9. Cocina, V., Colella, P., Pons, E., Tommasini, R., Palamara, F. (2016). Indirect contacts protection for multi-frequency currents ground faults. 2016 IEEE 16th International Conference on Environment and Electrical Engineering (EEEIC). doi: https://doi.org/10.1109/eeeic.2016.7555701
10. Czapp, S., Guzinski, J. (2018). Electric shock hazard in circuits with variable-speed drives. Bulletin of The Polish Academy of Sciences: Technical Sciences, 66 (3), 361–372. doi: http://doi.org/10.24425/123443
11. Czapp, S. (2010). The effect of PWM frequency on the effectiveness of protection against electric shock using residual current devices. 2010 International School on Nonsinusoidal Currents and Compensation. doi: https://doi.org/10.1109/isncc.2010.5524515
12. Syvokobylenko, V. F., Vasylets, S. V. (2017). Matematychne modeliuvannia perekhidnykh protsesiv v elektrotekhnichnykh kompleksakh shakhtnykh elektrychnykh merezh. Lutsk: Vezha-Druk, 272.
13. Pat. No. 135438 UA. Sposib kontroliu aktyvnoho oporu izoliatsiyi vidhaluzhennia elektrychnoi merezhi z napivprovidnykovym peretvoriuvachem chastoty (2019). MPK6 G01R 27/18, H02H 3/16. No. u201901598; declareted: 18.02.2019; published: 25.06.2019, Bul. No. 12.

UDC 389:14:621.317:354

DOI: 10.15587/1729-4061.2019.175811

QUALITY ASSESSMENT OF MEASUREMENT INSTRUMENT SOFTWARE WITH ANALYTIC HIERARCHY PROCESS

O. Velychko
Doctor of Technical Sciences, Professor, Director
Scientific and Production Institute of Electromagnetic Measurements
State Enterprise "All-Ukrainian State Scientific and Production Centre for Standardization, Metrology, Certification and Protection of Consumer",
(SE "Ukrmetrteststandard")
Metrolohychna str., 4, Kyiv, Ukraine, 03143

O. Hrabovskyi
PhD, Associate Professor, Director
Educational and Scientific Institute of Metrology, Automation, Intellectual Technologies and Electronics*

T. Gordiyenko
Doctor of Technical Sciences, Professor, Head of Department
Department of Standardization, Conformity Assessment and Educational Measurements*
E-mail: t_gord@hotmail.com
*Odessa State Academy of Technical Regulation and Quality
Kovalska str., 15, Odessa, Ukraine, 65020

Проведено порівняльний аналіз результатів оцінювання відповідності програмного забезпечення (ПЗ) для засобів вимірювальної техніки (ЗВТ). Для порівняльного оцінювання обрано вісім ПЗ ЗВТ з вмонтованим та універсальним комп'ютерами. Обрані ПЗ ЗВТ попередньо пройшли оцінювання за методиками та алгоритмами, які базуються на вимогах національних стандартів та документів міжнародних і регіональних організацій законодавчої метрології OIML та WELMEC. За результатами проведеного аналізу вимог настанови WELMEC щодо випробування ПЗ ЗВТ були виділені узагальнені та часткові показники для оцінювання якості ПЗ ЗВТ. Сформовано вирази для отримання чисельного значення кожного часткового показника за кожним узагальненим показником.

Для порівняльного оцінювання обрано метод аналітичної ієрархії (МАІ), оскільки він дозволяє порівняти і виконати кількісну оцінку альтернативних варіантів рішення. З метою релевантного порівняння під час оцінювання конкретного ПЗ ЗВТ були враховані всі порівнювані елементи. Останні були згруповані в узагальнені показники, кожен з яких оцінено окремо. Попарні порівняння й усі інші етапи оцінювання з використанням МАІ виконувались на основі узагальнених показників. Для попарного порівняння всіх кількісних та якісних показників з поданням результату зрівняння у кількісній формі, було використано шкалу Сааті. Експертним методом визначено коефіцієнти ваги кожного часткового показника.

Визначено основні показники для ПЗ ЗВТ з вмонтованим та універсальним комп'ютером, які мають найбільший вплив на результати оцінювання відповідності. Встановлено, що без представлення документації та ідентифікації ПЗ ЗВТ з вмонтованим та універсальним комп'ютерами неможливо розпочинати процедуру оцінювання відповідності згідно з вимогами. Показник перевірки запам'ятовуючих пристроїв та спеціальний показник перевірки ПЗ для певного ЗВТ є одними із вагомих показників. У той же час показники перевірки зчитування та перевірки рівнів розділення ПЗ практично не застосовні і ними можна знехтувати

Ключові слова: програмне забезпечення, засіб вимірювальної техніки, оцінювання якості, метод аналізу ієрархій

Received date 10.06.2019
Accepted date 25.07.2019
Published date 28.08.2019

1. Introduction

The requirements of the Measuring Instruments Directive 2014/32/EU (MID) [1] form the basis of the legislation of Ukraine on conformity assessment of measuring instruments (MI). According to the new version of the Law of Ukraine "On metrology and metrological activity" (came into force on 01.01.2016), MI intended for application in the field of legally regulated metrology must undergo a procedure of conformity assessment with the requirements of technical regulations (TR). This procedure is the process of proving that the essential requirements of the TR regarding the MI have been fulfilled. The conformity assessment procedure covers specialized MI software. Such MI must meet the essential requirements, including those related to the software, namely: suitability for use and protection against unauthorized interference.

The rules and procedures for testing MI software are established by the document [2] of the International Organization of Legal Metrology (OIML), as well as documents and recommendations of regional metrology organizations. Testing procedures for MI software are governed by the recommendation [3] of the Euro-Asian Cooperation of National Metrological Institutions (COOMET), document [4] and guidelines [5, 6] of the European Cooperation in Legal Metrology (WELMEC). At the national level, appropriate lists of national standards have been established, which, in particular, give the presumption of MI compliance with the essential requirements of the TR. The analysis of the state of the regulatory framework for testing MI software at the international, regional and national levels has been the subject of previous research [7–10].

National metrology institutes and conformity assessment bodies test MI software according to pre-established methods and algorithms. From January 2016 to the present, the number of MI software tests has been more than 500 in Ukraine and is constantly growing at a significant rate. The latter are based on the requirements of the national standard [11] with the additional use of the OIML D 31 [2] and WELMEC 7.2 requirements [5]. This approach contributes to the consideration of all the elements necessary to achieve the presumption of compliance of the software with the essential requirements of the TR. However, these approaches do not answer the question of the quality of software compliance assessment.

The urgency of the work is confirmed by the urgent need for conformity assessment of legally regulated MI in accordance with the requirements of national legislation, TR or European directives. National metrology institutes and conformity assessment bodies are interested in effective testing methods for MI software and in assessing the risks associated with the application of such MI software. Given this, the pressing issue is to validate the MI software test results.

2. Literature review and problem statement

The analysis of various aspects of MI software testing has been the subject of previous research of the authors [7–10].

In [7], the peculiarities of the regulatory support of MI software testing are investigated. The main stages of MI software testing and features in accordance with the requirements [2, 5, 6] are discussed in [8]. The main factors and algorithms for MI software testing in accordance with OIML and WELMEC requirements are considered, a universal algorithm for MI software testing is proposed in [9]. However, these studies did not analyze quality assessment indicators of MI software regarding the effect on the overall test results of MI software.

In [10], the main differences are identified and the necessary elements are established to achieve the presumption of software compliance with the essential TR requirements when conformity assessment of MI. However, the methods and algorithms described in [10] make it impossible to determine the validity of the results obtained from conformity assessment of MI software.

In [12], approaches to software quality requirements and software testing methods are compared, and different approaches to software quality assessment in different international standards and guidelines, in particular on issues related to the quality assessment of MI software, are determined.

In [13], issues related to the validation of MI software covered by the MID [1] are considered. A methodology is presented that can be extended not only to software on MI categories that fall under the MID, but also to most other MI categories. The paper [14] also discusses the validation of MI software based on risk classes for MI software and some possible testing methods.

In [15–18], a method for assessing the risks and current threats posed by MI software, including those integrated into open networks is presented. The method uses a structure and combines elements of specialized international standards and may be useful for conformity assessment bodies and industry.

However, [11–15] do not provide a comparative analysis of the importance of the impact of specific characteristics of MI software on the overall result of the software quality assessment. The requirements of the international document [2] and the possibility of software application for local MI are not taken into account in [16–18].

In [19], a system architecture is considered that could eliminate the risks of general-purpose operating systems. This is achieved both through the use of custom software and control of communication between major software components and the environment.

Thus, it can be concluded that the above studies did not analyze the influential indicators and results of quality assessment of MI software, and did not apply methods of their validation. Therefore, the analytic hierarchy process (AHP) was chosen to investigate such complex objects as MI software: the basic method [20, 21] and its modifications [22–24]. This method allows to structure a complex decision-making problem in the form of a hierarchy in a clear and rational way, to compare and quantify alternative solutions. Recently, AHP has been actively used in practice in various fields of activity. The AHP mathematical apparatus is described in detail in [25]. It is therefore necessary to conduct research and identify the most influential indicators that are analyzed in assessing the suitability of MI software, with a view to improving MI software testing methods.

3. The aim and objectives of the study

The aim of the study is to develop approaches to improve methods of testing and conformity assessment of special MI software at the national level.

To achieve this goal, the following objectives were set:

– to carry out a comparative analysis of the MI software testing results by all indicators using the chosen method;

– to determine the quality indicators of both built-in and universal computer MI software, which have the greatest impact on the overall assessment results.

4. Materials and methods of research for measuring instrument software quality assessment

The problem of comparative analysis of the MI software testing results using AHP is solved by means of three hierarchy levels:

– the first level of the hierarchy corresponds to the aim – to define the most preferred MI software;

– the second – contains criteria (indicators) to define the most preferred MI software;

– the third is the specific MI software that should be compared.

In general, the list of indicators should be such that the most comprehensive evaluation of each MI software is made. Each generalized indicator can be estimated by partial indicators contained in software documents or other available sources. For a relevant comparison, when assessing a particular MI software, it is necessary to consider all the elements compared. Therefore, they are grouped into generalized indicators, each of which is evaluated separately, and pairwise comparisons and all other stages of assessment, using the AHP, are performed on the basis of generalized indicators.

The main stages of comparative quality assessment of MI software based on AHP are as follows [25]:

1. Perform the following actions:

– to compile a list of M of MI software that will be compared;

– to carry out an analysis of available information about MI software (software description, user manual, etc.);

– to determine the list of indicators for comparative assessment, which should contain a sufficient number of indicators (not more than 9) in order to fully reflect all the essential features of MI software.

2. Determine the numerical values of the relative importance of the indicators a_{ij} (i, j=1, 2, ..., N), included in the pairwise comparison matrix (PCM) **A** of assessment indicators, established by the evaluation expert, directly during each specific MI software comparison.

3. Determine:

– Normalized eigenvector A_i of the matrix **A**;

– consistency of local priorities (C_d) included in the matrix **A**;

– consistency index of output data I_c;

– maximum eigenvalue $\lambda_{\max}$.

4. Determine:

– numerical values of the elements of the PCM $\mathbf{B}_k$ – b_{ij}^k (i, j=1, 2, ..., M), using available information from various sources to establish the array of $\mathbf{B}_k$ pairwise comparison matrices of MI software, according to each indicator;

– normalized eigenvectors for each constructed matrix $\mathbf{B}_k$.

5. Implement the consistency check of the local priorities that are included in the $\mathbf{B}_k$ matrices using the consistency relation C_d^k ($C_d^k \le 0{,}1$ – consistency condition). Determine the consistency index of the original data I_c^k and maximum eigenvalues $\lambda_{\max}^k$.

6. Determine the generalized priorities G_n for each of M of MI software compared by:

$$G_n = \sum_{i=1}^{N} B_i^0 \cdot B_n^i, \quad n = 1,\ 2,\ \ldots,\ M \tag{1}$$

with the further ranking of global priorities G_n for all MI software and definition of MI software that has the greatest advantage – the software with the maximum value G_n.

The structure of the model of quality assessment of MI software using AHP is shown in Fig. 1.

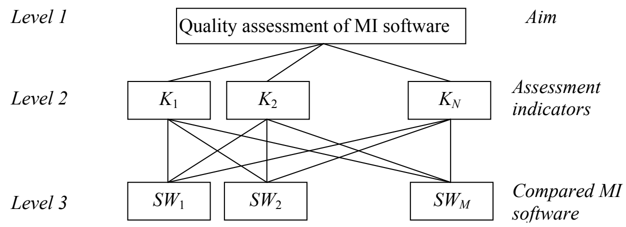

Fig. 1. Structure of the model of quality assessment of MI software using AHP

According to the results of the analysis of the requirements for testing the MI software of the WELMEC 7.2 guidelines [4], the following generalized indicators can be distinguished for assessing the quality of the MI software:

K_P – built-in computer software characteristic (P);

K_U – universal computer software characteristic (U);

K_L – test indicator of storage devices (L);

K_T – test indicator of data transfer devices (T);

K_S – reading test indicator (S);

K_D – test indicator of software separation levels (D);

K_I – specific test indicator of software for specific MI (I).

The structure of links between the requirements for quality assessment of MI software according to WELMEC 7.2 [4] is shown in Fig. 2.

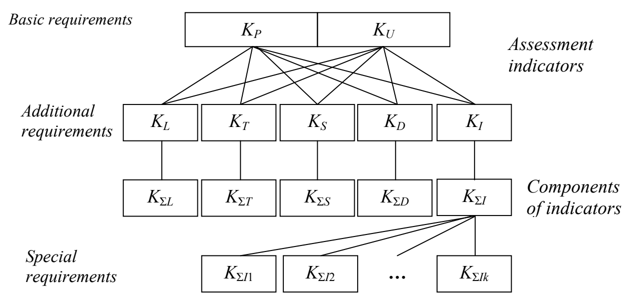

Fig. 2. Structure of links between requirements for quality assessment of MI software

For pairwise comparison of all generalized indicators with the help of AHP, including quantitative and qualitative, with the submission of the comparison result in quantitative form, one should use the Saati scale [20].

The list of partial indicators, which make up the generalized indicators K_P, K_U, K_L, K_T, K_S, K_D, K_I and expressions to obtain a numerical value for each generalized indicator are determined.

The numerical value of the built-in computer software characteristic K_P is determined by

$$K_P = \sum_{i=1}^{7} P_i w_i^P \big/ N_P, \tag{2}$$

where N_P – the total number of estimated MI software indicators (N_P=7);

P_i – constituent estimates of the generalized indicator K_P (numerical characteristics of partial indicators with certain weight coefficients w_i^P):

P_1 – software user documentation;
P_2 – MI software identification;
P_3 – impact through user interfaces;
P_4 – impact through transmission interfaces;
P_5 – modification protection;
P_6 – software modification protection;
P_7 – parameter protection.

The numerical value of the universal computer MI software characteristic K_U is determined by

$$K_U = \sum_{i=1}^{9} U_i w_i^U / N_U, \tag{3}$$

where N_U – the total number of estimated MI software indicators (N_U=9);

U_i – constituent estimates of the generalized indicator K_U (numerical characteristics of partial indicators with certain weight coefficients w_i^U):

U_1 – documentation;
U_2 – software identification;
U_3 – influence through user interfaces;
U_4 – impact through transmission interfaces;
U_5 – modification protection;
U_6 – software modification protection;
U_7 – parameter protection;
U_8 – software authentication and results transfer;
U_9 – impact of other software.

The numeric value of the test indicator of storage devices K_L is determined by

$$K_L = \sum_{i=1}^{8} L_i w_i^L / N_L, \tag{4}$$

where N_L – the total number of estimated MI software indicators (N_L=8);

L_i – constituent estimates of the generalized indicator K_L (numerical characteristics of partial indicators with certain weight coefficients w_i^L):

L_1 – completeness of stored data;
L_2 – protection against accidental or conscious modification;
L_3 – data integrity;
L_4 – authenticity of stored data;
L_5 – conference keys;
L_6 – recovery of stored data;
L_7 – automatic storage;
L_8 – storage capacity and sequence.

The numeric value of the test indicator of data transfer devices K_T is determined by

$$K_T = \sum_{i=1}^{8} T_i w_i^T / N_T, \tag{5}$$

where N_T – the total number of estimated MI software indicators (N_T=8);

T_i – constituent estimates of the generalized indicator K_T (numerical characteristics of partial indicators with certain weight coefficients w_i^T):

T_1 – completeness of the transmitted data;
T_2 – protection against accidental or conscious modification;
T_3 – data integrity;
T_4 – authenticity of transmitted data;
T_5 – conference keys;
T_6 – spoiled data processing;
T_7 – transmission delay;
T_8 – suitability of transmission services.

The numeric value of the reading test indicator K_S is determined by

$$K_S = \sum_{i=1}^{3} S_i w_i^S / N_S, \tag{6}$$

where N_S – the total number of estimated MI software indicators (N_S=3);

S_i – constituent estimates of the generalized indicator K_S (numerical characteristics of partial indicators with certain weight coefficients w_i^S):

S_1 – implementation of software separation;
S_2 – fixed indication;
S_3 – software interface protection.

The numeric value of the test indicator of software separation levels K_D is determined by

$$K_D = \sum_{i=1}^{4} D_i w_i^D / N_D, \tag{7}$$

where N_D – the total number of estimated MI software indicators (N_D=4);

D_i – constituent estimates of the generalized indicator K_D (numerical characteristics of partial indicators with certain weight coefficients w_i^D):

D_1 – reading mechanism;
D_2 – authenticity of read software;
D_3 – integrity of read software;
D_4 – reading traceability of legally significant software.

The numeric value of the specific test indicator of software for specific MI K_I is determined by

$$K_I = \sum_{i=1}^{6} I_i w_i^I / N_I, \tag{8}$$

where N_I – the total number of estimated MI software indicators (N_I=6);

I_i – constituent estimates of the generalized indicator K_I (numerical characteristics of partial indicators with certain weight coefficients w_i^I):

I_1 – failure recovery;
I_2 – availability of duplicate equipment;
I_3 – suitability indication;
I_4 – preventing the reset of accumulation registers;
I_5 – dynamic behavior;
I_6 – protection of parameters specific to electricity meters.

Specific requirements have been set for software of certain MI groups, in particular for: electricity meters, water meters, heat meters and more.

5. Results of comparative software quality assessment

The numerical values of the PCM elements of indicators **A** with the normalized eigenvector A_i for the selected indicators for comparative quality assessment of built-in computer (P) or universal computer (U) MI software are shown in Table 1. If the quantitative relationships between the indicators do not satisfy the expert performing a certain comparative quality assessment of MI software, then they can be modified as necessary.

Table 1

Numerical values of the elements of the PCM of indicators for comparative quality assessment of MI software

	K_P	K_L	K_T	K_S	K_D	K_I	K_U	A_i
K_P	1	1	1	3	3	2	1	0.194
K_L	1	1	2	3	3	2	1	0.214
K_T	1	0.5	1	3	3	2	1	0.175
K_S	0.33	0.33	0.33	1	2	2	0.33	0.083
K_D	0.33	0.33	0.33	0.5	1	1	0.33	0.062
K_I	0.5	0.5	0.5	0.5	1	1	0.5	0.078
K_U	1	1	1	3	3	2	1	0.194

In the case of the data in Table 1, the consistency condition for the consistency ratio (C_d≤0.1) is satisfied (C_d=0.02). The consistency index is I_c=0.028, and the largest eigenvalue of the vector is λ_{max}=7.17.

The weight coefficients w_i for the selected MI software quality assessment indicators are defined in Table 2.

At the beginning of the assessment, it is necessary to determine the basic configuration of software: with a built-in computer *P* or with a universal computer *U*. Then, a complete set of requirements relating to the corresponding basic configuration must be used.

Eight software were selected for comparative quality assessment:

– six built-in computer MI software: for electricity meters (SW1, SW2, SW6), gas distribution columns (SW3), gas chromatograph (SW4), heat meter (SW5);

– two universal computer MI software: for cardiac monitoring (SW7) and liquid chromatograph (SW8).

The numerical values of the MI software quality indicators were converted into the numbers needed for assessment by the AHP in the range from 1 to 9 using the Saati scale.

The results of assessment of MI software quality indicators in accordance with the presented methodology are shown in Table 3. Specialized software "AHP Competence 1.2" (Ukraine), which implements AHP, was used for the necessary calculations.

The comparison of the global priorities G_n of the MI software under consideration with the ranking by their reduction (AHP 1.2 Competence, Ukraine) is shown in Fig. 3.

Table 2

Weight coefficients for the selected MI software quality assessment indicators

i	w_i^P	w_i^U	w_i^L	w_i^T	w_i^S	w_i^D	w_i^I
1	2	2	3	3	2	3	4
2	5	5	5	5	3	3	3
3	4	4	4	4	3	4	3
4	2	2	3	3	–	3	3
5	4	4	2	2	–	–	2
6	3	3	3	3	–	–	3
7	3	3	2	2	–	–	2
8	–	2	3	2	–	–	–
9	–	2	–	–	–	–	–
10	–	–	–	–	–	–	–

Table 3

Results of assessment of MI software quality indicators

Indicators	Vectors for MI software								λ_{max}^k	I_c^k	C_d^k
	SW1	SW2	SW3	SW4	SW5	SW6	SW7	SW8			
K_P	0.143	0.143	0.143	0.143	0.143	0.143	0.071	0.071	8.000	0.0000	0.0000
K_U	0.143	0.143	0.143	0.143	0.143	0.143	0.071	0.071	7.994	0.0008	0.0005
K_L	0.127	0.071	0.044	0.127	0.127	0.078	0.213	0.213	8.073	0.0105	0.0072
K_T	0.205	0.057	0.106	0.205	0.106	0.106	0.106	0.106	8.099	0.0142	0.0098
K_I	0.127	0.127	0.072	0.032	0.127	0.072	0.222	0.222	7.994	0.0008	0.0005
K_S	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125	8.000	0.0000	0.0000
K_D	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125	7.985	0.0020	0.0014
G_n	0.147	0.109	0.107	0.139	0.129	0.114	0.127	0.127	–	–	–

Fig. 3. Results of assessment of MI software quality indicators

The analysis of the obtained results shows the advantage of MI software in the following sequence: SW1>SW4>>SW5>SW7>SW8>SW6>SW2>SW3. It should be noted that the universal computer MI software has an average level of quality. The built-in computer MI software for the SW1 and SW2 meters has significantly different levels of quality.

6. Discussion of the results of comparative software quality assessment

The consistency relation C_d^k for all MI software with a large margin meets established requirements $\left(C_d^k \leq 0{,}1\right)$. This means that the results obtained are acceptable and that further analysis is possible. Maximum eigenvalues $\lambda_{\max}^k$ vary within a wide range from 7.985 (for the K_D indicator) to 8.099 (for the K_T indicator) for all quality indicators of MI software.

It is important to analyze the values of the MI software quality indicators to identify the weight of their impact on the overall software quality assessment. Comparative analysis of the priority vectors of the MI software quality indicators shown in Table 3 showed the following.

Without the submission of documentation on software with MI with built-in and universal computers and its identification, it is not possible to start the quality assessment process in accordance with the established requirements. That is, built-in computer (K_P) and universal computer (K_U) software indicators are important by default, although their maximum eigenvalues are average (8.000 and 7.994, respectively).

The test indicator of storage devices (K_L) and the test indicator of data transfer devices (K_T) are some of the important quality indicators (maximum eigenvalues of 8.073 and 8.099, respectively).

At the same time, reading test indicator (K_S), specific test indicator of software for specific MI (K_I), and test indicator of software separation levels (K_D) have a small contribution to the overall quality indicator (maximum eigenvalues of 8.000, 7.994 and 7.985, respectively).

The diagram of the weight of MI software quality indicators, which allows to present clearly the obtained results, is shown in Fig. 4.

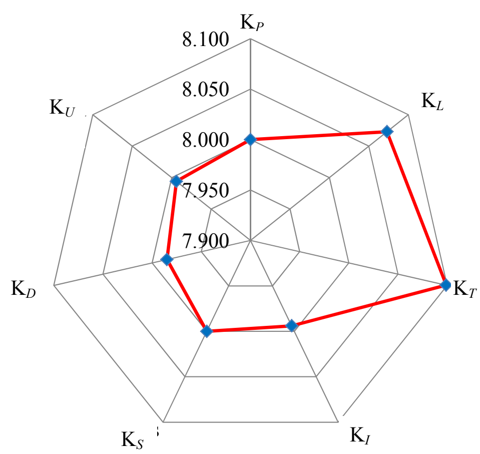

Fig. 4. Diagram of the weight of MI software quality indicators

It should be noted that the reading test indicator (K_S) and the test indicator of data transfer devices (K_T) are practically impossible to apply to software with a built-in computer that is used in simple MI (such as simple electricity meters). This is because such MI lack reading and data transfer devices. At the same time, these MI software quality indicators are important for software with a versatile computer used in complex MI, such as cardio monitors and liquid chromatographs.

Thus, the main quality indicators of MI software with a built-in and versatile computer that have the greatest impact on the results of conformity assessment, are identified.

The results of the study can later be used to modify and improve the algorithm and methodology for conformity assessment of MI software.

7. Conclusions

1. A comparative analysis of the testing results of software for MI with built-in and universal computers using AHP is carried out. Based on the analysis of the requirements of the

WELMEC 7.2 guidelines for MI software testing, generalized and partial indicators are identified to assess the quality of MI software. Expressions to obtain the numerical value of each partial indicator by each generalized indicator are generated. The results of the comparison showed the AHP suitability for pairwise comparisons of all quantitative and qualitative indicators of quality assessment of MI software.

2. The quality indicators of MI software with built-in and universal computer, which have the greatest impact on quality assessment results are determined. It is found that without the submission of documentation and identification of MI software with built-in and universal computer, it is not possible to start the conformity assessment procedure as required. That is, the quality indicators of built-in computer (K_P) and universal computer (K_U) software are important by default. Also, the test indicator of storage devices (K_L) and the specific test indicator of software for specific MI (K_I) are some of the important indicators. It is determined that the reading test indicator (K_S) and the test indicator of software separation levels (K_D) are practically inapplicable and can be neglected.

References

1. Directive 2014/32/EU of the European Parliament and of the Council of 26 February 2014 on the harmonisation of the laws of the Member States relating to the making available on the market of measuring instruments (recast) // Official Journal of the European Union, 2004. L96/149. Available at: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32014L0032
2. OIML D 31:2008. General Requirements for Software Controlled Measuring Instruments (2008). OIML. Paris, 53.
3. COOMET R/LM/10:2004. COOMET Recommendation: Software for Measuring Instruments: General Technical Specifications (2004). COOMET, 10.
4. WELMEC 7.1. Informative Document: Development of Software Requirements. Available at: http://www.welmec.org/fileadmin/user_files/publications/WG_07/7-1_FRPO.pdf
5. WELMEC 7.2. Software Guide (Measuring Instruments Directive 2014/32/EU1). Available at: http://www.welmec.org/fileadmin/user_files/publications/WG_07/Guide_7.2_2015__Software.pdf
6. WELMEC 2.3. Guide for Examining Software (Non-automatic Weighing Instruments). Available at: http://www.welmec.org/fileadmin/user_files/publications/2-3.pdf
7. Velichko, O. N. (2007). Normative base for certification of measurement provision software. Measurement Techniques, 50 (4), 364–371. doi: https://doi.org/10.1007/s11018-007-0076-5
8. Velichko, O. N. (2009). Basic tests, stages, and features in monitoring measuring instrument software. Measurement Techniques, 52 (6), 566–571. doi: https://doi.org/10.1007/s11018-009-9308-1
9. Velychko, O., Gordiyenko, T., Hrabovskyi, O. (2018). Testing of measurement instrument software on the national level. Eastern-European Journal of Enterprise Technologies, 2 (9 (92)), 13–20. doi: https://doi.org/10.15587/1729-4061.2018.125994
10. Velychko, O., Gaman, V., Gordiyenko, T., Hrabovskyi, O. (2019). Testing of measurement instrument software with the purpose of conformity assessment. Eastern-European Journal of Enterprise Technologies, 1 (9 (97)), 19–26. doi: https://doi.org/10.15587/1729-4061.2019.154352
11. DSTU 7363:2013. Prohramne zabezpechennia zasobiv vymiriuvalnoi tekhniky. Zahalni tekhnichni vymohy (2013). Kyiv: Minekonomrozvytku Ukrainy, 11.
12. Tasić, T., Grottker, U. (2006). An overview of guidance documents for software in metrological applications. Computer Standards & Interfaces, 28 (3), 256–269. doi: https://doi.org/10.1016/j.csi.2005.07.012
13. Richter, D., Grottker, U., Talebi, D., Schwartz, R. (2006). The new European software guide for legal metrology: Basic principles. Computer Standards & Interfaces, 28 (3), 270–276. doi: https://doi.org/10.1016/j.csi.2005.07.005
14. Jacobson, J. (2006). Validation of software in measuring instruments. Computer Standards & Interfaces, 28 (3), 277–285. doi: https://doi.org/10.1016/j.csi.2005.07.006
15. Esche, M., Thiel, F. (2015). Software Risk Assessment for Measuring Instruments in Legal Metrology. Proceedings of the 2015 Federated Conference on Computer Science and Information Systems. doi: https://doi.org/10.15439/2015f127
16. Sadiq, M., Md. Khalid Imam Rahmani, Mohd. Wazih Ahmad, & Jung, S. (2010). Software risk assessment and evaluation process (SRAEP) using model based approach. 2010 International Conference on Networking and Information Technology. doi: https://doi.org/10.1109/icnit.2010.5508535
17. Peters, D., Grottker, U., Thiel, F., Peter, M., Seifert, J.-P. (2014). Achieving Software Security for Measuring Instruments under Legal Control. Position Papers of the 2014 Federated Conference on Computer Science and Information Systems. doi: https://doi.org/10.15439/2014f460
18. Boccardo, D. R., dos Santos, L. C. G., da Costa Carmo, L. F. R., Dezan, M. H., Machado, R. C. S., de Aguiar Portugal, S. (2010). Software evaluation of smart meters within a Legal Metrology perspective: A Brazilian case. 2010 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT Europe). doi: https://doi.org/10.1109/isgteurope.2010.5638881
19. Peters, D., Peter, M., Seifert, J.-P., Thiel, F. (2015). A Secure System Architecture for Measuring Instruments in Legal Metrology. Computers, 4 (2), 61–86. doi: https://doi.org/10.3390/computers4020061
20. Saaty, T. L. (1992). The Hierarchon: A Dictionary of Hierarchies. Pittsburgh, Pennsylvania: RWS Publications, 496.
21. Saati, T. L. (2008). Prinyatie resheniy pri zavisimostyah i obratnyh svyazyah: Analiticheskie seti. Moscow: Izd-vo LKI, 360.
22. Drake, P. R. (1998). Using the Analytic Hierarchy Process in Engineering Education. International Journal of Engineering Education, 14 (3), 191–196.

23. Roy, R. (Ed.) (2004). Strategic Decision Making: Applying the Analytic Hierarchy Process. Springer-Verlag, 170. doi: https://doi.org/10.1007/b97668
24. Longaray, A. A., Gois, J. de D. R., Munhoz, P. R. da S. (2015). Proposal for using AHP Method to Evaluate the Quality of Services Provided by Outsourced Companies. Procedia Computer Science, 55, 715–724. doi: https://doi.org/10.1016/j.procs.2015.07.083
25. Velychko, O., Gordiyenko, T., Kolomiets, L. (2017). A comparative analysis of the assessment results of the competence of technical experts by different methods. Eastern-European Journal of Enterprise Technologies, 4 (3 (88)), 4–10. doi: https://doi.org/10.15587/1729-4061.2017.106825

UDC 621.01;61;616-02;331

DOI: 10.15587/1729-4061.2019.176174

DEVELOPMENT OF THE METHOD OF STRUCTURAL AND PARAMETRIC SYNTHESIS OF THE QUANTON DIAGNOSTIC AND HEALTH COMPLEX

I. Ogorodnyk
Engineer-Inventor, Head of Department
Department of Innovative Medical Technology*
Email: oigor67@gmail.com

O. Vysotska
Doctor of Technical Sciences, Professor
Department of Radio-Electronic and Biomedical Computer-Aided Means and Technologies
National Aerospace University Kharkiv Aviation Institute
Chkalova str., 17, Kharkiv, Ukraine, 61070

M. Ternyuk
Doctor of Technical Sciences, Professor, President*

H. Bilovol
PhD, Associate Professor
Department of Heat Engineering, Heat Engines and Energy Management
Ukrainian State University of Railway Transport
Feierbakha sq., 7, Kharkiv, Ukraine, 61050
*International Academy of Science and Innovation Technologies
Koltsova blvd., 14-E, Kyiv, Ukraine, 03148

Запропоновано метод синтезу, що базується на застосуванні інформаційних інваріантів, і виконано структурно-параметричний синтез діагностично-оздоровчого комплексу «Quanton». Проведена структурно-параметрична оптимізація комплексу по критерію продуктивності. Як інформаційні інваріанти використані повні (у межах прийнятої класифікації) множини шляхів отримання функціональних властивостей комплексу, фазових циклів життєвого циклу, структур технічних підсистем і способів управляння рівнями технізації, продуктивності та енергоефективності процесів. Множини шляхів отримання функціональних властивостей комплексу та фазових циклів життєвого циклу сформовані шляхом поелементного ускладнення відповідних атрибутів. Множина структур технічних підсистем, що відповідають певним рівням технізації функцій, визначена на основі періодичної системи технічних елементів. Повні множини можливих структурних рішень по способам управління продуктивністю та енергоефективністю процесів, отримані топологічним добутком множин видів об'єктів на види прийомів забезпечення потрібних властивостей чи якостей об'єктів. Для кожного структурно відмінного варіанту застосована типова процедура параметризації об'єктів та система залежностей задачі параметричної оптимізації дедуктивного типу. Система залежностей є конкретизованим випадком параметричних інформаційних інваріантів. Конкретизацію залежностей здійснено з використанням інформації про потрібні вихідні дані та цільові перетворення, що виникають у комплексі «Quanton», при взаємодії підсистем. Алгоритм пошуку гранично ефективного рішення є покроковим. Цим алгоритмом передбачається покрокове визначення оптимальних по продуктивності процесів значень параметрів в межах блокуючих контурів та послідуюче їх покращення по енергоефективності та якості. Внаслідок використання повних множин структур процесів, елементів та дискретно-континуальної процедури пошуку оптимального рішення досягнута комплексність оптимізації технічної інновації

Ключові слова: параметричний синтез, комплекс «Quanton», неінвазивна діагностика, спектральний метод, терапія високочастотна

Received date 03.06.2019
Accepted date 05.08.2019
Published date 28.08.2019

1. Introduction

The Quanton diagnostic and health complex is a technical innovation designed to perform a triple function: non-invasive diagnostics of people, gaining information about the characteristics of the desired health normalizing effect on them and implementation of this effect. Non-invasive diagnostics combines spectral and binary methods. The spectral method with a certain level of reliability allows identifying organs that have deviations from the standards. The binary