Evaluation of significance of control directives for software errors detection in functional programming of on-board computers

V A Drabov, N V Romanova, T S Novikova, A V Vernigor, A V Rekovets and N V Ptitsyna

Smolensk State Agricultural Academy, 10/2, Bolshaia Sovetskaia street, Smolensk, 214000, Russian Federation

E-mail: romanovanv0403@yandex.ru, sgsha@sgsha.ru

Abstract. Modern farm machinery is equipped with on-board computers that execute complicated tasks. The main efforts are aimed at selecting and justifying methods for adequate hardware-software translation of functional programming from one computer system to another, while increasing its stability to failures of computer hardware. The development of approaches to the selection of procedures for detecting errors determines the degree of their relative information value when analyzing the programs progress. As it follows from the analysis of error types in functional programming, it is established that errors occur at the stages of forming the technical task, when developing algorithms, when programming, and when configuring programs. Errors have different frequency and different weight in terms of the impact on the functioning of on-board computers used in agricultural machinery. The article contains a detailed list of errors that must be detected when debugging working programs of on-board computers. Based on expert assessments, the errors are divided into 33 different types and formed into 6 groups. By combining errors, it is possible to minimize the time spent on performing static debugging of the functional programming of on-board computers. Using the hierarchy analysis method, matrices of paired comparisons of control directives with error types in each group are formed. Thus, the relationship between groups of errors, control directives, and values of error types is established. The first thing, programs debugging is performed using the control directives that have the highest degree of significance, since they can be used to identify and eliminate the most dangerous errors.

1. Introduction

Computer-based system as a main element in agricultural machinery, implemnts the operation algorithm and completes tasks of functional programming in real time mode. Moreover with a serial upgrading of any unit of agricultural machinery where on-board computers are used, the existing functional programming is widely used along with a newly written one, as it is practically impossible to repeate the whole volume of its in-process tests [1].

The so-called simulators, special program models that simulate the functioning of computer systems are used for debugging the functional software. Nowadays this issue is becoming more and more intensified as far as the volume of processed data in on-board computers of agricultural machinery is increased and the topology of information flow is becoming more complicated [2; 3].

The main stages of validity testing of functional programming are the so-called static and dynamic
debugging. The main purpose of a static debugging is the identification and elimination of errors in program bodies.

2. Materials and methods

The research material is functional programming of on-board computers used in agricultural machinery. In the process of developing functional software errors inevitably occur in the programs. They are caused by inaccuracies in the technical tasks, imperfection of the used algorithm and programming errors. [4]

The assessment of the degree of experts’ opinions was made on rank correlation coefficients:

$$P_{\alpha\beta} = 1 - \frac{6}{N(N^2-N)} \sum_{i=1}^{N}(P_\alpha - P_\beta)^2$$  \hspace{1cm} (1)

Where $N$ – the initial number of experts, $P_\alpha$ and $P_\beta$ – evaluation of ranking of some parameters of text tasks $\alpha$ and $\beta$ by the experts.

To calculate the expert evaluation coefficient (EEC) the expert is not to specify a certain number but a possible interval $[q_1, q_2]$, $q_2 > q_1$, then for the $j$-type error and $m$ number of experts a $Q_j$ range of interval expert assessments is formed:

$$Q_j = \left\{ q_{j1}^{(1)}, q_{j2}^{(1)} \right\}, \left\{ q_{j1}^{(2)}, q_{j2}^{(2)} \right\}, ..., \left\{ q_{j1}^{(m)}, q_{j2}^{(m)} \right\}$$  \hspace{1cm} (2)

To choose a generalized interval of rates of EEC the minimum and maximum intervals are used:

$$q_{j1} = \min\left\{ q_{j1}^{(1)}, q_{j1}^{(2)}, ..., q_{j1}^{(m)} \right\}$$

$$q_{j2} = \max\left\{ q_{j2}^{(1)}, q_{j2}^{(2)}, ..., q_{j2}^{(m)} \right\}$$

The arithmetical average of EEC is taken as shared experts’ opinion

$$Q_j = \frac{1}{2}(q_{j2} + q_{j1})$$  \hspace{1cm} (3)

3. Evaluation of software error detection

The organization of the functional programming involves the development of supporting means of the static debugging process based on the application of special patches in the general program body, the so-called control directives (CD) for the search for possible crashes or errors in main program commands. It is obvious that their quantity should not be large as they demand some amount of memory and time resources consumption of the computer system.

The evaluation of control directives for software error detection of functional programming is a complicated, time-consuming and considerably creative process. Its peculiarity results in impossibility of its formalization by means of common mathematical methods as they do not allow working out a solution in conditions of fuzzy initial data presented not in numeral but in qualitative form. That is why a method of expert evaluation is applied for the offered methodology for solving this hard-to-formalize task. [5]

The first stage of a methodology for evaluating of control directives for software error detection of functional programming is the forming of the technical expert advisory group.

The final choice of expert group is carried out according to a maximum value of the rank correlation coefficient based on the Formula 1. This figure is $m = 6$.

At the second stage of a methodology for each error type of the functional software EEC within the interval $[0,1]$ is calculated. It reflects the influence rate of errors on the on-board computer system
At the same time the one-dimensional or qualimetry condition is met. Due to this condition EEC for each error types is calculated by one number. The error classification is shown in the Figure 1.

Figure 1. Error Classification

At the third stage a choice of control directives for software error detection in functional programming is carried out. A control directive is a special technological program as a component of the software development automatization system. It is designed for support a user’s checkout method of the selected part of the program with address (memory and register), operand values and results visualization.

It is a natural programmers’ attempt to compose a separate control directive to detect each type of errors. However, this approach makes the task of debugging the programs more complicated. It is important to minimize the quantity of directives. One and the same control directive should be preferably used to detect several error types in different groups. It can be actualized on the assumption of the possibility of joint application of several control directives for error detection in each group.

As a result of this approach the experts calculated and formed five universal control directives for the error detection in six selected groups. (Table 1.).

Table 1. Control Directive’s (CD) Name and Purpose

| Code | CD Name                                      | CD Purpose                                                                                   |
|------|---------------------------------------------|----------------------------------------------------------------------------------------------|
| CD1  | Recording the program progress by the control counter when passing the address interval \(Ai - Aj\) | Reading a functional programming fragment from the address \(Ai\) to \(Aj\) for further analysis and checkout |
| CD2  | The timing of the progress of the program counter between addresses \(Ai - Aj\)               | Changing the program execution time and comparing it with the allocated time resource          |
| CD3  | Circularity Measurement                      | Calculation of the number of cycles and comparison of it with the required value              |
| CD4  | Recording the program progress at the RAM Executive address                                | Checking operands and results accuracy in RAM                                                 |
| CD5  | Stop recording the program progress at the RAM executive address                           | Control of the fact that a given operand is written to a specific RAM executive address       |
At the fourth stage of the methodology, models of relationships between control directives and error types in each group are built to assess the significance of each control directive.

In Figure 2 for example, a model of relationship between control directives and error types in inter-program interfaces. The control directives values for each type of error are shown in circles.

![Diagram of control directives and error types](image)

Figure 2. Model for informative value of control directives for detecting errors in inter-program interfaces

The hierarchy analysis method is used to process expert information when determining the significance values of control directives [5; 6].

The value of each control directive for a group is defined as the weighted average sum of the values of the control directives in relation to the types of errors in the group (Table 2).

| CD1 | CD2 | CD3 | CD4 | CD5 |
|-----|-----|-----|-----|-----|
| CE  | 0.57| 0.15| 0.08| 0.2 |
| LE  | 0.77| 0.23| 0.315| 0.315|
| IOE | 0.37| 0.0 | 0.0 | 0.0 |
| DME | 0.42| 0.0 | 0.0 | 0.0 |
| IIE | 0.56| 0.1 | 0.29| 0.29|
| SE  | 0   | 0   | 0.17| 0.17|

Table 2. Degree of Control Directives Value for Groups of Errors

4. Conclusion

Thus, to find and localize errors while analysing the programs progress, it is proposed to use special control directives designed to control the process of debugging functional software. The control directives are divided into two main types. The first type is implemented by placing stop addresses or interrupting the program's progress. The second type is implemented by specifying the operands names (physical addresses), whose values are registered each time the program accesses them, either only when reading or only when writing their values.
References

[1] Morozov D S 2014 Topicality of the principles of test driven development within the scope of assessing results of practical activity regarding software engineering *European Applied Sciences* **10** 69-72

[2] Bazhenov A S and Itsykson V M 2013 Forecasting software development project characteristics using meta-modeling *ACM Int. Conf. Proceeding Series*

[3] Furrer F J 2019 Future-proof software-systems: a sustainable evoution strategy *Future-Proof Software-Systems: A Sustainable Evolution Strategy* pp 1-376

[4] Kalikhman D M, Kalikhman L Y, Sadomtsev Y V, Deputatova E A, Turkin V A, Nakhov S F, Nikiforov V M and Tchaikovsky M M 2013 Principles for designing digital robust controllers for measurement channels of angular elocity and proper acceleration in modern strapdown inertial navigation systems and software for their monitoring *20th Saint Petersburg Int. Conf. on Integrated Navigation Systems, ICINS 2013 - Proceedings* pp 343-350

[5] Godlevsky M D, Shekhovtsov V A and Braginsky I L 2012 Principles of modeling, assessment and quality management of the software development process *East European J. of Advanced Technologies* **5(3(59))** 45-49

[6] Kulakov A V, Golovanov V E, Gorbacheva L A, Yusupov S K, Semishkin V P and Shary N V 2016 Development of principles for restoring software and hardware complexes as a means of increasing reliability and maintainability in complex automated process control systems *Heavy mechanical engineering* **5** 22-26