On the application of a model for assessing the transactional reliability of information systems of industrial enterprises

I V Kovalev¹²³⁴, D I Kovalev², N A Testoyedov⁴⁵, D V Borovinskii⁶, A A Koltashev⁵, A K Shatrov⁵ and V A Bartenev⁵

¹Siberian Federal University, 79, Svobodny pr., Krasnoyarsk, 660041, Russia
²Krasnoyarsk State Agrarian University, 90, Mira pr., Krasnoyarsk, 660049, Russia
³China Aviation Industry General Aircraft Zhejiang Institute Co., Ltd, China
⁴Reshetnev Siberian State University of Science and Technology, 31, Krasnoyarsky Rabochy Av., Krasnoyarsk, 660037, Russia
⁵JSC “Academician M F Reshetnev Information satellite systems”, 52 Lenin street, Zheleznogorsk, Krasnoyarsk region, 662972, Russia
⁶FSBEE HE Siberian Fire and Rescue Academy EMERCOM of Russia, Severnaya street, Zheleznogorsk, Krasnoyarsk region, 662972, Russia

E-mail: kovalev.fsu@mail.ru

Abstract. The article examines an example of the application of a model for assessing the transactional reliability of information systems of industrial enterprises. This transactional reliability determines the correctness of the calculations performed and the integrity of the data, and, consequently, the characteristics of the quality of software tools that are used in information systems of industrial enterprises to perform control and data processing functions. It is shown that all the parameters of this model can be obtained from the results of testing the system, which makes it possible to construct adequate estimates of the reliability parameters. The calculated indicators obtained using the developed model differed from the real ones by no more than 5%, which indicates the sufficient adequacy of the model. The modern system Microsoft Dynamics 365 for Finance and Operations (Dynamics AX/Axapta) was used as an environment for modeling and analyzing the transactional reliability of software tools.

1. Introduction
When developing critical projects of information systems of industrial enterprises, developers try, to one degree or another, to obtain an assessment of the reliability of software. This is usually done based on the final test results of the system. However, modern software used in complex information systems uses a colossal amount of data in its work, passing through standard modules and functions. Therefore, it is almost impossible to identify all the connections and ways of information processing, even for a fairly simple program. Based on this, the detailing of the elements (conditionally called software modules) of the reliability calculation should be limited to complete software components. These components, interacting with each other, constitute a more complex combination, for which we need to calculate the operational (hereinafter referred to as "transactional") reliability.

Currently, transactional information processing is becoming one of the most important aspects that determine the correctness of calculations and data integrity, and, consequently, the characteristics of the
quality of software used in control and information processing systems [1-4]. The large number and ever-increasing complexity of software tools for transactional information processing require increased attention to transaction modeling throughout the development process. The transactional computing model implemented in a software tool essentially determines whether the software tool will be in a consistent state and maintain the required level of system reliability.

The main advantage of this model is that all the parameters of this model can be obtained from the data after the system test, which makes it possible to construct estimates of the reliability parameters [5-7].

2. Materials and methods

2.1. Information systems of industrial enterprises

Such a model is useful for evaluating information systems for processing and storing data, for example, for modern Enterprise Resource Planning (ERP) systems [8]. These systems allow not only to maintain operational (goods, services, production), financial, tax accounting at the enterprise, but also allow the management, the planning department to build budgets and monitor their implementation, plan the activities of the enterprise, manage the enterprise, analyzing the planned and actual indicators in the system.

Systems of this class are more focused on working with financial information for solving problems of managing large corporations of various types of activities with geographically separated resources. This includes everything that is necessary to obtain resources, manufacture products, transport them and settlements for customer orders. In addition to the listed functional requirements, ERP implements new approaches to the use of graphics, the use of relational data, CASE-technologies for their development, architecture of computing systems of the "client-server" type and their implementation as open systems.

ERP is a methodology aimed at efficient management of all production resources of an enterprise [9]. It provides a solution to the problems of planning the activities of the enterprise in physical and monetary terms, modeling the capabilities of the enterprise. This methodology is based on a number of large interrelated functionalities, including:

- Business Planning (BP);
- Sales and Operations Planning (S&OP);
- Production Planning (PP);
- Master Production Scheduling (MPS);
- Material Requirements Planning (MRP);
- Capacity Requirements Planning (CRP);
- Shop Floor Control (SFC);
- Just-in-Time (JIT);
- Forecasting (F);
- Project and program management (PPM);
- Maintaining information about the range and composition of products (MIR&CP);
- Maintaining information on technological routes, supply chains (MITR&SC);
- Cost management (CM);
- Financial management (FM);
- Human resources management (HRM).

An enlarged ERP management structure is shown in figure 1. The presence of a powerful infrastructure and methodology for building systems contributes to the achievement of a high level of efficiency in the implementation of management systems such as ERP in industrial enterprises. According to some estimates, the introduction of such systems can lead to a reduction in stocks up to
30%, an increase in labor productivity up to 25%, an increase in the number of orders completed on time, up to 20%.

2.2. Dynamics AX / Axapta

The modern system Microsoft Dynamics 365 for Finance and Operations (Dynamics AX/Axapta) was chosen as an environment for the development of a system of model-algorithmic support for analyzing the transactional reliability of software tools. This is an automated enterprise management system that provides the management of the company and its employees with the most complete information necessary for a successful business.

Dynamics AX is a scalable system for midsize and large enterprises, corporations and holding structures, providing a single, integrated solution aimed at increasing business manageability and increasing enterprise profits.

Dynamics AX is a powerful tool that realizes opportunities for both operational operation of the enterprise and for the formation of a long-term strategy, which allows you to lay the foundation for successful operations.

The uniqueness of the Dynamics AX system is in its modern advanced technology, which provides a single information space for the enterprise. Powerful functionality, optimal price / quality ratio, ease of learning, ease of use and configuration of the system make it possible to build an information management system at an enterprise that corresponds to its industry and individual characteristics.

![General scheme of the ERP](image)

**Figure 1.** General scheme of the ERP.

Facts in favor of choosing Dynamics AX:

- Dynamics AX system is written in the object-oriented language X ++, similar in syntax to C ++ and Java;
- Dynamics AX system has an open source code available for modifications;
Dynamics AX system has built-in visual programming tools;
Dynamics AX system has about 2000 classes that implement the business logic of the system;
Dynamics AX system has a built-in code profiler with run-time granularity down to a line of source code;
Dynamics AX system has built-in tools for monitoring and tracing errors;
Dynamics AX system has the ability to organize distributed computing.

The system of model-algorithmic support for the analysis of transactional reliability of software is implemented as a module "Reliability Models" of the ERP-system Dynamics AX.

The logic of the system of model-algorithmic support for the analysis of transactional reliability of software is written in X++ in the form of a class hierarchy and is integrated into the logic of Dynamics AX [8, 9]. Final and intermediate results are saved in the database for further analysis. The module is built as a set of dialog boxes and reports for interactive interaction with the user [10]. The user can create new parameters for the model, adjust existing parameters, make calculations of reliability parameters, save results and build reports on already performed and saved calculations. After opening the module, the user can configure the main reference books of the module: parameters of class descriptions, descriptions of transactions. After choosing a model for assessing transactional reliability, the user must fill out reference books about the classes used or transactions included in the calculations, as well as operational profiles based on the test results. Calculations are performed automatically and controlled by the user. The entered and calculated data are saved in the database. The results of past calculations can be printed using reports.

2.3. Model for assessing the transactional reliability of information systems

Reliability of software is defined as the probability of functioning without errors for a certain period of time in a particular operating system. For most developers, reliability is defined as the correctness of the software, i.e. the number of errors to be corrected during the test.

Transactional reliability differs in its meaning from the classical understanding of reliability. It can be used to assess the reliability of software for processing and storing data, where the logical unit of work is a transaction [11].

Transactional reliability depends not only on the reliability of the components, but also on the specific problem area for which the software was developed, formally - on the set of operational profiles of the components. An operational profile is a set of disconnected and non-overlapping input ranges and the probabilities of their use by components. Profiles may differ for different operating systems. Obtaining an operational profile requires dividing the input range space into subsets (leaves), and then estimating the probabilities of using each subset (leaf). A subset with a sufficiently high probability of use can be divided into smaller zones. Profiles are tables that include information about the ranges of the input data of the module and the probability of failure on a given range of data [2-4].

To calculate the overall transactional reliability \( R_{tr} \), it is necessary to compile a complete list of all transactions and calculate the reliability of each transaction using the operating profiles.

A transaction is a sequence of actions that is considered completed successfully if there were no failures at any step. A transaction can be either a single transaction of information systems, or a certain, logically completed, independent sequence of user actions in the system. For correct calculations, you must select one type of transaction.

Let's define:
- \( m \) – total number of classes;
- \( n \) – total number of transactions;
- \( K_i \) – class, \( i = 1, \ldots, m \);
- \( O_i \) – operating profile (multiple input ranges) \( i \)-th class, \( i = 1, \ldots, m \);
- \( F_i \) – vector representing the probabilities of failure \( i \) class for each input range, \( i = 1, \ldots, m \);
- \( f_{i,l} \) – \( l \)-th element vector \( F_i \), \( l \) – vector dimension \( F_i \);
- \( T_j \) – transaction, \( j = 1, \ldots, n \);
\(D_j\) – many classes, belonging to the \(j\)-th transaction, \(j=1,\ldots,n\);

\(H_{ij}\) – vector of probabilities of using the range of values of the \(i\)-th class in the \(j\)-th transaction;

\(h_{ijk}\) – \(k\)-th vector element \(H_{ij}\);

\(W_{ij}\) – weight (reliability) \(i\)-th class \(j\)-th transaction;

\(P_U\) – the probability of using the \(j\)-th transaction, where \(\sum_{j=1}^{n} P_U = 1\);

\(R_o\) – reliability of the \(j\)-th transaction;

\(R_t\) – transactional reliability of the system.

The weight (reliability) of the \(i\)-th class of the \(j\)-th transaction is determined by the formula:

\[
W_{ij} = \sum_{k=1}^{l} h_{ijk}(1 - f_{ik}).
\]  

(1)

The reliability of a transaction can be calculated as:

\[
R_{tj} = \prod_{k=1}^{n} W_{jk}.
\]  

(2)

After calculating the reliability of all transactions, the total transactional reliability of the system is calculated:

\[
R_{tr} = \sum_{j=1}^{m} P_U \times R_{tj}.
\]  

(3)

3. Results

As an example, consider the “Master Planning” module of the Dynamics AX system, implemented at the enterprise of the Krasnoyarsk Territoy in the «FORS» company [6].

As a result of testing, 4 transactions were identified, which in total use 18 classes with a maximum number of input ranges of 4.

Table 1. Operating profiles of classes.

| Classes \((K_i)\) | Failure probabilities in the range |
|----------------|-----------------------------------|
| 1 | 2 | 3 | 4 |
| K1 | 0.000010 | 0 | 0.000070 | 0 |
| K2 | 0 | 0 | 0.000010 | 0.000010 |
| K3 | 0 | 0.000040 | 0.000550 | 0 |
| K4 | 0 | 0 | 0 | 0.000050 |
| K5 | 0 | 0 | 0.000020 | 0 |
| K6 | 0.000340 | 0 | 0 | 0 |
| K7 | 0 | 0 | 0.001100 | 0 |
| K8 | 0 | 0 | 0 | 0.000090 |
| K9 | 0 | 0.000001 | 0 | 0 |
| K10 | 0 | 0.000440 | 0 | 0 |
| K11 | 0.000670 | 0 | 0 | 0 |
| K12 | 0 | 0 | 0 | 0.000010 |
| K13 | 0.000180 | 0 | 0 | 0 |
| K14 | 0 | 0.000580 | 0 | 0 |
| K15 | 0 | 0 | 0.000001 | 0 |
| K16 | 0.000010 | 0 | 0 | 0 |
| K17 | 0 | 0 | 0.000050 | 0 |
| K18 | 0 | 0.000010 | 0 | 0 |

The probabilities of using transactions are:
Let's describe subsets of classes included in transactions:

- \( D_1 = \{ 1, 2, 4, 5, 6, 8, 9, 11, 13, 14, 15, 17, 18 \} \);
- \( D_2 = \{ 1, 2, 3, 4, 5, 6, 16, 17, 18 \} \);
- \( D_3 = \{ 2, 3, 5, 7, 8, 10, 12, 15, 16, 17 \} \);
- \( D_4 = \{ 6, 7, 8, 15, 16, 17, 18 \} \).

The vectors of the probability of using the ranges and the vectors of the weights are as follows (see table 2).

The vectors of the probability of using the ranges and vectors of weights of transactions T2, T3 and T4 are formed in a similar way.

**Table 2. Vectors of the probabilities of using the ranges and of the weights of the T1 transaction.**

| Classes \((K_i)\) | Ranges use probabilities |
|-------------------|--------------------------|
|                   | 1 | 2 | 3 | 4 | \( W_{ij} \) |
| K1                | 0 | 0.100000 | 0.900000 | 0 | 0.999937 |
| K2                | 0.500000 | 0.500000 | 0 | 0 | 1 |
| K4                | 0 | 0.140000 | 0.030000 | 0.830000 | 0.999959 |
| K5                | 0.150000 | 0.250000 | 0 | 0.6 | 1 |
| K6                | 0.005000 | 0.044000 | 0.456000 | 0.495000 | 0.999998 |
| K8                | 0 | 0.550000 | 0.262000 | 0.188000 | 0.999983 |
| K9                | 0.999000 | 0.001000 | 0 | 0 | 1 |
| K11               | 0 | 0.727000 | 0.005000 | 0.268000 | 1 |
| K13               | 1 | 0 | 0 | 0 | 0.999820 |
| K14               | 0 | 0.441000 | 0.334000 | 0.225000 | 0.999744 |
| K15               | 0 | 0.456000 | 0 | 0.544000 | 1 |
| K17               | 0.745000 | 0.111000 | 0 | 0.144000 | 1 |
| K18               | 0 | 0.498000 | 0.229000 | 0.273000 | 0.999995 |

After that, having received all the necessary data, we can calculate the reliability of each transaction using the methodology presented in the article:

- \( R_{t1} = 0.999436 \);
- \( R_{t2} = 0.999855 \);
- \( R_{t3} = 0.999662 \);
- \( R_{t4} = 0.999710 \).

As a result, we get the final transactional reliability of the system \( R_{tr} = 0.999649 \).
4. Discussion
Analyzing the calculation results, we conclude that when working with these types of documents (four types of transactional data processing), the system will fail in 351 cases out of 106, provided that the operator does not make mistakes.

As a result of the real test, according to which the above calculation was made, out of 106 transactions, 338 transactions were unsuccessful.

The value of the experiment lies in the fact that both classes and their input ranges were identified that contribute the most to the probability of failure. This allowed us to pay more attention to testing and identifying errors in the code and in the "blockage" of the system under test.

The greatest weight in the probability of failure is made by the classes: K7, K10, K14. By reducing the probability of failure on input ranges in these classes by half, you can achieve the following results:

- \(R_{t1} = 0.999564\);
- \(R_{t2} = 0.999855\);
- \(R_{t3} = 0.999784\);
- \(R_{t4} = 0.999860\).

As a result, we get a new transactional reliability of the system \(R_t = 0.999750\).

5. Conclusion
During the full system testing phase, you can find the blockage in the system - those components that contribute the most to the likelihood of failure.

By knowing exactly which operating profiles and input ranges are most likely to fail, we spend less time finding and fixing bugs in our code.

For a ready-made system for processing and storing information with specified operating profiles, it is possible to calculate the final transactional reliability and system availability, as well as track their changes in the course of eliminating errors in the blockage.

As a result of using the system in a real project, it was possible to identify the blockage in the system and improve the reliability indicators. The calculated indicators obtained using the developed models differed from the real ones by no more than 5%, which indicates the sufficient adequacy of the model.

References
[1] Kovalev I V et al 2020 IOP Conf. Ser.: Mater. Sci. Eng. 862 062105
[2] Kovalev I V et al 2019 IOP Conf. Ser.: Mater. Sci. Eng. 537 052038
[3] Petrosyan M O et al 2016 IOP Conf. Ser.: Mater. Sci. Eng. 122 012026
[4] Tsarev R Yu et al 2016 IOP Conf. Ser.: Mater. Sci. Eng. 155 012041
[5] Antamoshkin O A et al 2016 IOP Conf. Ser.: Mater. Sci. Eng. 155 012043
[6] Kovalev I V et al 2015 IOP Conf. Ser.: Mater. Sci. Eng. 70 012008
[7] Kuznetsov P A et al 2015 IOP Conf. Ser.: Mater. Sci. Eng. 94 012019
[8] NetU: Business solutions Retrieved from: https://www.netugroup.com/category/our-services/business-solutions/
[9] Kozhuhova O A and Kukartsev V V 2011 Actual problems of aviation and astronautics 1(7) 448-9
[10] Selezneva S V 2010 Science and modernity 1-2 121-3
[11] Demin A M 2007 Bulletin of the Ivanovo State Power Engineering University 2 60-8
[12] Kukartsev V V et al 2011 Reshetnev readings 2(15) 703-5