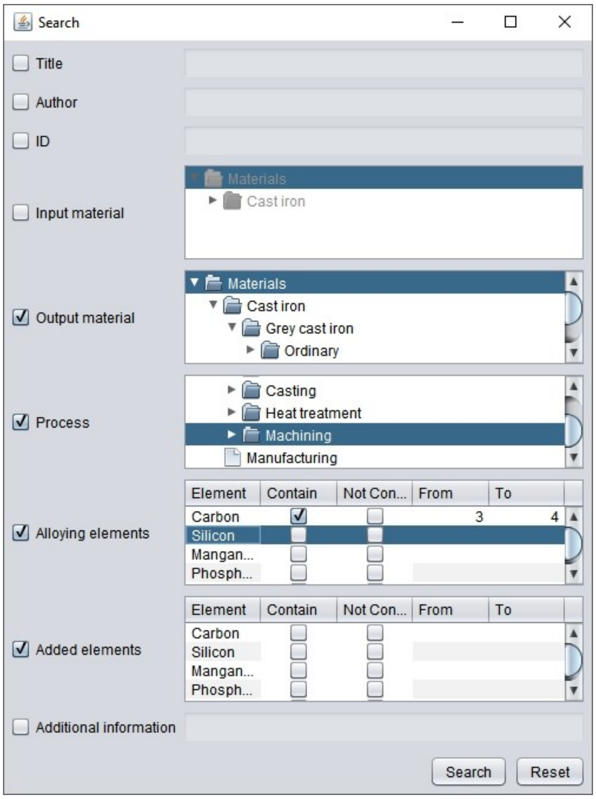

Fig. 3. The graphical user interface allowing for identification of the search criteria

As a result of the conducted test, the system has returned four documents properly filtering data that do not meet the specified search criteria.

6. Conclusions

The goal of presented here research is the design and the implementation of the computer system, which allows cataloging of documents related to the cast iron processing. As a document a simple computer file is considered. The developed system enables to add semantic information to every document, which relates to basic knowledge according the material and the type of the cast iron processing. The semantic information allows search of cataloged documents considering the semantic meaning – e.g. for the "process" criterion the hierarchy of processes is taken into account. Performed tests on the semantic search among sample documents indicate correctness of the system functioning and achievement of the goal of presented here work.

The performed system does not intervene in the content of documents. It is also assumed lack of homogeneity of cataloged documents – some of them can ignore geometry issues or other important elements involved in the cast iron processing. This conviction caused the decision that only basic knowledge about the process and the material is considered by the performed cataloging and semantic search [32]. Adding information relating to geometrical issues and other important parameters is possible however requires expansion of developed ontologies.

Acknowledgements

Financial support of The National Center for Research and Development (LIDER/028/593/L-4/12/NCBR/2013) is gratefully acknowledged together with the financial support of the Ministry of Science and Higher Education (AGH 11.11.110.300).

References

[1] Daber, S., Ravishankar, K., S., Rao, P., P. (2008). Influence of Austenitising Temperature on the Formation of Strain Induced Martensite in Austempered Ductile Iron. *Journal of Material Science.* 43, 4929-4937.

[2] Putatunda, S.K. (2001). Development of Austempered Ductile Cast Iron (ADI) with Simultaneous High Yield Strength and Fracture Toughness by a Novel Two-step Austempering Process. *Materials Science and Engineering A.* 315, 70-80.

[3] Bayati, H. & Elliot, R. (1999). The Concept of an Austempered Heat Treatment Processing Window. *International Journal Cast Metals Research.* 11, 413-417.

[4] Tybulczuk, J. & Kowalski, A. (2003). *ADI. The properties and application in industry. Castings Atlas*, Instytut Odlewnictwa, Kraków (in Polish).

[5] Gumienny, G. (2013). Carbidic Bainitic and Ausferritic Ductile Cast Iron. *Archives of Metallurgy and Materials.* 58, 1053-1058.

[6] Olejarczyk-Wozenska, I., Adrian, H. & Mrzyglod, B. (2014). Mathematical model of the process of pearlite austenitization, *Archives of Metallurgy and Materials*. 59(3), 981-986.

[7] Pietrowski, S. (1998). Alloyed Cast Iron with Compacted Graphite. *Solidification of Metals and Alloys*. 37, 105-111. (in Polish).

[8] Cueva, G., Sinatora, A., Guesser, W.L. & Tschiptschin, A.P. (2003). Wear resistance of cast irons used in brake disc rotors. *Wear.* 255, 1256–1260.

[9] David, J., Švec, P., Garzinová, R., Kluska-Nawarecka, S., Wilk-Kołodziejczyk, D. & Regulski, K. (2016). Heuristic modeling of casting processes under the conditions uncertainty. *Archives of Civil and Mechanical Engineering*. 16(2), 179-185.

[10] Kluska-Nawarecka, S., Regulski, K., Krzyżak, M., Leśniak, G. & Gurda, M. (2013). System of semantic integration of

non-structuralized documents in natural language in the domain of metallurgy. *Archives of Metallurgy and Materials. 58(3)*, 927-930. DOI: 10.2478/amm-2013-0103

[11] Kluska-Nawarecka, S., Śnieżyński, B. Parada, W., Lustofin, M. & Wilk-Kołodziejczyk, D. (2014). The use of LPR (logic of plausible reasoning) to obtain information on innovative casting technologies. *Archives of Civil and Mechanical Engineering*. 14(1), 25-31.

[12] Ozkarahan, E. (1991). System architecture for information processing. *Information Processing & Management.* 27(4), 347-369. DOI: 10.1016/0306-4573(91)90089-5.

[13] Regulski, K., Wilk-Kołodziejczyk, D., Rojek, G., Kluska-Nawarecka, S. & Adrian, W.T. (2016). Application of ALSV(FD) logic and XTT knowledge representation in the range of ADI properties. *Archives of Foundry Engineering*. 16(2), 75- 78.

[14] Frakes, W. B., Baeza-Yates, R. (1992). *Information Retrieval Data Structures and Algorithms*. Englewood Cliffs, NJ: Prentice-Hall, Inc.

[15] Grossman, D. A., Frieder, O. (1998). *Information Retrieval: Algorithms and Heuristics*. Norwell, MA, USA: Kluwer Academic Publishers.

[16] Lesk, M. (1997). *Practical Digital Libraries: Books, Bytes, and Bucks*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

[17] Booch, G., Maksimchuk, R., Engle, M., Young, B., Conallen, J., Houston, K. (2007). *Object-Oriented Analysis and Design with Applications*. (3rd ed.). Redwood City, CA, USA: Addison-Wesley Professional.

[18] Uzdanaviciute, V. & Butleris, R. (2011). Ontology-based Foundations for Data Integration, BUSTECH The First International Conference on Business Intelligence and Technology. 34-39.

[19] Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D. & Patel-Schneider, P.F. (2003). *The description logic handbook: theory, implementation, and applications*.

[20] Cristani, M. & Cuel, R. (2005). A Survey on Ontology Creation Methodologies, *International Journal of Semantic Web Information Systems 1*. 49–69.

[21] Natalya, D.L.M. & Noy, F. (2005). *Ontology Development 101: A Guide to Creating Your First Ontology*.

[22] Mrzygłód, B., Kowalski, A., Olejarczyk-Wożeńska, I., Adrian, H., Głowacki, M. & Opaliński, A. (2015). Effect of heat treatment parameters on the formation of ADI microstructure with additions of Ni, Cu, Mo. *Archives of Metallurgy and Materials*. 60(3A), 1941-1948.

[23] Hartmann, D., (2006). Process management and virtual engineering In foundries, Foundry – Science and Practice, 50, COCAFTEC, Foundry Research Institute, Cracow.

[24] Angele, J. & Gesmann, M. (2006). Data integration using semantic technology: a use case, *Rules and Rule Markup Languages for the Semantic Web*. 58-66, DOI: 10.1109/RULEML.2006.9

[25] Smyksy, K., Ziółkowski, E., Wrona, R. & Brzeziński, M. (2013). Performance evaluation of rotary mixers through monitoring of power energy parameters. *Archives of Metallurgy and Materials*. 58(3), 911-914.

[26] Glowacz, A. (2015). Recognition of Acoustic Signals of Loaded Synchronous Motor Using FFT, MSAF-5 and LSVM. *Archives of Acoustics.* 40(2). 197-203. DOI: 10.1515/aoa-2015-0022.

[27] Richardson, C. (2006). Untangling Enterprise Java. Queue - *Component Technologies.* 4(5), 36-44. DOI: 10.1145/1142031.1142045

[28] Berardi, D., Calvanese, D., De Giacomo, G. (2005). Reasoning on UML class diagrams. *Artificial Intelligence.* 168(1), 70-118. DOI: 10.1016/j.artint.2005.05.003.

[29] Fowler, M., Scott, K. (1997). *UML distilled: applying the standard object modeling language*. Essex, UK: Addison-Wesley Longman Ltd.

[30] Mullet, K., Sano, D. (1995). *Designing Visual Interfaces: Communication Oriented Techniques*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc.

[31] Galitz, W. O. (1997). *The Essential Guide to User Interface Design: An Introduction to GUI Design Principles and Techniques*. New York, USA: John Wiley & Sons, Inc.

[32] Rojek, G., Cieciwa, R., Cetnarowicz, K. (2006) Heterogeneous behavior evaluations in ethically-social approach to security in multi-agent system In 6th International Conference on Computational Science (ICCS 2006), 28-31 May 2006 (823-830) Lecture Notes In Computer Science.