Materials Research Proceedings **17** (2020) 100-107 https://doi.org/10.21741/9781644901038-15

Implementation of Fuzzy Logic in Industrial Databases

KARPISZ Dariusz[1,a*] and KIEŁBUS Anna[1,b]

[1]Cracow University of Technology, Faculty of Mechanical Engineering, Institute of Applied Informatics, al. Jana Pawła II 37, 31-864 Krakow, Poland

[a]dkarpisz@pk.edu.pl, [b]anna.kielbus@pk.edu.pl

Keywords: Fuzzy Logic, Industrial Databases, Manufacturing Databases, Production Engineering

Abstract. The paper presents selected solutions for the implementation of fuzzy logic in industrial databases. Streaming data processing and classification is one of the most important problems in the Industry 4.0 era. The use of a database engine and appropriate design of the data model for the use of fuzzy logic is a response to expectations of the market. Examples of four types of fuzzy attributes are described. The universal fuzzy data model and its implementation are presented in the article for various internal industry information systems.

Introduction

A new approach to building industrial information systems ICT (Information and Communication Technologies) promotes the bottom-up method again instead of the top-down one [1, 2] in building computer applications. It is related to the needs of Industry 4.0 - entry of industrial systems into Big Data problems known in other areas of the market. There are also known problems of processing large amounts of data, where NoSQL class database systems are used as the mid-range layer. Streaming data processing systems may still require a complete or partial transfer of data to relational databases for further analysis. Modern industrial controllers, despite the evolution of the technology, though having the functionality of a computer (Open Controllers) do not have any mechanisms to handle Big Data.

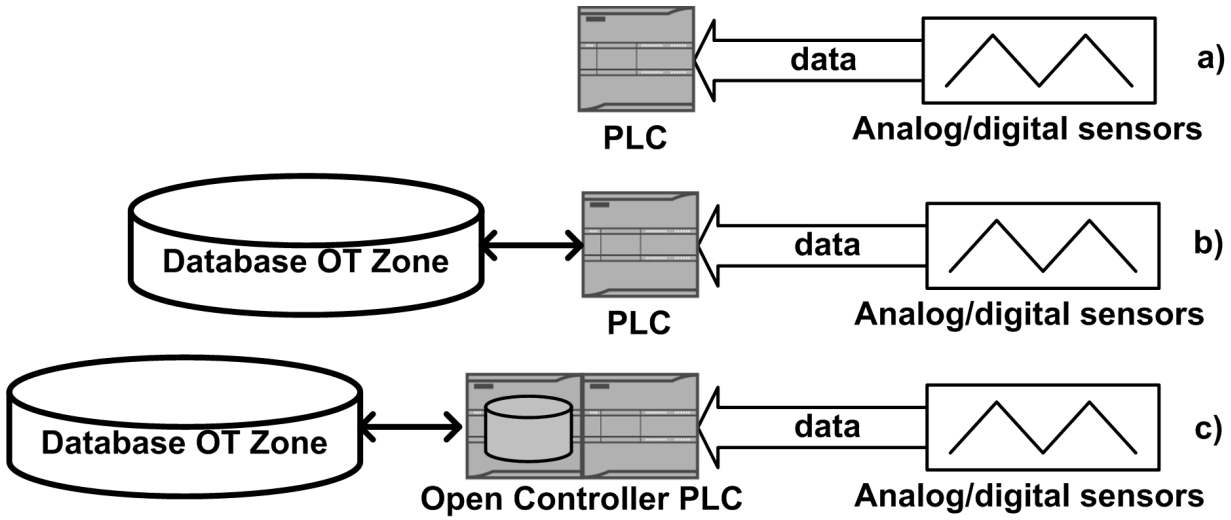

Fig. 1. Data flow from sensors to databases in OT zone: a) data processing only in PLC, b) using an external industry computer with database, c) using Open Controller.

Industry 3.0 era, as referred to in Fig. 1a, provided the ability to process selected data on PLC controllers in an OT (Operational Technologies) environment. In this approach, it was possible to respond only to analog or binary values. The use of fuzzy logic was limited only to the use of threshold structures with *if-elseif-else* type of programs without saving data (current data available in one cycle without history or with history limited by a small amount of internal memory). Limitations of data processing in PLCs have forced the use of industrial computers with the ability to store large amounts of data in standard relational databases or NoSQL databases in advanced applications. (Fig. 1b) [3]. With the emergence of Industry 4.0, industrial controllers equipped with a PC class computer with standard operating systems such as MS Windows or Linux as shown in Fig.1c have also been created. The low performance in such a solution is suitable for storing hundreds of thousands of rows of data in classic or NoSQL databases and using more sophisticated mechanisms for data processing and classification. In both cases, according to Fig. 1b and Fig. 1c, it is possible to use fuzzy logic for data classification and processing. It seems that due to the need to install the database server, it may be the best solution to use it as a fuzzy logic processing engine. The database management system (DBMS) uses a large pool of computer resources (CPU and RAM) but does not utilize them. This is the reason for using fuzzy logic in the active database environment in the OT zone.

In many industrial and research areas, there are increasingly imprecise or incomplete requirements, often from end users, that contractors have to deal with when collecting data during research and production. Hence, the presented approach, although strongly formalized, should be interesting for many potential users, including biotechnologists [4, 5], materials [6, 7] and biomaterials [8] engineers, technologists (in heat transfer [9], hydraulics [10] and surface machining [11-13]) or industrial and research data analysts [14-16].

Methods

The fuzzy set has been well described in the literature [17] as a set of pairs according to Eq. 1 as a set A over a universe of discourse X (a finite or infinite interval within which the fuzzy set can take a value). For proper implementation, the $\mu_A(x)$ is named the **membership degree of the element x** to the fuzzy set A, but for universal implementation in technical system it is convenient to assume definition of μ_A as **member functions** (characteristic functions) without giving an exhaustive list of all the pairs that make the fuzzy set. Also, the **fuzzy domain** as a set of possible values for an attribute was named [3].

$$A = \{\mu_A(x)/x: x \in X, \mu_A(x) \in [0,1] \in \Re\}. \quad (1)$$

However, the problem is a suitable description of the technical object. It is necessary to refer to internal and external standards (legal and industry) and to build the characteristics of the object using a set of coefficients as shown in the simplified Entity Relationship Diagram (ERD) in Fig. 2. The description of a technical object (marked on ERD as *Object of interest*) by linking to the dictionary of usable parameters (marked on ERD as *Factors*) implemented using a connecting entity *Object factors*. The *Factors* typically consist of numerical or boolean coefficients (e.g. TRUE/FALSE values used in electronics as the TTL signal levels). The use of fuzzy factors values has not yet been widely used despite such needs in many modern production management methodologies (e.g. to use in Total Production Management to the assessment of machine condition: good, acceptable, requires immediate intervention).

According to the classification of fuzzy attribute types [16], for the purposes of fuzzy database the following types have been defined:

Materials Research Proceedings **17** (2020) 100-107 https://doi.org/10.21741/9781644901038-15

Type 1 - Attributes containing precise, non-fuzzy data that allow for the extension of classic databases to use fuzzy queries. In addition to the numerical values, they may also have linguistic labels.

e.g. measuring instruments - permissible error: *value* (±)2 [μm], *label*: default
query: *select * from measuring_instruments where permissible_error < default.*

Type 2 - Attributes that are an extension of Type 1 allow for storing imprecise data. These factors admit crisp and fuzzy values in the same domain.

e.g. set *U:(5, 5.8, ~6, 5.5, ~5)*[V] where "~" is refer to "*approximately*".

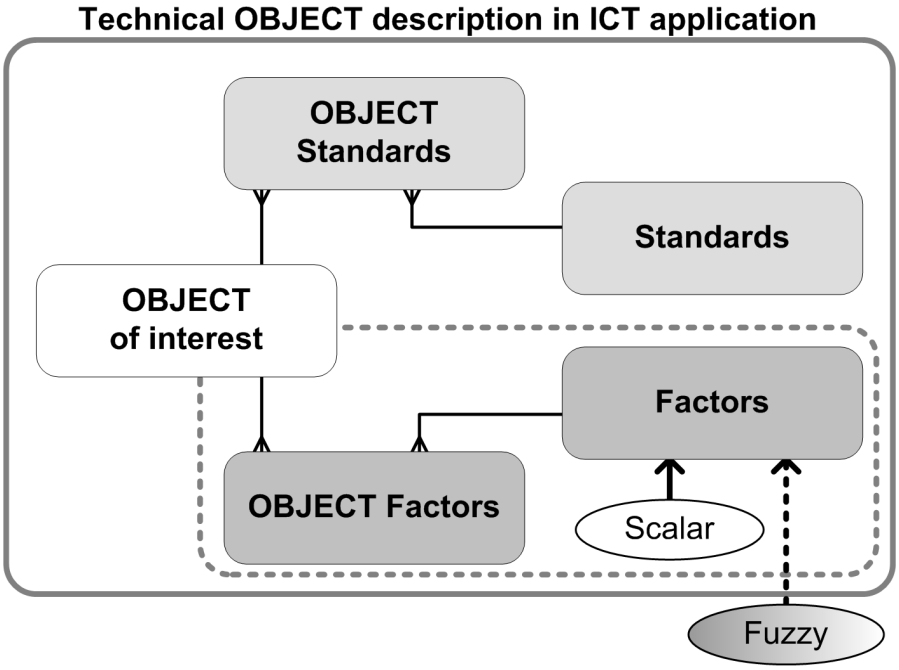

Fig. 2. Description of a technical object in standard ICT application without including fuzzy data.

Type 3 - Discrete attributes with defined linguistic labels with some degree of similarity. These types of attributes refer to a concave fuzzy set which complies with Eq. 2 with a similarity or proximity relationship. This type allows for examining the similarity of labels. Type 3 also allows for the examination of possibility distributions over the objective attribute domain according to Eq. 3.

$$\forall x,y \in X, \forall \lambda \in [0,1]: \mu_A(\lambda \bullet x + (1 - \lambda) \bullet y) \leq \min(\mu_A(x), \mu_A(y)). \quad (2)$$

$$A = \mu_1/x_1 + \mu_2/x_2 + ... + \mu_n/x_n , \quad (3)$$

where "+" is the aggregation operator, "/" is the operator of association of both values.

e.g. Selection of diameter of milling tool from tool storage to cutting a slot with specific dimension (100,100):

SlotsMillingTools.Diameter: (mikro2,mikro1,small2,small1,default,large1,large2),
SlotsMillingTools.Dim (100,100).Selection(0.7/small1, 1.0/default).

Type 4 - Attributes defined identically to Type 3, but without the need for a similarity relationship between labels or values within the fuzzy domain. This type is suitable for storing non-normalized values, which simplifies this type of attribute compared to Type 3. However, it can be assumed that they are different if we assume a degree of similarity of 0 for each pair of different labels, or attributes are equal if the degree is set to 1.

e.g. *temperature* in case 1 and its impact on tool machining accuracy,
and in case 2, *temperature* and its impact on tool life
(without taking into account the similarity between the two cases).

It is also possible to use fuzzy degrees [18] (associated or nonassociated) instead of (or additional to) fuzzy values of attributes with use either interval values or linguistic labels.

To build the test system, the x86 64-bit server with assigned 8 cores (from AMD Opteron 6344 series physical processor, 2.6 GHz clock) and 80 GB of RAM was used. The database server was located on a virtual machine in the Citrix Server (Xen) environment. As the DBMS the Oracle database 12c was installed and configured with 64 [GB] of System Global Area size. Instance of a fuzzy database was set to use Sort-merge join as primary to optimize all queries joins. The dedicated user schema (*fuzzy_user*) was used to implement the database in DDL SQL language.

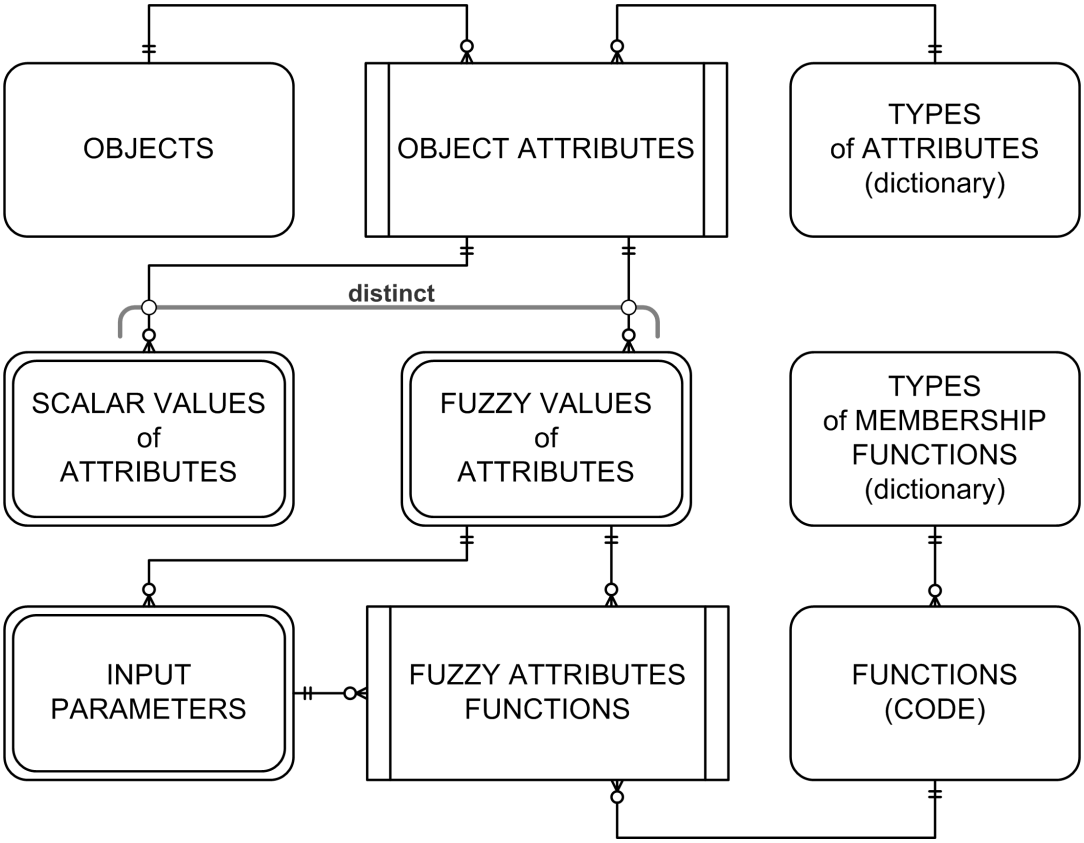

Fig. 3. A simplified ERD (Martin notation) to describe objects using fuzzy logic.

Results

Based on a study of requirements for a minimum description of technical objects using fuzzy logic, a database design was prepared using the Entity-Relationship Diagram (ERD) as shown in

Fig. 3. This is a simplified diagram, which does not have a complete set of tables and their relationships. In addition, an exclusive condition was introduced between two entities dependent on *OBJECT ATTRIBUTES*, where the attribute can be either scalar (*SCALAR VALUES of ...*) or fuzzy (*FUZZY VALUES of ...*) (for fuzzy type 1-4). In fact, type 1 of attributes can be stored as scalar but cannot be thresholded using linguistic descriptions.

For the proposed project (Fig. 3), an implementation was made in accordance with the database skeleton SK1:

SK1 :=
{ (TA; { id, name, description }), /* Types of fuzzy attributes */
(TF; { id, name, description }), /* Types of membership functions */
(ATFS; {id, label1, label2, degree}), /* Similarity functions between the labels */
(OBJ; { id, name, description }), /* Object of interest */
(OBJA; { id, obj_id, name, description }), /* Attributes of the object*/
(ATS; { id, obja_id, value }), /* Scalar Attributes*/
(ATF; { id, obja_id, ta_id, atfs_id, value }), /* Fuzzy Attributes*/
(ATFF; { id, atf_id, tf_id, input(a,b,c,d,margin){1-4}^{RegExp}, function_call})
/* Function for fuzzy attribute */ },

where for each table $T \in$ [TA, TF, ATFS, OBJ, OBJA, ATS, ATF, ATFF]

each *id* defined as a non-empty subset of the heading of T: ($\forall$t1,t2 $\in$ T: t1$\downarrow$A = t2$\downarrow$A => t1=t2),

and condition as above applied to TA(name), TF(name) as unique,

and OBJA(obj_id) $\in$ OBJ, ATS(obja_id) $\in$ OBJA, ATF(obja_id) $\in$ OBJA, ATF(ta_id) $\in$ TA,

ATF (atfs_id) $\in$ ATFS, ATFF(atf_id) $\in$ ATF, ATFF(tf_id) $\in$ TF.

For such a fuzzy database schema, it is possible to store any scalar and fuzzy data. Only the dictionary tables (*TA*, *TF*, *ATFS*) have predefined tuples with the possibility of extending with new definitions. Defined tuples for *TA* table:

TA := { {(id;1),(name;'Type 1')}, {(id;2),(name;'Type 2')},
{(id;3),(name;'Type 3')}, {(id;4),(name;'Type 4')} }.

The basic membership functions of triangular, trapezoidal and gaussian sets stored in *TF* dictionary were used for testing. For test cases, the input parameters of implemented programming functions in *ATFF* table were stored. The functions were implemented in the PL/SQL language in Oracle environment.

Example of definition tuples for *ATFS* table

(sets of Labels1 and Labels2={'VeryGood', 'Good', 'Moderate', 'Sufficient', 'Bad', 'VeryBad'}

with *degree* step 0,2):

ATFS := {{(id;1),(label1;'VeryGood'),(label2;'VeryGood'),(degree;1.0)}, ..., ...,

(4)

{(id; 21), (label1;'VeryBad'), (label2;'VeryBad'),(degree;1.0)} }, where *tuples* $\in$ { Label1 x Label2}.

For the exemplary object "*Machine electrical installation*" and the attribute "*Supply voltage*" it is possible to use both specific values (direct readings from the machine measuring system) of **type 1** and fuzzy values of **type 2** ("*about 255*" [V]) and **type 4** (set {"*too low*", "*normal*", "*too high*"} voltage). It should be noted that for such a case it is also possible to define the commonly

used "input voltage" attribute and to test the correlation between input and supply voltage. For the attribute defined as type 4, to labels (sets) "*too low*" and "*too high*" it is possible to use the trapezoidal membership function *T(x)*, while for the "*normal*" set it is possible to use the triangular function *A(x)*, where normal is defined as 230 ± 10% [V], where implementation is:

$\forall x_{input} \in \Re$, $\forall a,b,c,d \in \Re$:

T(x,a,b,c,d) = {if ($x \leq a$) => y:= 0.0; elsif ($x > a \wedge x \leq b$) => y := (x - a)/(b - a);
elsif ($x > b \wedge x \leq c$) => y := 1.0; ($x > c \wedge x \leq d$) => y := (d - x)/(d - c); else y:= 0.0; return y; }

A(x,a,b,c) = {if ($x \leq a$) => y := 0.0; elsif ($x > a \wedge x \leq c \wedge x$!= b) => y := (x - a)/(c - a);
elsif ($x > c \wedge x < b \wedge x$!= b) => y:= (b - x)/(b - c); elsif (x = b) => y := 1.0;
else y := 0.0; return y},
and e.g. "*too low*" set: {(a;0),(b;0) ,(c;207),(d;215)},
"*normal*" set: {(a;207),(b;230) ,(c;253) }, /* (d==NULL) */
"*too high*" set: {(a;245),(b;253) ,(c;999),(d;999)}.

The mechanisms of active databases [5,6] can, for this defined example, automatically use a trigger of the type "*AFTER INSERT for each row*" for each incoming scalar voltage value (stored in *ATS* table) and perform automatic classification with a fuzzy counterpart defined in *ATF* table. The whole analysis is in this case completely configurable and operated on the database server side. The result of analysis with linguistic label can be sent back to the PLC device and to the operator panel (HMI) and trigger the appropriate alert or alarm.

The example that will be used to describe **Type 3** attributes includes linguistic labels with some degree of similarity. The attribute "*Particulate matter PM10*" will be used to describe the "*Machine exhaust system*" (the object in *OBJ*). For this example, the *ATFS* table (according to Eq. 4 as above) defines a list of similarities between the exhaust gas quality indexes with the PM10 dust for two attributes (in the *OBJA* table) defined for two smokestack or two measuring points:

OBJ := {{(id;1),(name;'*Machine exhaust system*')}, }
OBJA := { {(id;1), (obj_id;1),(name;'PM10-measuring point 1')},
{(id; 2), (obj_id;1),(name;'PM10-measuring point 2')}},
ATF := {{(id;1), (obja_id;1),(ta_id;3),(value;y)}, ...,{(id;n), (obja_id;1),(ta_id;3),(value;u)}
{(id;n+1), (obja_id;2),(ta_id;3),(value;v)}, ...,{(id;m), (obja_id;2),(ta_id;3),(value;w)}},
where y,u,v,w ∈{'VeryGood', 'Good', 'Moderate', 'Sufficient', 'Bad', 'VeryBad'}, n,m ∈ ℕ*.

In practical application, the defined linguistic labels for *obja_id =1* and *obja_id =2* (as defined above) do not have to correspond to the same scalar values incoming from the measuring systems (e.g. where is setting the measuring points before and after the dedusting cyclone).

Discussion

The application of fuzzy set theory in classic databases does not have to be complicated. However, there are no standards to describe attributes and analyze their values. The study shows that the high performance database server used to operate with fuzzy sets does not introduce a visible delay in performing queries, despite the needs for joins in queries with additional 2 or 3 database tables. This is a result of the adopted DBMS platform (which is also a standard for many industrial applications such as MES, SCADA, ERP), which operates fast on small data sets of up to 1 million tuples. The proposed data model (ERD in Fig. 3) can be successfully applied

directly also on small servers installed in Open Controllers devices as shown in Fig. 1, and the mechanisms of active database systems can classify data in real time using the membership function and similarity matrices data with feedback to the PLC.

Various programming implementations of the proposed data model and functions can be used. What is important for the production implementation of fuzzy database is that variable number of input parameters for membership functions makes it possible to prepare a universal program code using overloaded functions (like in C++ style programming).

Attempts to adapt the proposed data model to issues related to ... [], ... [] and [] have also been undertaken. In all cases it was possible to build structures that fulfill the needs of data analysis requirements in the study area.

References

[1] D. Karpisz, A. Kielbus, Selected problems of designing modern industrial databases, MATEC Web of Conferences 183 (2018) art. 01017, https://doi.org/10.1051/matecconf/201818301017

[2] D. Karpisz, Design of manufacturing databases, Technical Transactions 113 (10) (2016) 73-77, doi: 10.4467/2353737XCT.16.123.5734

[3] D. Karpisz, A. Kielbus, M. Zembytska, Selected problems of industry databases and information infrastructure security, QPI 1 (1) (2019) 371-377, https://doi.org/10.2478/cqpi-2019-0050

[4] E. Skrzypczak-Pietraszek, J. Pietraszek, Phenolic acids in in vitro cultures of Exacum affine Balf. f. Acta Biol. Crac. Ser. Bot. 51 (2009) 62-62.

[5] E. Skrzypczak-Pietraszek, I. Kwiecien, A. Goldyn, J. Pietraszek, HPLC-DAD analysis of arbutin produced from hydroquinone in a biotransformation process in Origanum majorana L. shoot culture. Phytochemistry Letters 20 (2017) 443-448. https://doi.org/10.1016/j.phytol.2017.01.009

[6] T. Lipinski, The structure and mechanical properties of Al-7%SiMg alloy treated with a homogeneous modifier. Solid State Phenomena 163 (2010) 183-186. https://doi.org/10.4028/www.scientific.net/SSP.163.183

[7] T. Lipinski, Double modification of AlSi9Mg alloy with boron, titanium and strontium. Arch. Metall. Mater. 60 (2015) 2415-2419. https://doi.org/10.1515/amm-2015-0394

[8] D. Klimecka-Tatar, Electrochemical characteristics of titanium for dental implants in case of the electroless surface modification. Arch. Metall. Mater. 61 (2016) 923-26. https://doi.org/10.1515/amm-2016-0156

[9] L. Dabek, A. Kapjor, L.J. Orman, Boiling heat transfer augmentation on surfaces covered with phosphor bronze meshes. MATEC Web of Conf. 168 (2018) art. 07001. https://doi.org/10.1051/matecconf/201816807001

[10] M. Domagala, H. Momeni, J. Domagala-Fabis, G. Filo, M. Krawczyk, J. Rajda, Simulation of particle erosion in a hydraulic valve. Materials Research Proceedings 5 (2018) 17-24.

[11] D. Przestacki, M. Kuklinski, A. Bartkowska, Influence of laser heat treatment on microstructure and properties of surface layer of Waspaloy aimed for laser-assisted machining. Int. J. Adv. Manuf. Technol. 93 (2017) 3111-3123. https://doi.org/10.1007/s00170-017-0775-2

[12] S. Wojciechowski, D. Przestacki, T. Chwalczuk, The evaluation of surface integrity during machining of Inconel 718 with various laser assistance strategies. MATEC Web of Conf. 136 (2017) art. 01006. https://doi.org/10.1051/matecconf/201713601006

[13] Radek, N., Kurp, P., Pietraszek, J., Laser forming of steel tubes. Technical Transactions 116 (2019) 223-229. https://doi.org/10.4467/2353737XCT.19.015.10055

[14] J. Pietraszek, A. Gadek-Moszczak, The Smooth Bootstrap Approach to the Distribution of a Shape in the Ferritic Stainless Steel AISI 434L Powders. Solid State Phenomena 197 (2012) 162-167. https://doi.org/10.4028/www.scientific.net/SSP.197.162

[15] J. Pietraszek, A. Gadek-Moszczak, T. Torunski, Modeling of Errors Counting System for PCB Soldered in the Wave Soldering Technology. Advanced Materials Research 874 (2014) 139-143. https://doi.org/10.4028/www.scientific.net/AMR.874.139

[16] A. Pacana, K. Czerwinska, R. Dwornicka, Analysis of non-compliance for the cast of the industrial robot basis, METAL 2019 28th Int. Conf. on Metallurgy and Materials (2019), Ostrava, Tanger 644-650. https://doi.org/10.37904/metal.2019.869

[17] J. Galindo, A. Urrutia, M. Piattini, Representation of Fuzzy Knowledge in Relational Databases, in: Fuzzy databases: modeling, design and implementation, Idea Group Publishing, London, 2005, 145-151. https://doi.org/10.4018/978-1-59140-324-1.ch005

[18] J. Widom, S. Ceri, Active database systems: triggers and rules for advanced database processing, Burlington, Morgan Kaufmann, 1996.