Scientific Journals
of the Maritime University of Szczecin

Zeszyty Naukowe
Akademii Morskiej w Szczecinie

2016, 47 (119), 22–28
ISSN 1733-8670 (Printed)
ISSN 2392-0378 (Online)
DOI: 10.17402/144

Received: 29.04.2016
Accepted: 01.09.2016
Published: 20.09.2016

The use of text models in the formation of heuristics to solve tasks of diagnosing technical objects

Jerzy Korostil[1]✉, **Olga Korostil**[2]

[1] Maritime University of Szczecin
1–2 Wały Chrobrego St., 70-500 Szczecin, Poland, e-mail: j.korostil@am.szczecin.pl
[2] Ukrainian Printing Academy, Lviv, Ukraine, e-mail: korostilo_15@meta.ua
✉ corresponding author

Key words: heuristics, diagnostics, monitoring, malfunction, complex technical object

Abstract

This paper describes research related to the use of heuristics in diagnostic tasks of complex technical objects. To build heuristics, the use of text models for technical objects is proposed. Therefore, this paper examines output methods of heuristics from text models and their transformation into logical formulae suitable for use in diagnostic algorithms. Analysis has been carried out for tasks solved during diagnostics, and methods of using heuristics in certain tasks have been reviewed. It is proposed to use heuristics for decision making while implementing certain algorithm steps of monitoring tasks for diagnostic parameters that are solved during diagnostics.

Introduction

The task of diagnostics for complex technical objects (*STO*) is actually in the sphere of naval technologies, because examples of such objects are ships themselves or power units of large cargo ships. The task of diagnostics involves the detection of malfunctions which start taking place or are forecasted, their localisation and further elimination. We can accept that *STO*, in general can be described by some mathematical model, adequate for processes taking place in *STO*, so that we can solve the aforementioned tasks in the framework of the model.

In the framework of this work, the task of diagnostics is reviewed as a composite task, consisting of monitoring diagnostic and functional parameters, allowing the detection and localisation of possible malfunctions as well as the task of recognising and fixing the detected malfunctions. The monitoring task involves checking the parameter values in one or more sequences. As *STO* are large and complex, in the framework of one monitoring cycle it is impossible to check all possible parameters. Therefore, the task of forming a monitoring strategy arises, which would enable the detection of malfunctions at an early stage.

To form the appropriate monitoring strategy, in this work the use of heuristics is proposed, based on diagnosed process text descriptions, using formal means of non-monotone logic. This will allow the development of separate steps for the real-time monitoring strategy, to ensure the strategy is efficient in malfunction detection.

Basic concepts and tasks

There is a wide range of definitions and concepts of heuristics which represent features such as soft asks, and require them to be solved (Michalewicz & Fogel, 2006). To interpret the concept used in this paper unambiguously, let us introduce a definition of heuristics.

Definition 1. Heuristics ($\mathcal{E}r_i$) is a single rule or a system of rules for making a certain decision while solving a task. These rules cannot be deduced within the scope of functional tools or a mathematical model of task solving, but they represent features of a subject area where the task is solved.

Based on this definition, in order to build a heuristic system, further note das $\mathfrak{H}$, a certain extended description is needed for a subject area where the task which is solved can be interpreted. This extension must allow analysis and description of the situations which can occur in the object related to the task being solved, and it has to enable entering of changes, occurring within the object, into the corresponding object description. To implement such a description, it is proposed here that text models previously described (Korostil & Korostil, 2012) be adopted. In general, a text model is a text description of technological processes that take place in a complex technical object in the normalised form, in the native language of its user. Let us review the definition of such an object, which will be used in this paper.

Definition 2. A complex technical object (*STO*) is a technical object which does not allow, at least not without difficulties, to build a unified functioning model, which provides a sufficiently relevant and constructive object description, with regard to all possible malfunctions which may occur during its functioning.

Because a text model (TM_i) is a text description of the processes taking place in *STO*, this description is relevant enough to represent processes of functioning (PR_i) for the correspondent *STO*. To ensure the necessary amount of constructiveness of the TM_i, the model has to provide the following functional possibilities:

- the model TM_i has to be able to be modified in order to represent changes occurring in Pr_i and take place in *STO*;
- the model TM_i has to give the possibility to detect dependencies, not only between directly related parameters from Pr_i, but also indirectly related parameters, as well as to give the opportunity to detect dependencies that could be latent within *STO*;
- the model TM_i has to give the possibility to transform the processes Pr_i or their fragments into a system of logical models which will be written as $\mathcal{L}_i\,(STO)$.

It is planned, on the basis of the model TM_i $[Pr_i\,(STO)]$, to implement processes for deducing rules or conditions that describe certain heuristics $\mathcal{E}r_i$, which compose the system $\mathfrak{H}$. Thus, a system of rules for deducing heuristics from TM_i must be built, which will be written as $\mathfrak{I}$. For this system we have to determine input formulae as well as formulae obtained on the basis of deduction, which would allow for their interpretational extension, so that these formulae would transform into certain heuristics $\mathcal{E}r_i \in \mathfrak{H}$. Extension of $\mathcal{E}r_i \in \mathfrak{H}$ with interpretations $j(\mathcal{E}r_i)$ will allow us to obtain the necessary rules g_i, that represent heuristics $\mathcal{E}r_i$. Then, the following correlation can be written:

$$\mathcal{E}r_i = F\,[\mathfrak{I}_i\,(L_{i1},\ldots,L_{im})\&J(\mathfrak{H})] \qquad (1)$$

where $\mathcal{E}r_i$ are single heuristics in the diagnostic model of *STO*, $\mathfrak{I}_i$ is a system of rules for transforming $L_{i1}, \ldots, L_{im}$, built on the basis of transformations $\varphi_i\,(TM_i) \rightarrow L_{ij}$, F is a function that describes the interrelation between $\mathfrak{I}_i(L_{i1}, \ldots, L_{im})$ and $J(\mathfrak{H})$, $J(\mathfrak{H})$ is a system of interpretation for certain L_{ij}, from which they obtain the form of $\mathcal{E}r_i$. The system $\mathcal{L}_i$ is oriented towards using a finite area of interpretation for logical variables that is defined by a model $TM_i\,(STO)$. So, a logical system must be formed on the basis of formal logics, an example of which is non-monotonous logics (Thayse et al., 1990). One of the features of this logic, unlike classical logic, is that it gives rise to formulae which do not have to be valid on infinite sets of their interpretation (Mordechai, 2005).

Diagnostic tasks that are solved within the diagnostic model (*MD*) have to be able, within their own interpretation of diagnostic tasks, to use the correspondent $\mathcal{E}r_i$. The diagnostic tasks, within the scope of this paper, are the following (Korbic et al., 2002):

- task of monitoring (*Zm*) diagnostic parameters p_i^d and functional parameters p_i^f;
- task of detecting the appearance of a malfunction in *STO*, which will be written as (*Zz*;
- task of detecting a malfunction not yet manifested in the functioning process of *STO*, (*Zv*);
- task of recognizing the malfunction (*Zr*).

The tasks of withstanding detected malfunctions, or detected factors of their appearance, are not reviewed in this paper because they are related to the tasks solved within the security system of *STO*.

Each of the above-stated tasks *Zm*, *Zz*, *Zv* and *Zr* required the use of its own heuristics $\mathcal{E}r_i$. Heuristics for each task must allow interpretation, consistent with the interpretation of the corresponding task, described by the following relation:

$$J(\mathcal{E}r_i)\&J(Zk_i) \rightarrow J(Zk_i,\mathcal{E}r_i) \qquad (2)$$

where Zk_i is one task in a set $\{Zm_i, Zv_i, Zz_i, Zr_i$. In the case of a monitoring task, heuristics $\mathcal{E}r_i(Zm_i)$ takes part in determining the implementation strategy for the corresponding monitoring. In the case detecting the appearing malfunction, heuristics $\mathcal{E}r_i(Zz_i)$ takes part in identifying the fact of the malfunction occurring. In the case of detecting a malfunction not yet manifested in the functioning process of *STO*, heuristics $\mathcal{E}r_i(Zv_i)$ takes part in the analysis of diagnostic parameters read within the task Zm_i in order to detect changes related to the malfunction which occurred. In the case of recognizing the malfunction, heuristics $\mathcal{E}r_i(Zr_i)$ takes part in recognizing the type of malfunction, if it is unknown and only certain parameters characterizing the malfunction are known. Heuristics $\mathcal{E}r_i$ also takes part in determining the possible ways of affecting the parameters under change to in appropriate values. In this case, we cannot speak in detail about diagnostic parameters as about a separate class of functional parameters.

Solving of basic tasks

Using text models

A text model TM_i is a text description, in the normalised form of *STO* design and processes occurring within *STO*. Using TM_i allows the automation of processes for implementing different tasks which need to be solved during *STO* maintenance. Using TM_i, for solving tasks of *STO* maintenance is rational because of the following key features of *STO*:

- periodic modification of *STO*, which is necessary because of the requirement to update resource parameters for system nodes or components;
- inclusion of new elements in *STO* that represent new technological means which can be used during *STO* maintenance;
- carrying out repairs on *STO*, which can lead to some changes in the original set of system components, the design of *STO* and so on.

While performing these changes, the occurrence of critical situations during certain conditions of *STO* maintenance can eventuate. Because TM_i (*STO*) contains descriptions of all functioning conditions of *STO*, within TM_i it is possible to detect inappropriate values of parameters on the basis of conducting a semantic analysis of TM_i. Thanks to the introduction of semantic parameters σ^i it is possible not only to detect inconsistencies at quality level, but also to estimate their values in numbers (Korostil & Korostil, 2013).

A text model at quality level, or with a certain approximation, can be considered to be an analogue of technical documentation for the corresponding *STO*. To make it possible to perform analysis and modification of TM_i (*STO*), semantic vocabularies S_C and S_P are used. A vocabulary S_C contains data regarding all components and process fragments implemented in *STO*, and is a text description of the corresponding components which compose TM_i. A vocabulary S_P contains identifiers of diagnostic and other parameters which are used to analyse the current state of *STO*. A structure of these vocabularies is described by the following relations:

$$S_P = \left\{p_1^d, \ldots, p_n^d\right\}$$
$$p_i^d < \beta_{i1} * \cdots * \beta_{im} > I < P_1 \leq \delta_1, \ldots, P_m \leq \delta_m > \quad (3)$$

where β_{ij} is a word from a phrase oftext interpretation of the parameter, p_i^d is an identifier of a diagnostic parameter, P_i is a value of the parameter p_i, and δ_i is a boundary value of the parameter p_i^d. A certain value is introduced for p_i^d which, depending on the current value of the parameter p_i^d, will have different text interpretations $j(p_i^d,\delta_{ij})$. For example, within the threshold δ_{ij} the value of parameter p_i^d can have the following interpretation: [$\delta_{ij} \rightarrow j(p_i^d)$ = <excessive value>], another example of interpretation is [$\delta_{i+1} \rightarrow j(p_i^d)$ = <dangerous value>], or [$\delta_{ij+2} \rightarrow j(p_i^d)$ = <unacceptable value>]. From these examples it can be seen that within the vocabularies S_P and, respectively, in TM_i(*STO*), discrete methods of parameters evaluation, described by interpretative extensions, are used.

To implement processes for transferring information regarding changes of values of parameters occurring in *STO*, a system *SMA* is used that transfers the corresponding information in the model TM_i to modify the corresponding text fragments in TM_i. *STO* modification during maintenance is implemented in the following cases:

- in the case of implementing changes in the *STO* system, caused by the replacement of certain system components, conducted by maintenance staff;
- in the case when it is necessary to enter the current data analysed by the diagnostic system;
- in the case of changes occurring in *STO* as a result of external factors influencing the system, which can demand to add new fragments in the text description of the corresponding model TM_i.

In the first case, the maintenance staff enter the corresponding data into the system *SMA*, using different semantic vocabularies S_i, to describe the subject area W_i of an *STO* object. This generates an information stream IP_i and transfers it into the corresponding model TM_i. In the second case, the system *SMA* chooses the necessary text description from vocabularies of S_i type, on the basis of parameter identifiers p_i^d and the values of the corresponding parameters. After obtaining the information, *SMA* generates the corresponding IP_i for the model TM_i. The transfer of

IP_i into TM_i is implemented at the end of the current system work cycle or the work cycle of a monitoring model (*SMO*), because it may become necessary to make changes in the descriptions of certain parameters in TM_i. If the current parameter values changed at the amount which foresees the change of interpretation description used in TM_i, the system *SMA* activates the transfer of generated IP_i into the corresponding model TM_i. Synthesis of IP_i and TM_i is implemented corresponding to synthesis algorithms (Korostil, 2012). In the third case, when changes in *STO* are caused by an influence from external factors on the system, it can eventuate that there are no interpretation descriptions for these factors, which are necessary to describe the corresponding changes, in the S_i vocabulary system. In this case, system *SMA*, using text model systems (*STM*), S_i vocabulary systems and systems of text description output, implements the extension of the corresponding S_i with new components for which a text description has been deduced.

In the instance of detecting parameters with values not within the acceptable threshold, or which fail to meet the specified requirements, a diagnostic model (*MD*), solves the task of detecting, localizing and recognizing malfunctions to the extent of its capabilities. This is necessary because in these cases an effective reaction on detected unacceptable deviations is needed. Obviously, in these cases some processes may have to be blocked.

Using logical methods in tasks of heuristics generation

Because diagnostic tasks can be solved on the basis of using heuristics $\mathcal{E}r_i$, and the latter are a system of rules that are used while making decisions in a process of implementing a diagnostic algorithm, or for its preparation, it is reasonable to use methods of mathematical logic to generate them (Slupiecki, Halkowska & Pirog-Rzepecka, 1999).

Mathematical logic, in this case, is used to describe methods for implementing links between the system of text models $STM(TM_i, SMA, Sc_i)$, diagnostic models and other models used for the control and maintenance of *STO* in general. To implement these methods, the following tasks must be considered:

- tasks of transition from text fragments $tm_i \in TM_i$ to logical formulae describing tm_i with corresponding adequacy, which would provide effective usage of deduced heuristic rules;
- determining methods for choosing necessary $tm_i \in TM_i$;
- proving consistency or absence of contradictions $L(\mathcal{E}r_i)$ with the system of logical formulae describing the solutions of tasks implemented in $MD(STO)$;
- developing methods for the extension of logical formulae $L(\mathcal{E}r_i)$ interpretation to the interpretation of tasks solved in *MD*;
- if the result of *MD* functioning is obtained data for which there is no corresponding interpretation in *MD*, then it is necessary to solve the task of output for corresponding interpretative extensions and their transfer to the corresponding TM_i and *MD*.

Transformation of fragments $tm_i \in TM_i$ to a logical form is based on methods of using semantic parameters σ^i, as stated previously (Korostil, 2013). Considering this, to form $L_i(\mathcal{E}r_i)$, limitations in the interpretation of corresponding variables must be taken into account, and modal logic used. For this reason, logical formulae of deduced heuristics can be consistent within the limited boundaries of the definition of variables. This is implemented due to the use of modal logical operators. One example of logic that includes these operators is non-monotonous logic with defaults (Reiter, 1980). An example of one output rule which is called a default is $(\alpha \& M\beta) \rightarrow \gamma$, which has the following interpretation. If we believe in α and if β is true, then we take for granted that γ takes place, where M is a default operator. In this case, extensions of the output system is used with the corresponding interpretations, listed below.

1. A modal rule of output by observability, or introspection is: $p \rightarrow Lp$, and its interpretation lies in the following: «p has to be correct», provided p is correct.
2. A scheme of knowledge axiom: $Lp \rightarrow p$, which means «something that is known to be true».
3. A scheme of positive introspective axiom is described by the following relation: $Lp \rightarrow LLp$, where operator L means «known». Then, the mentioned scheme asserts that «if I know p, then I know that p is known to me».

Similarly, interpretation for a scheme of negative introspection axiom is introduced. Introspection, within the scope of this paper, is used to interpret limitation conditions for the definition range of the corresponding logical variables (McDermott, 1982).

Using non-monotonous logic allows us to prove that a certain statement is possible or can be deduced on the basis of using the corresponding logic. This logic indirectly allows us to accept the deduced formula as true, and using the corresponding operators L and M, gives a non-monotonous nature to the corresponding logic. Modal operators do not affect the interpretation of logical functions. According to a classic axiom system, modal axiom schemes and output rules, generating an output of formulae $L(\mathcal{E}r_i)$ does not lead to inconsistencies within the scope of the subject area of interpretation. This means that within the logical approximation, description of $\mathcal{E}r_i$ as $L(\mathcal{E}r_i)$ will not lead to inconsistencies with the logical interpretation of processes implemented in *MD*.

Let us review the ways of implementing solutions to the above-stated tasks, which occur while using logical formulae in diagnostic tasks. A choice of $tm_i \in TM_i$ is performed on the basis of text descriptions of the diagnostic system $TM_i(MD)$. This description represents processes implemented in *MD*. Thus, $TM_i(MD)$ contains descriptions of all known parameters which are related to the class of diagnostic parameters.

Within the $TM_i(STO)$, diagnostic parameters are also described, because they represent, on the level of functional parameters, the functioning processes of the corresponding fragments of *STO*. However, from the point of view of functioning process description $Pr_i(STO)$ they can be redundant. Their description in $TM_i(STO)$, unlike the description in $TM_i(MD)$, besides the identifiers, contains a description of interpretation of their current values. A simple example of $\mathcal{E}r_i$ could be heuristics of threshold analysis type, which is formally described by this relation:

$$\mathcal{E}r_i\left(p_i^d\right)=\left\{\left[p_i^d(\xi_i)<\Delta_i^d\delta(\xi_i)\right]\rightarrow\left[Pr_i(St)\setminus p_i^d(\xi_i)\right]\right\} \tag{4}$$

Let us examine how, based on the semantic analysis of TM_i, implicit functional links between different parameters are detected. Implicit functional links can only exist in instances when the corresponding parameters in TM_i are described in the different sentences, ψ_i and ψ_j. The system of semantic analysis *SMA* establishes an implicit link between the parameters $p_i{}^d$ and $p_j{}^d$, if in the corresponding $tm_i \in TM_i$ and $tm_j \in TM_i$ the values of these parameters are changed synchronously. Synchronisation in this case means that during two consequent cycles of analysis of TM_i, the corresponding parameters would be changed by the values described by a certain known functional dependency. In most cases, a linear function is chosen as this dependency.

If the generated $\mathcal{E}r_i$ has not led to, for example, the shortening of a monitoring cycle, then *SMA* system implements an extension $\mathcal{E}r_i$, that lies in continuing additional analysis of implicit links between $p_i{}^d$ and $p_j{}^d$. This would allow a shortened time of implementation vof the cycle of diagnostic process strategy of $St_i(MD)$.

Organising the process of diagnostics of a technical object

Let us examine the process of forming and implementing strategy for *STO* parameter monitoring. A monitoring strategy $St_i(SMO)$, implemented within the corresponding methods or the model *SMO*, is defined by the following parameters and characteristics of strategy St_i:

- time interval τ_i, when the monitoring process is implemented;
- number of parameters $p_{ij}{}^d$, chosen for control, where *i* is a monitoring cycle number, *j* is a corresponding parameter index, written as (λ_i);
- way store act, or reaction discipline of St_i on the value of parameter under check (Dr_i);
- controllability measure of the strategy (*Mu*).

Time interval τ_i is a parameter of St_i and its value can be defined for the current time point, including using the corresponding heuristics $\mathcal{E}r_i(\tau_i)$. Obviously, during the operation of a diagnostic model MD(*STO*) the value of τ_i can decrease or increase.

Parameters $p_{ij}{}^d$, controlled by the monitoring system *SMO*, can also be chosen on the basis of using certain heuristics $\mathcal{E}r_i(p_{ij}{}^d)$. This is caused by the fact that the check-up of all $p_{ij}{}^d$ each time *SMO* is activated is not reasonable, because certain $p_{ij}{}^d$ can remain unchanged during the current check-up period. A check-up period in this case is understood as a value $\Delta t_i = \tau_i + t_i$, where t_i is an interval duration between regular check-up cycles. Within the scope of processes $Pr_i(St_i)$, the way of reaction of St_i on changes of certain parameter values is determined. Because the value of t_i can vary, a task appears to determine the conditions of activation for the next monitoring cycle. Solutions for this tasks can be based on using data regarding the current level of system safety, taking heuristics $\mathcal{E}r_i(t_i)$ into consideration as well. A measure that determines the strategy controllability is defined by the following features:

- possibility to change, during implementation of the strategy St_i, the tactics determined when it was prepared, which means that the influence of external factors can change the sequence of parameter monitoring and one or another analysis depth of a certain parameter within the acceptable range;
- monitoring processes can change the trajectory of the search for data sources that characterise the

corresponding events, appearing during the system functioning and caused by external factors.

Using heuristics in diagnostic tasks is based on the analysis of conditions generated on the basis of the following rules:

- heuristics gives a binary result regarding the making of the corresponding decision;
- heuristics can use logical variables that are not present in the diagnostic model but characterise W_i;
- variable values are determined on the basis of setting up thresholds for parameter values identified by these variables;
- current values of diagnostic parameters, if they have gone from one value range to another one, are written in the vocabularies Sp_i in the data register mode;
- activation of recording the value of the next parameter is performed by each single source of corresponding information, based on their priorities.

An important element of *MD* is the task of localizing the malfunction. This task is closely related to recognizing the malfunction. In this case, recognizing the malfunction corresponds to detecting the reasons that caused it, which is the final goal of *MD*. In this case when an unacceptable change of a diagnostic parameter value is directly related to the reason of its deviation from normal values, *Zr* and *Zv* are the same. These cases are grouped in a separate malfunction class. In most cases, sensors of diagnostic parameters p_i^d are not directly related to the reasons of occurrence of the corresponding deviations. In a general case, the change of a parameter p_i^d into an unacceptable value range is caused by known reasons, unknown reasons, or a combination of known and unknown reasons.

In all cases, the localization task lies in determining one possible reason for the occurrence of the deviation. For this to occur, additional tools must be used, namely heuristics. Because heuristics are out of range of *MD*, it becomes possible to extend data regarding possible unknown malfunctions. In general, reasons for malfunction occurrence will be considered for certain events, occurring within the diagnostic object.

If we accept that an event y_i is the diagnostic parameter p_i^d going outside the acceptable range because of some reason, or $p_i^d > \delta\ (p_i^d)$, the corresponding event is described by a logical formula $L_i\ (p_i^d)$. This formula, together with the logical formula of the corresponding heuristics $L_i\ (\mathcal{E}r_i)$, or $L_i^{\mathcal{E}}$, must ensure the possibility of detecting reasons for the occurrence of $p_i^d > \delta\ (p_i^d)$. If a possible reason for the occurrence of $p_i^d > \delta\ (p_i^d)$ from $[L_i\ (p_i^d) \& L_i\ (\mathcal{E}r_i)]$ cannot be determined, it means that from the system $[L_i\ (p_i^d) \& L_i\ (TM_i)]$ a new heuristics formula $\mathcal{E}r_i$ has to be deduced, which would ensure the necessary output: $[L_i\ (p_i^d) \& L_i\ (TM_i)] \rightarrow L_i\ (\mathcal{E}r_i)$. Solving this task on a basic level is performed on the basis of implementing the following processes.

1. Modification of TM_i is implemented by the system of modification and analysis *SMA* using semantic vocabularies S_c, $TM_i \rightarrow TM_i^*$.
2. In the modified TM_i^* a semantic analysis of description for the corresponding *STO* components is implemented. On the basis of this analysis, indirect links between reasons of malfunction occurrence Ne_i and diagnostic parameters p_i^d are established, described by the relation:

$$\mathcal{F}\{\sigma_i^d(TM_i^*) \rightarrow \sigma_i^d[j(p_{i1}^d,\dots,p_{ik}^d) * j(p_{e1}^d,\dots,p_{eg}^d)]\}$$
$$\rightarrow \{\mathcal{E}r_i[P(p_{i1}^d,\dots,p_{ik}^d) * P(p_{e1}^d,\dots,p_{ek}^d)]\} = \{\mathcal{E}r_i^*(Ne_i)\} \tag{5}$$

3. $\{[L_i(p_{i1}^d,\dots,p_{ik}^d) * \mathcal{E}r_i^*(Ne_i)] \rightarrow L_i^*(\mathcal{E}r_i)\} \rightarrow \mathfrak{H}^R(L_i)$ is generated.
4. From the system $\mathfrak{H}(L_i)$ the formula $L_i^{Ne}\left(p_{i1}^d,\dots,p_{ik}^d\right)$ is deduced, from which the reason of Ne_i occurrence is determined.

Conclusions

The approach given in this paper is proposed to be used to organise a diagnostic system based on using text description models for technological processes of *STO* system functioning, and it is oriented towards use in a system complex enough with regards to its structure, as well as the number of functional processes implemented in such an object. This approach is especially effective for *STO* which have a big service life, which provides a great number of functioning cycles.

By using text models, it is possible to detect malfunctions which arise, because the data from the text models allows the detection of implicit relations between parameters, if they exist.

Since heuristics are formed not only on the basis of data possessed by the diagnostic model of a technical object, but also on the basis of data of the text model that describes a technical object in general, it becomes possible to extend the conditions which can be used by diagnostic algorithms. This extension ensures the correct decision is made by the diagnostic algorithm, if an ambiguous situation arises during its functioning.

References

1. Korbic, J., Kościelny, J.M., Kowalczyk, Z. & Cholewa, W. (Eds) (2002) *Diagnostyka procesów. Modele. Metody sztucznej inteligencji. Zastosowania*. Warszawa: WNT.
2. Korostil, O. (2012) Tools of descriptions of methods of text model synthesis with text information streams. Modelling and information technologies: collected articles. *G. Puchov IPME of National Academy of Sciences of Ukraine* 65. pp. 95–104.
3. Korostil, O. (2013) Analysis of parameters of text forms of representing the information. Collected articles. *G. Puchov IPME of National Academy of Sciences of Ukraine* 67. pp. 89–97.
4. Korostil, Y. & Korostil, O. (2012) Analysis and interpretation of text models. *Journal of KONBIN* N4(24). Safety and reliability systems. Warszawa: Wydawnictwo Instytutu Technologicznego Wojsk Lotniczych. pp. 51–56.
5. Korostil, Y. & Korostil, O. (2013) Methods and extension of abilities of text model semantic analysis. *Scientific Journals Maritime University of Szczecin* 35 (107). pp. 90–96.
6. McDermott, D. (1982) Non-monotonie logic: non-monotonie modal theories. *J.ACM.* 29, 1. pp. 34–57.
7. Michalewicz, Z. & Fogel, D.B. (2006) *Jak to rozwiązać, czyli nowoczesna heurystyka*. Warszawa: WNT.
8. Mordechai, B.A. (2005) *Logika matematyczna w informatyce*. Warszawa: WNT.
9. Reiter, R.A. (1980) Logic for default reasoning. *Artificial Intelligence* 13, 1–2. pp. 81–131.
10. Slupiecki, J., Halkowska, K. & Pirog-Rzepecka, K. (1999) *Logika matematyczna*. Warszawa: WN.
11. Thayse, A., Gribomont, P., Louis, G., Snyers, D., Wodon, P., Gochet, P., Gregoire, E., Sanchez, E. & Delsarte, P. (1990) *Logical approach to the artificial intelligence: from classical logics to logical programming*. Moscow: Mir.