Marine Electric Generating Plants Control Systems Software Functional Testing

MAHMOUD MOHAMMAD AL-SUOD[1], A. USHKARENKO[2], O. DOROGAN[3].
[1]Department of Electrical Power Engineering and Mechatronics, [2, 3]Department of Electrical and Electronic Engineering
[1]TafilaTechnical University, [2,3]Admiral Makarov National University of Shipbuilding
Tafila -11660, Nikolaev
[1]JORDAN, [2,3]UKRAINE
m.alsoud@ttu.edu.jo, maestrotees@gmail.com, dorogan.olga.n@gmail.com

Abstract: - The characteristics of the quality of the specialized software for a ship power system control and monitoring are discussed. The methodology of the evaluation of functionality, usability and reliability of specialized software for monitoring and managing ship's power plants are presented. The availability of such a methodology makes it possible to obtain a composition of software quality attributes and quantitative indicators, indicate the direction of priority changes and assess the results of their implementation, efficiency and reliability of specialized software.

Key-Words: - functional testing, integration testing, software quality, ship power system.

1 Introduction

Modern control systems of marine electric power systems are distributed and conditionally divided into three hierarchical levels. Hardware and software of the upper level coordinates the operation of the entire system and contains the data collection server and the operator's automated workstation. Hardware and software of the lower level is a set of territorially distributed automation hardware. Communication infrastructure is an intermediate layer element.

The quality is one of the key issues in the top level software design for the automated control system of the marine electric generating plant. The software quality model SQuaRE is recognized to be generally accepted, it is based on series of ISO/IEC standards (25000-25099) [1]. According to this model the quality evaluation is three-level and consists in determination of required features for each software type, attributes for each feature and metrics for each attribute. Standards also specify main features for all software types: functionality, usability, efficiency, reliability, maintainability and mobility.

Several software functionality attributes can be distinguished such as suitability and precision of functions performance, interoperability with third party software and hardware automation [2]. The paper [3] provides software functionality evaluation methods and justifies testing utilization for this purpose.

2 Problem Formulation

At present, the problem of creating a methodology for testing and evaluating the main quality indicators of specialized software for monitoring and managing ship's power plants in accordance with international standards is topical. The availability of such a methodology will make it possible to obtain a composition of software quality attributes and quantitative indicators, indicate the direction of priority changes and assess the results of their implementation, efficiency and reliability of specialized software.

3 Problem Solution

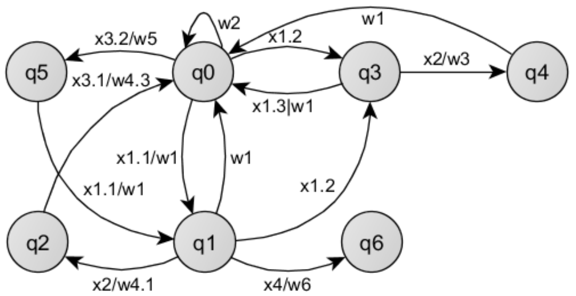
The requirements to be met by the software are the basis for the execution of its functionality testing. Since software is an integral part of the hardware-software system the availability of models that simulate behavior of hardware and automation object as an entity is required for its testing. The functionality testing is usually divided into several levels: unit, integration and system [4]. The test procedure consists of the execution of a sequence of actions: the definition of testing completeness criterion; the development of a complete set of test cases; the execution of a report with the information on testing results. The results of the testing execution should be metric values of functionality attributes – suitability, precision and interoperability.

The task of the unit testing is metrics definition of functionality attributes of separate software components. The behavior in the design mode differs significantly from the behavior in the electric plant operation mode for most software components and is implemented (designed and coded) as two individual finite-state automata. The use of two automata enables to separate calculation of suitability metrics (completeness of test coverage for automaton that is simulating behavior in the diagram design mode) and interoperability metrics (for behavior in the electric plant operation mode).

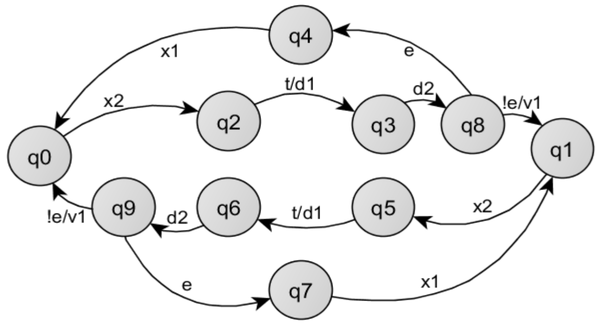
As shown in [5, 6], the transition graphs of digital automata are successfully used to describe ship automation systems. In this paper, transition diagrams are used to describe the behavior of software' components.

Thus, the completeness of automata transition coverage is determined to be the criteria of unit testing completeness for all components; the complete set of test cases is being written using automata transition conditions. Since this test phase refers to debugging, the transition to functionality adding and performance of the next test phase only occurs if 100% of test cases are covered.

As an example the automatic circuit breaker component testing is examined. Fig.1 shows automata that describe component behavior in the diagram design mode (Fig.1, a) and in the electric plant operation mode (Fig.1, b).



a



b

Fig.1. Behavior of the automatic circuit breaker component.

Set of states of the automata shown in Fig.1, a, is

$$Q = \{ q0, q1, q2, q3, q4, q5, q6 \},$$

$q0$ – object is created (the initial state); $q1$ – object is in focus (highlighted by markers); $q2$ – the subject is moving (condition is complex); $q3$ – communication with bus available; $q4$ – start communication with bus; $q5$ – end of the communication with bus; $q6$ – the object is removed (final state).

Set of states of the automata shown in Fig.1, b, is

$$Q = \{q0 - q9\},$$

$q0$ – the circuit breaker in the initial state; $q1$ – The circuit breaker in a state opposite to the initial, $q2$ – waiting for the expiry of the circuit breaker trip time; $q3$ – waiting for a response from the working area; $q4$ – waiting for the user's actions on the elimination of errors; $q5$ – waiting for the expiry of the return trip the circuit breaker time; $q6$ – waiting for a response from the working area; $q7$ – waiting for the user's actions on the elimination of errors; q8 – verification of the data packet with the odd change of state of the circuit breaker; $q9$ – verification of the data packet for even change the status of circuit breakers.

The same automata are used both for testing and for describing and coding of component behavior; the state transition verification is done by built-in compiler facilities for programs debugging. The full coverage of the automaton shown in Fig.1, a, comes with the following message sequence generation

$$x1.1 \to x2 \to x3.1 \to x1.2 \to x2 \to w1 \to w2 \to x3.2$$
$$\to x1.1 \to\to x1.2 \to x1.3 \to x1.1 \to w1 \to x1.2 \to x2$$
$$\to w1 \to x1.1 \to x4;$$

For automata shown in Fig.1, b,

$$x2 \to t \to d2 \to e \to x1 \to x2 \to t \to d2 \to !e \to$$
$$\to x2 \to t \to d2 \to e \to x1 \to x2 \to t \to d2 \to !e.$$

The final testing level is the system one. The power plant operation is a specific sequence of diesel generator units’ and actuating units’ operation according to the technological process. The aim of the system testing is a software health check when controlling electric power plant, that’s why the software requirements coverage can be distinguished as a criteria of testing completeness.

The software for the monitoring and control of the electric power plant is a discrete-event system, so finite-state automata could be used as means of the test case’s development [3]. Although, the software is also a component of hardware-software system, so the R-diagram test sequence should be specified to enable the definition of functions that are subject to examination, as well as software models and tools used for this process [7].

System testing tasks include also functions of the checking of tools required for performing actions by the Analyst – analysis of content of the data exchange packages, data channel load statistics, work schedule by power units and electric load. This checking is possible only after certain time of software operation in the electric plant operation mode, and the only functionality attribute resulting from such checking is suitability.

Weighting factors are defined for the calculation of numeric values of suitability, interoperability and precision for each component and for each tool required for the Analyst work. Then attribute values are calculated via additive reduction and normalization:

$$M = \frac{\sum_{i=1}^{15} k_{m.n.i} \cdot k_{p.m.i} \cdot \alpha_i}{\sum_{i=1}^{15} \alpha_i}, \qquad (1)$$

where M – attribute value; $k_{m.n.i}$ – test coverage value; $k_{p.m.i}$ – testing results; α_i – metric's weighting factor.

The following attribute values were calculated for the monitoring and control software of the marine electric generating plant basing on the results of testing: suitability – 0.91; interoperability – 0.86; precision – 0.81. In analyzing of all test phases' results it can be concluded that the values of «suitability» and «interoperability» attributes can be improved by system testing strengthening, and the improvement of precision is related to changes in software functionality.

The functionality value is calculated by similar algorithm: weighting factors are given for attributes and the numeric value is calculated via reduction and normalization. When setting factors for suitability and interoperability as 1, and for precision – as 0.8, the functionality value will be – 0.86.

Evaluation software is practicality in terms of values of its three attributes: ergonomics, clarity and efficiency of development. For each of the attributes which are allocated sub characteristics – indicators t are measured numerically. In [8], a methods of evaluating usability metrics was among the most widely used identified, peer review and survey tests. There were m users involved in order to determine the parameters of the study. Based on the data obtained from the survey, the normalized average and the resulted statistical will be used as indicator.

$$S_j = \frac{1}{k} \cdot \frac{\sum_{i=1}^{m} \alpha_{ij}}{m}, \qquad (2)$$

where k – normalization factor (maximum score – 10); α_{ij} – assessment of the j-index i-th user; m-number of users surveyed.

Further, introducing weighting factors has kind of importance to the additive convolution calculates attribute values

$$A_j = \frac{1}{\sum_{i=1}^{m_j} p_{ij}} \cdot \sum_{i=1}^{m_j} \left(S_{ij} \cdot p_{ij}^{(n)} \right), \qquad (3)$$

where $p_{ij}^{(n)}$ weight of i-j-index of the attribute; S_{ij} – normalized average statistical value of i-index j-th attribute; m_j-number of indicators of the j-th attribute.

To visualize the results of the calculation are encouraged to use the metric radial diagram, with the help of which it is convenient to display the "covering" performance requirements (Fig.2).

Presentation of usability evaluation using a single number does not identify possible problem areas, but it allows you at different stages of development and implementation of software to determine the need for further changes to the results of the implementation.

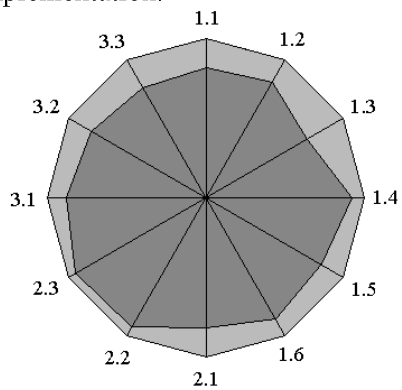


Fig.2. Coverage to the practical performance requirements.

The most important weights of the attributes, chosen among – the lowest calculated values

$$\min_{A_j} \max_{p_j} \left\langle A_j,\ p_j^{(a)} \right\rangle, \qquad (4)$$

and then select the smallest value from the largest weight indexes of attributes that have been selected in the previous step

$$\min_{S_{ij}} \max_{p_{ij}} \left\langle S_{ij},\ p_{ij}^{(n)} \right\rangle. \qquad (5)$$

The result of this procedure will be having a variety of indicators, in the direction where the improvement is most important. To facilitate the calculation of minimax criteria, you can calculate the values of attributes and metrics based on the "reverse" weight, calculated as the cofactor 2 "true" weight.

$$A_j^* = \left(2 - p_j\right) \cdot A_j, \qquad (6)$$

$$S_{ij}^* = \left(2 - p_{ij}\right) \cdot S_{ij}. \qquad (7)$$

Then determine the most critical indicators in order to improve software usability which will meet determination procedure lowest values supplemented by attributes, and then – among the lowest values of indicators supplemented by

$$\min_{p_{ij}} \min_{A_j} \left\langle A_j^*, \left\{ S_{ij}^* \right\} \right\rangle \quad (8)$$

After calculation chart "covering" the requirements and the numerical value of practicality were obtained – 0.86. For example, the value and coating were found unsatisfactory, therefore it decided to modify the software. For that of the weights greatest attributes is chosen – 1.0, which stands for "for this software ergonomics is the most important". Further parameters of the weighting factors, which are components of ergonomics, the largest has been chosen.

The modern software development process is iterative and includes several stages: requirements analysis, architecture development, coding, testing and debugging, documentation, implementation and maintenance. Calculation of the key quality indicators for the considered category of software determines the need to return to earlier stages of development (analysis of requirements – coding) or transition to the following (documenting and further). After selecting the features in the direction of which it is necessary to make priority changes, the software interface is corrected and then the survey is conducted again and its results are processed in this way. This process is repeated until the "coverage" of requirements and the value of practicality are not satisfactory.

4 Conclusion

In work the methodology of a quantitative estimation of the basic indicators of the specialized software quality for monitoring and management of marine electric power systems – functionality, practicality and reliability is considered. Using this methodology allowed to obtain the composition of software attributes and quantitative values of metrics, specify the directions of priority changes and evaluate the results of their implementation, determine the reliability of software. Based on the analysis of all stages of testing, it is established that the functionality of the software can be enhanced by strengthening system testing and changing some algorithms of the software.

The software functionality enhancement is related to the strengthening of system testing (enhancement of suitability and interoperability) and to the upgrade of interaction algorithms between hardware and software automation (improving of precision). The software usability can be improved by upgrade of the means of specifying incorrect setting up for data exchange and by simplifying of the succession of the user moves when performing complex operations (data flow analysis etc.). Temporary software efficiency depends on topology and electrical power system composition as well as on the rate of traffic from software components, so the attributes should be calculated for each automated control system of the marine electric generating plant.

References:

[1] K. Esaki, System Quality Requirement and Evaluation, *Global Perspectives on Engineering Management*, Vol.2, No.2, 2001, pp. 52-59.

[2] T. Danilina, Software Quality Evaluation With Respect to International Standards, *Electronic and computer systems*, Vol.7, 2012, pp. 266-269.

[3] V. Kuliamin, A. Petukhov, Review of the Techniques for Constructing of Covering Sets, *Programming*, Vol.3, 2011, pp. 3-11.

[4] V. Kuliamin Component Architecture of Model-based Testing Environment, *Programming and Computing software*, Vol.36, No.5, 2010, pp. 289-305.

[5] Al-Suod Mahmoud M., A. Ushkarenko, O. Dorogan, Automatic Approach for Estimating the Protection Elements of Electric Power Plants, *World Academy of Science, Engineering and Technology, International Science Index, Energy and Power Engineering*, Vol.2, No.12, 2015, pp. 3935–3937.

[6] Al-Suod Mahmoud M., A. Ushkarenko, O. Dorogan, Monitoring and Automatic Control for Ship Power Plants Based Logical Algorithms, *International Journal of Advanced Computer Research (IJACR)*, Vol.4, No.17, 2014, pp. 966–972.

[7] O. Tarasiuk, V. Kharchenko, Dynamic Radial Metric Diagrams in Tasks of Software Quality Management, *Collection of studies of Pukhov Institute for Modelling in Energy Engineering*, Vol.22, 2003, pp. 202-205.

[8] H. Rex Hartson, Terence S. Andre, Robert C. Williges, Criteria for Evaluating Usability Evaluation Methods, *International Journal of Human-Computer Interaction*, Vol.15, No.1, 2003, pp. 145-181.