D. WILK-KOŁODZIEJCZYK*,**,#

SUPPORTING THE MANUFACTURING PROCESS OF METAL PRODUCTS WITH THE METHODS OF ARTIFICIAL INTELLIGENCE

The aim of this research programme was to develop a series of methods and solutions to support the decision-making process in foundry and materials engineering. The specific problems discussed included the selection of methods for data processing and knowledge representation formalisms, backed up by the creation of decision algorithms based on contemporary achievements of artificial intelligence, tailored to the needs of foundry industry and metallurgy. The manufacturing process of metal items is associated with many aspects, which affect the quality of end product. For process engineers responsible for the supervising and planning of production, an important feature is, among others, the diversified nature of numerous aspects of the knowledge acquisition and integration from distributed sources of information which, when made available in an appropriate manner, can support the improvement of manufacturing process.

Keywords: manufacturing process of metal products, decision algorithms, the implementation of IT solutions

1. Introduction

The problem of the diagnosis of defects in castings, comprising the determination of the type of defect, the causes of its formation and identification of the relevant preventive measures, has for years been the subject of interest of many research centres. As is clear from the world literature, there are many solutions for computer-aided diagnostic systems, but for the time being they still do not provide fully satisfactory results. It seems that one of the causes of a stagnation in the field of IT solutions supporting the diagnosis of cast products is the fact that too little attention has been devoted to the essential characteristics of the data underlying diagnoses. The problem most important among these features is the incompleteness and uncertainty, the distributed nature of information sources and linguistic forms of communication - both between the source and the system, and between the system and the user. Therefore, as a guiding idea in the studies described below has been adopted the search for the formalisms of knowledge engineering, for the intelligent decision-making procedures, and for the system and software solutions that enable these characteristics to be integrated and, consequently, improve the efficiency of the diagnostic process and better adapt the available solutions to the needs of the user.

The study described in [1] introduced the concept of the representation of knowledge about casting defects in the form of an array of attributes. The table presented there integrated information used for the description of defects, the causes of their occurrence and means to prevent them. The table was based on the analysis of descriptions included in national and international standards. A list of the attributes of defects found in any of the considered sources was developed, thus giving rise to the definition of a set of attributes. For each of these attributes, the domains which contained the respective values were defined. The essential difference between the common decision-making tables and attribute tables built here lies in the fact that in the embodiment proposed here some of the fields are empty, thus reflecting the incomplete nature of the available knowledge and leading to the application of the principles of the rough set theory. Directly from the table one can define a logical expression (for the case under consideration) and the name of the defect as a conjunction of attribute values associated with it.

It is worth noting that this form of knowledge representation about the defects takes into account the incomplete character of the knowledge but, at the same time, enables the integration of information from distributed sources, applying finally the linguistic character of description, which helps the user to make his own interpretation.

The subsequent articles [2,3,4] present a summary of various formalisms enabling the representation of knowledge incomplete and uncertain coming from distributed sources. Large fragments of these works are related to the presentation of domain knowledge using formalisms applied in the logic of plausible reasoning (LPR). The novelty here is the use of LPR to write this kind of knowledge by describing the relationships between the sets of defects, their characteristics and causes of formation, assigning parameters of the uncertainty of information that have a significant impact on the outcome of inference and reliability of the results obtained. In parallel, the record shows a similar piece of knowledge using the formalism of fuzzy logic. For the presentation of results, as well as the construction of rules and checking the validity of inference, a MATLAB package with the module of fuzzy logic was used.

* AGH UNIVERSITY OF SCIENCE AND TECHNOLOGY, AL. A. MICKIEWICZA 30, 30-059 KRAKÓW, POLAND
** FOUNDRY RESEARCH INSTITUTE, ZAKOPIANSKA 73 STR., 30-418 KRAKÓW, POLAND
# Corresponding author: dwilk@agh.edu.pl

To test the knowledge base and inference procedures based on the use of LPR, a proprietary version of the prototype decision-making and advisory system implemented in the AGH Department of Computer Science was used.

2. Methods

The experimental results confirming that the methods of reasoning used by humans can be represented in the LPR are presented in subsequent papers [4, 5]. The objective set by the creators has caused that LPR is significantly different from other known knowledge representation methods mentioned in the Introduction (rule-based systems, classical logic, fuzzy logic, Demster-Shafer theory, Bayesian networks, rough sets). Firstly, there are many inference rules in LPR, which are not present in the formalisms mentioned above. Secondly, many parameters are specified for representing the uncertainty of knowledge. Studies described in [6] present RESCUER, a UNIX shell support system. By tracing changes in the file system and knowledge of the interpreter commands, the system is able to recognize the wrong commands and suggest appropriate substitutes. In this work only small part of LPR is used. In [7] LPR-based tutoring tool is presented. Knowledge for teaching is represented in LPR and the system is able to infer what the student should know after learning.

Another field of formalism application is presented in the work done by Cawsey [8]. It discloses a system generating a description of the concepts based on the recipient's model, taking into account his/her current knowledge. Hierarchies' concept is also core element of ScubAA system [9], recommending best services in analyzed context, e.g. most accurate Internet search engines. It stores system knowledge in a tree, automatically updated during reasoning process and according to users' feedback. Moreover, in comparison to LIIS, solution uses only three transformations (generalization, specialization and similarity) and limits statements to hierarchy-related.

On the basis of LPR, a DIH (Dynamically Interlaced Hierarchies) formalism was developed [10, 11]. Knowledge consists of a static part represented by hierarchies and a dynamic part, which are traces, playing a role similar to statements in LPR. The DIH distinguishes three types of hierarchies: types, components and priorities. The latter type of hierarchy can be divided into subclasses: hierarchies of measures (used to represent the physical quantities), hierarchies of quantification (allowing quantifiers to be included in traces, such as e.g. one, most, or all) and hierarchies of schemes (used as means for the definition of multi-argument relationships and needed to interpret the traces).

Research on LPR applications has been also performed at the AGH University. It concerned, in particular, diagnostics, knowledge representation and machine learning [12, 13].

2.1. LPR-Based Information System

Specialized software allowing to make inferences in LPR^0 have been developed. The goal of the system is to provide an inference tool for searching items matching a description given by the user. Architecture of the system is presented in Fig. 1. All the domain knowledge is stored in a Knowledge Base (KB). When user searches for the information, it provides a query, which is translated into LPR query formula and proof searching algorithm is executed. It returns items matching the query sorted by label values. Query q has the following form:

$$q = \alpha_1 \bigwedge \alpha_2 \bigwedge \ldots \bigwedge \alpha_m \tag{1}$$

where α_i has a form of $V(o_i, a_i, v_i)$ or $P(v_i, v_i')$. Query q represents description of some concept the user is looking for. It usually contains some variable representing the concept, and the inference algorithm returns substitution for this variable, which represents an answer.
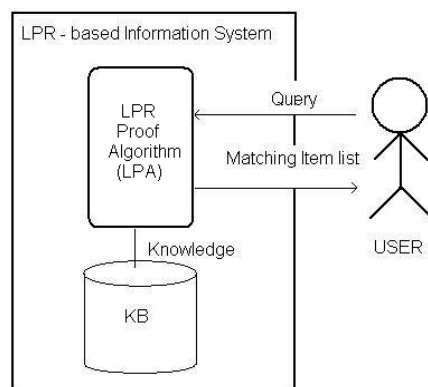

Fig. 1. Architecture of LPR-based Information

3. Casting technologies knowledge base

In this section we present sample Knowledge Base describing casting technologies. It was prepared to demonstrate LPR^0 reasoning method. Figure 2 presents cast material hierarchy.

Software for the inference engine is developed in Java [11]. In Fig. 2. one can see the class diagram of the main package. Engine class is responsible for proof searching. Knowledge
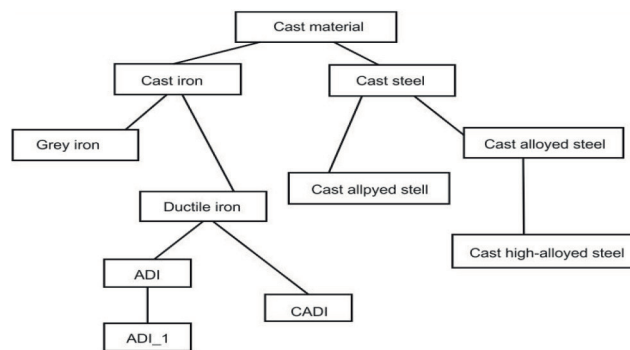

Fig. 2. Hierarchy of cast materials

Base stores all the knowledge available for the inference engine. Proof represents the proof tree being built. It is connected with sub trees by Vertex class. Substitution and Unification Exception classes are used to process variables.

For example, if we have to introduce the information that folds as a type of damage are defined as scratches, wrinkles, and erosion scabs in the description of defects, this statement will be introduced by writing down the following notation:

typeDefect (fold)={scratch, wrinkles, erosion scab,..}

On the basis of defined concepts, a software was developed in the LPR. To the database, the above Despite the fact that, as regards the damage type, these two defects are determined in the attribute table by the same parameters, it is not possible to state if they cover entirely the same area. One of the defects folds is from the Czech classification, while another fold comes from the Polish classification. Although the definitions of the damage type and shape are the same, the other definitions differ, which can suggest that the case may not entirely refer to the same defect, and this fact should be allowed for in the system of inference.

Yet, to prevent introducing into the system the terms on the damage type separately for the defect fold and for the defect folds, and for other similar cases, a notation of similarity has been added. The similarity may occur as a similarity of arguments or of values. In this case we are dealing with the similarity of arguments.

Mentioned notation has been introduced as:

V (typeDamage, fold, scratch)
V (typeDamage, fold, wrinkles), etc.

The next step is to write down similar parameters listed in the attribute table to identify various defects.

For example, in the case of defect fold and defect folds that occur in the Polish and Czech classification, the determined distribution of attributes in the table for the location and shape of damage is the same.

So, it means that the following statements will be introduced:

typeDamage(fold)=(scratch, wrinkles, erosion scab)
typeDamage(folds)=(scratch, wrinkles, erosion scab)
typeDamage (fold) = (scratch, wrinkles, erosion scab)
fold SIM folds in CX (defect, defect name (defect))
defect name (defect) damage type (defect)
fold SPEC defect
folds SPEC defects
typeDamage (folds) = (scratch, wrinkles, erosion scab)

In the database, such a statement is introduced by means of the term.
Similarity
S (fold, folds, typeDamage)

Relationships
E(fold, nameDefect, fold, typeDefect)
E(folds, nameDefect, folds, typeDefect)

However, a restriction should be put here. A notation like this can be entered into the system if and only if for at least two different defects the values of one of the conditional attributes are equal.

The second example of the practical use of this type of notation occurs when a system reasoning is based on user responses, or on other external factors and the data are incomplete. The problem is that while preparing a system we never know the type of the data that will be supplied from the outside.

If this is the case, then it is recommended to look for the internal links in the system, to enable deducing some of the parameters based on similarities. Besides, it is also recommended to save all the rules that can somehow affect the creation of other parameters, e.g. having only the location and shape of the defect, its name should be given in approximation.

Here it is also important that the diagnosis or the outcome of reasoning with a small number of the available attributes was defined as a result with the low parameter g, which is the formula confidence index.

It is important to remember in the process of reasoning about the parameters of uncertainty. In the example, the following parameters have been used: y1φ

The parameters are responsible for the formula degree of certainty. The parameter ma is responsible for the multiplicity of argument, s for the degree of similarity between concepts in a given context, while a is the force with which the left side of the implication affects the right side. At present, the introduction of these parameters is done manually for each statement entered into the system, which is a great impediment. Therefore the intention is to develop an automatic input of these parameters, as they affect not only the reasoning process, but also an outcome of this process.

The value of these parameters determines whether the result of inference is true. The mechanism works on the principle that the lower is the value of the parameter g, the lower is the value of the result of inference, meaning that it is less probable. This allows using the LPR in reasoning, e.g. in expert systems creating a ranking of the results, which will enable arranging them in certain order, e.g. by relevance of the diagnoses, which is highly recommended in the case under consideration (e.g. the system in which the reasoning will be conducted will have as a resultant task specifying the names of foundries which have the technical potential sufficient to produce the material searched for). Owing to the applied parameters of uncertainty, the obtained result will have the form of a list of companies where on the top of the list will be placed the foundry plants for which the inference has the highest values of the parameters).

4. Conclusion

The developed method opens the possibility to increase the efficiency of computer-assisted foundry processes after the introduction of information and decision-making systems based on artificial intelligence, the use of which allows covering the subject area with a much wider range of data, hitherto overlooked because of its incompleteness, uncertainty, heterogeneity and the fact that many of the requirements in respect of foundry technology are of an intuitive character. In the proposed embodiment, until now, these formalisms have never been used in the task of creating innovative production technologies, processing of domain knowledge, or diagnosis of casting defects.

So, it can be said that the problems already known have been given a new dimension due to the innovative approach to solving them. Scientific qualities of the developed solutions are also confirmed by the fact that they constitute a methodology by means of which a knowledge base is created and used not only in the foundry industry, but also in other areas of materials technology.

REFERENCES

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

[2] K. Regulski, S. Kluska-Nawarecka, D. Wilk- Kołodziejczyk, Practical aspects of knowledge integration using attribute tables generated from relational databases, in: Studies in Computational Intelligence, Springer-Verlag **381**, 13-22 (2011).

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

[4] S. Kluska-Nawarecka, D. Wilk-Kołodziejczyk, K. Regulski, G. Dobrowolski, Rough Sets Applied to the RoughCast System for Steel Castings, in: Intelligent Information and Database Systems **6592,** 52-61(2011).

[5] D. Boehm-Davis, K. Dontas, R.S. Michalski, A validation and exploration of the Collins-Michalski theory of plausible reasoning. Tech. rep., George Mason University, 1990.

[6] M. Virvou, B.D. Boulay, Human plausible reasoning for intelligent help. User Modeling and User-Adapted Interaction 9, 321-375 (1999).

[7] M. Virvou, A cognitive theory in an authoring tool for intelligent tutoring systems, in: A.E. Kamel, K. Mellouli, P. Borne, (Eds.) Proc. of the 2002 IEEE International Conference on Systems, Man and Cybernetics 2002.

[8] A. Cawsey, Using plausible inference rules in description planning. In: Proc. of the Fifth Conference of the European Chapter of the Association for Computational Linguistics (EACL-91). Congress Hall, Berlin, 1991.

[9] S. Abedinzadeh, S. Sadaoui, A trust-based service suggestion system using human plausible reasoning. Applied Intelligence, 2014.

[10] M.R. Hieb, R.S. Michalski, Multitype inference in multistrategy task-adaptive learning: Dynamic interlaced hierarchies. Tech. rep., George Mason University, 1993.

[11] M.R. Hieb, R.S. Michalski, A knowledge representation system based on dynamically interlaced hierarchies: Basic ideas and examples. Tech. rep., George Mason University, 1993.

[12] B. Śnieżyński, S. Kluska-Nawarecka, E. Nawarecki, D. Wilk-Kołodziejczyk, D. Intelligent information system based on logic of plausible reasoning, in: Issues and Challenges in Artificial Intelligence, Springer International Publishing, 57-74, 2014.

[13] S. Kluska-Nawarecka, E. Nawarecki, B. Śnieżyński, D. Wilk-Kołodziejczyk, The recommendation system knowledge representation and reasoning procedures under uncertainty for metal casting. Metalurgija **54** (1), 263-266 (2015).