The application of relational interactive logic in control operation problems by the example of monitoring the server equipment of transport systems

D V Zakharov¹, O V Kuzmin¹

¹ Irkutsk State University, 1, Karla Marksa st., Irkutsk, 664003, Russia

E-mail: zakharov-danyl@mail.ru

Abstract. The article demonstrates the use of relational interactive logic in solving the problem of monitoring the server equipment of the transport control system. An example of such a problem is the problem of analyzing log records of monitoring server hardware. The server is considered as a special case of the physical control system. The conditions for the correctness of the current state of the equipment are defined in the form of systems of logical equations and inequalities. Solving a system of logical equations by executing a SQL query on a set of tables describing the elementary control systems in the server allows us to judge the deviation of the hardware parameters of the equipment. Recording the correctness conditions in the form of systems of logical equations provides the ability to take into account both the peculiarities of the functioning of the server hardware and some external context within which the server operates. The considered solution demonstrates the applicability of relational interactive logic in solving the problem of controlling individual classes of control systems within the scope of the system analysis discipline.

1. Introduction

With the ubiquity of automation tools, the problem of ensuring the reliability and control of various kinds of automated systems is becoming more and more urgent every day. There are a significant number of various techniques [1, 2] to ensure reliability and increase the resiliency of such systems as, for example, server equipment. The use of servers — highly targeted computers — can involve either the usual storage of files or their use for building high-load real-time systems.

It is worth noting that in a number of industries heavy demands are placed on the work of that kind of equipment with respect to stability and reliability [3]. For example, in transport systems, the life of passengers may depend on the performance of the server equipment. In addition, even a short-term failure in these systems can result in significant economic losses.

As part of this work, an approach to solving the problem of monitoring server hardware by analyzing log entries using relational interactive logic is demonstrated. An important feature is that the server is considered here as a special case of a physical control system, and the monitoring task is considered as the control task of this control system.

In paragraph 2, the concept of a control system is introduced, which is studied in the framework of cybernetics, and the main features are indicated.

The third paragraph describes the subject and method of relational interactive logic (RIL), introduces the necessary concepts and terminology.
Next, we consider the formalized model of the server in terms of the concept of a physical control system.

The fifth paragraph describes an example of analyzing the correctness of the state of the server through RIL. Special attention is paid to the representation of the model of the control system in terms of RIL and the relational data model (RDM).

2. The concept of control system

Server automation tools have been firmly anchored in railway transport systems. Both ticket services and train traffic monitoring function on the basis of a variety of server hardware platforms [4]. These systems must meet the highest standards of reliability. Various techniques are used for this in practice [5]. This section discusses the general concept of a control system.

The control system (CS) is one of the central concepts of cybernetics [6]. This term has a very broad interpretation by its very nature. Cybernetics considers the control system as the unity of the scheme, function and coordinates. The scheme CS defines the structure and interconnection of the system elements. An image of the arrangement of pieces on a chessboard or a directed graph indicating the structure of computer network nodes can be used as examples of the scheme. The function, in turn, reflects possible transformations of the system, taking into account its informational nature. In the case of chess pieces, the function defines a set of possible moves of the pieces from the current position, for a computer network, the function implements possible packet routes in the network, according to the specified addressing scheme. Coordinates allow you to set and record the current state of the system.

The implementation of the scheme and functions for the same control system are ambiguous and depend on which side of the object in question is studied within the scope of a specific task.

Control systems considered in cybernetics have a number of peculiar features. These are relativity and discreteness. Relativity in this context means that the constructed model CS always reflects a real-life object with some approximation. In solving the problem, the individual, key characteristics of the object under consideration are taken into account. For example, in the context of a separate problem, the key characteristic may be the topology of a computer network — the mutual arrangement of individual nodes, but not the type of protocol used for packet transport. Discreteness is determined by the fact that there is no continuous transition between the schemes and functions. In some cases, some parameters of the system can take on values from a certain segment of real numbers. However, as a rule, for the functioning of the system, it is only important that the value of the parameter remain within some specified range, therefore we are not interested in all the information about the parameter, but only in the fact of its deviation from the indicated boundaries. The discrete nature of control systems largely determines the tools used in their study. Set-theoretic, combinatorial [7], probabilistic [8] and especially logical methods are prevalent here [9].

When solving practical problems, it is convenient to consider CS as some set of elementary control systems. For example, when considering a computer server as a control system, it can be divided into such elementary control systems as a processor, random access memory and read-only memory.

When studying CS, we can distinguish a number of tasks: the task of analyzing the CS, the task of synthesizing the CS, and also the task of controlling the CS. The task of control is to establish and localize changes that the system undergoes under the influence of external factors. In this paper, we consider the problem of detecting incorrect states when monitoring server hardware, which can be regarded as a special case of a control task for a CS.

The next paragraph describes the subject and method of relational interactive logic, and also introduces the terminology required when describing the methodology for analyzing server logs using RIL.

3. Relational interactive logic

This section discusses the basic concepts and terms of relational interactive logic [10]. The concepts of a logical equation, a logical inequality and their systems are introduced.
RIL is one of the branches of formal logic. A special feature of RIL is that the representation of formulas is implemented using relational tables and SQL queries, and the processing of tables through the execution of queries is an act of inference.

We consider the basic terms and concepts used in RIL.

### 3.1 Relational table
Each relational table consists of columns (they are also called fields or attributes) and rows (they are called records or tuples). Relational tables must have a number of properties [11]. Relational tables are an integral part of the tools of relational logic.

### 3.2 Field term
The field term is used in the construction of logical formulas and exists solely in the context of RIL. On the one hand, a term denotes a variable as a part of a logical formula, and, on the other hand, it is used in the construction of a relational query. With the help of a relational term, reference is made to relational tables and their fields. A term is a construction of the form \([T].[F]\), where \(T\) is the table identifier, and \(F\) is the field identifier.

### 3.3 Constant
The constant specifies some fixed value. This value can be in the format of a text string, numbers, dates, etc. Constants are used when creating formulas for comparison operations, when restrictions are imposed on the field values of relational tables. The constant is a special case of a relational term and is a table of one field and one row.

### 3.4 Formula
We consider the concept of a logical formula. A formula can be atomic — indivisible or composed of atomic formulas. An example of the simplest atomic formula is the logical constants true and false. Atomic formulas are constructed using logical functions and comparison operators, from relational terms and constants. Based on atomic formulas, more complex formulas are constructed using logic connectives (NOT, AND, OR, IMP, XOR, EQV).

### 3.5 Logical equation
By a logical equation we mean a formula of the form: \([F1 \ EQV F2]\), where \(F1\) and \(F2\) are formulas, and \(\EQV\) means \(≡\) is an identity. Note that formulas \(F1\) and \(F2\) can, in turn, consist of other formulas, constants, and field terms. The logical equation sets the equivalence relation.

### 3.6 Logical inequality
By logical inequality we mean a formula of the form: \([F1 \ IMP F2]\), where \(F1\) and \(F2\) are formulas, and \(\IMP\) means \(⇒\) implication. A logical relation specifies the relations of consequence (derivability). In this case, the relation will be understood as a multiplace predicate or predicate property.

### 3.7 Systems of logical equations and inequalities
An ordinary system of logical equations and inequalities (SLEI) will be understood as a finite set of logical equations and inequalities united by the AND operator, which is equivalent to the logical conjunction operator \(∧\). The conjunction operation provides simultaneous fulfillment of conditions. Conditions are established by equations and inequalities in the system.

### 4. The formalized server model as a control system
Let us consider a formalized model of the server as CS. Note that the server can be viewed as a combination of hardware and software platforms. In order to create a consistent representation of the server as a control system, it is necessary to understand which parts of it we are interested in. As part of
our task, we will talk exclusively about the control of hardware parameters, such as, for example, processor temperature, voltage on the power supply unit, or the percentage of RAM load.

The server will be considered as a discrete set \( S = \{s_1, s_2, \ldots, s_n\} \) of its hardware parts where \( s_i \) is some device included in the server \((i = 1 \ldots n)\), \( n \) is a number of devices). It is worth noting that each device, in turn, may have a number of parameters that must be monitored. The set of parameters for all devices ultimately determines the current state of the server. In this case, it is convenient for us to move from considering the server as a whole to its elementary control systems.

For example, a processor can be considered as an elementary control system, which has a number of its parameters. These can be current temperature, power consumption and load percentage. Since we are not interested in the design and technical features of the device, we can represent it as a tuple \( s_i = \langle s_i^1, \ldots, s_i^j, \ldots \rangle \), where \( s_i^j \) is the specific device parameter. The tuple of these parameters can be considered as the scheme of the elementary CS (Fig. 1).

![CPU](image)

**Figure 1.** Representation of the processor as a tuple of parameters.

For the CS function we take the following function:

\[
F_i(s_i) = \begin{cases} 
1, & \forall s_i^j \in v_i^j, \\
0, & \exists s_i^j \not\in v_i^j, 
\end{cases}
\]

which takes the value 1 – true if all parameters are in the specified range and 0 – false if at least one of the components of the tuple deviates from the specified value (Fig. 2). Valid ranges are specified by a tuple \( v_i \), each component \( v_i^j \) contains the corresponding allowable interval. The coordinates of this elementary control system will be specified by a pair \( \langle s_i, t_k \rangle \), where \( t_k \) is the point in time at which the recording of device parameters occurs.

![Graph](image)

**Figure 2.** An example of the graph of the elementary CS state function.

In turn, the server as a set of control systems will be a set of tuples \( S = s_i \), each of which contains the parameters of the elementary systems being a part of the server (Fig 3.). For the server as a management system, we define the function \( P(S) = F_1 \wedge F_2 \wedge \ldots \wedge F_i \wedge F_{i+1} \wedge F_n \).

If the value of this function becomes false, it means that some of the server parameters deviated from the specified conditions.
Figure 3. Representation of the server as a set of tuples of parameters of elementary control systems.

Thus, we can say that the server state is characterized by a set of relations, where each relationship is an operation of comparing the components of a tuple with a certain numerical value. For example, let \( s^1_i \) be the temperature of the processor, then \( v_i = (35; 70) \). That is, CPU temperature cannot exceed 70 degrees and fall below 35 degrees.

5. The examples of control ratios

Having described the model of our control system, we can describe the control procedure with the help of RIL. To work with RIL, we need to translate the model of our system within the scope of RIL terminology and the relational data model.

As mentioned above, the main element of the formulas in the RIL is the field term. A term is a collection of a table and its field, for example, [Table_one]. [Field_one]. Note that in this case we can consider the table as a representation of a particular elementary system, and the field as one of the components of the tuple. Then, \( s_j \) will be represented by the corresponding table [Table_si], and \( s_j^i \) will be represented by its field [Field sj]. The rows of the table will represent the state of the elementary system at a particular point of time (Table 1).

| \( t_k \) | Temp | Percent_load | ... |
|---|---|---|---|
| 1 | 40 | 12 | - |
| 2 | 60 | 45 | - |
| 3 | 90 | 89 | - |
| 4 | 76 | 60 | - |
| ... | ... | ... | ... |

Hence for each of the tuples \( s_j \) we obtain a relational table. The structure of the table columns will coincide completely with the corresponding scheme of the elementary system. So, for example, for the tuple \( s_j \) we will create a CPU table that will contain the Temp, Percent_Load fields, and so on. When writing a specific relationship for specific parameters, we will use the corresponding relational term. We can store values of the tuple \( v_i \) containing sets of admissible values in separate tables; within the framework of RIL, they will act as constants in the construction of formulas.

Consider an example of a ratio that imposes a limit on the CPU temperature: [CPU].[Temp] > 35 AND [CPU].[Temp] < 70 EQV FALSE.

Using this relationship, we set a condition under which we consider incorrect all processor states under which the temperature is outside the range (35; 70). To describe the complete set of incorrect states
for the processor, we need to supplement the first condition with the rest. For example, we write the condition on the percentage of RAM usage: \([\text{CPU}.\text{Percent\_Load}] < 75\) EQV FALSE.

Next, combine the two ratios into one: ((CPU).[Temp] > 35 AND (CPU).[Temp] < 70) AND ((CPU).[Percent\_Load] < 75) EQV FALSE.

We have obtained a logical equation, in the left-hand side of which there are several relations connected by a logical AND. We used the logical AND, since we consider the server state to be correct, in which all its parameters are within the specified limits. To search the database for incorrect server states, we need to solve this equation, i.e. find the values of terms for which the left-hand side of the equation will become false. From the point of view of the RDM, we need to find in the database the lines for which the condition, written using the equation, would be fulfilled. To do this, based on the written equation, you need to create a SQL query to the database:

```sql
SELECT Temp, Percent_load
FROM CPU
WHERE ([CPU].Temp > 35 AND [CPU].Temp < 70) AND ([CPU].Percent\_Load < 75) EQV FALSE
```

The expression after the SELECT statement indicates which fields we want to see in the selection, the FROM clause indicates which table we want to select, and the WHERE clause specifies the selection condition — in our case, this is a logical equation. Note that in the WHERE clause we can apply various operations and perform a selection from several tables at the same time.

If, after performing a query on a set of tables, a solution is found, i.e. some strings are returned, this will mean that there are records in the logs with incorrect parameter values, and our server has deviated from the correct state.

To write the general correctness condition for the entire server, we need to write the correctness conditions for each elementary control system (for brevity, we denote them as Cond_si) and combine them as a system of logical equations: ((Cond_s1 EQV TRUE) AND (Cond_s2 EQV TRUE) AND ... (Cond_si EQV TRUE)) EQV FALSE

It is important to note that this is a relational representation of the function discussed above.

If necessary, we can flexibly change the conditions of correctness, add new or delete existing elementary systems or their parameters, we only need to rewrite the SLEI.

Separately, it is worth noting that logical equations allow us to write fairly complex correctness conditions taking into account the context of external factors.

Imagine that we need to monitor the percentage of memory usage. Suppose that in the normal state of the server, this figure should not exceed 65%. It is also known that a daily backup procedure is performed automatically between 00:00 and 03:00, which simultaneously significantly increases the load on the server, in particular, on the RAM, increasing its load percentage [12]. In this case, exceeding the specified limit is permissible and is not a deviation from the normal state.

Using RIL, we can write a control ratio, which, in addition to the direct restriction on the indicator, will take into account some external context, under which the condition must be fulfilled or not fulfilled. In this case, that external condition is the current time.

We write this expression:

```sql
([RAM].[PERCENT\_LOAD] > 65) AND (Time() < 00:00 AND Time() > 03:00) EQV TRUE
```

Note that in this expression, in addition to relational terms and constants, we also used the utility statement, which returns the current time. When an expression is translated into a SQL query, this statement can be converted to a SQL CURTIME() statement, which returns the current time.

This example shows that we can not only directly control the parameter values, but also take into account various extra conditions. This is especially important when solving these problems in real conditions, since very often the surrounding context plays a significant role in assessing the correctness of the system state.
6. Conclusion

By the example of the task of analyzing the hardware logs of server equipment, this article has demonstrated the use of relational interactive logic in solving the problem of monitoring the control system.

The use of systems of logical equations and inequalities in describing the conditions of the correct state of the system makes it possible to quite flexibly consider various sets of conditions and relationships, both between parts of the system itself and between the system and external factors.

Automating the SLEI solution by means of a SQL query provides an efficient process of analyzing the CS.

The process of using RIL for tasks of this kind can be divided into four steps:

1. Preliminary analysis of the system. Identification of key parameters, characteristics and elementary subsystems.
2. Building a formalized model of CS. Depending on the task, a decision should be made on the form of representation of the scheme and function both for elementary subsystems and for the entire control system as a whole.
3. Record the formalized model of CS in terms of RIL. Formation of a set of relational terms and constants necessary for building formulas. Formation of the list of relational tables for elementary subsystems.
4. Record and decision SLEI. Analysis of the results.

It should be noted that hypothetical limitation of being applied for some classes of systems is one of the disadvantages of this method. For example, it is problematic to use this method for complex hierarchical systems with a large number of cyclic links. While implementing step 3, the question arises about the validity of the CS model transformations into the terms of the RIL.

At the moment, the authors set out to continue testing this method in other practical tasks, and also to consider the possibility of formalizing the process of converting a CS model to an RIL. In particular, it seems very promising to use this technique in analyzing the correctness of data in various automated accounting systems.

In addition, the project of the “Object Logical Design Tool” software application has been launched. It is assumed that this application will automatically generate the necessary list of relational terms, tables and constants in the interactive mode according to a predetermined scheme of the control system.

References

[1] Ali M, Bagchi S 2019 Probabilistic normed load monitoring in large scale distributed systems using mobile agents Future Generation Computer Systems 96 148-167
[2] Wang T, Xu J, Zhang W, Gu Z, Zhong H 2018 Self-adaptive cloud monitoring with online anomaly detection Future Generation Computer Systems 80 89-101
[3] Hamamoto A H, Carvalho L F, Sampaio L D, Abrão T, Proença M L 2018 Network Anomaly
Detection System using Genetic Algorithm and Fuzzy Logic. *Expert Systems with Applications* 92, 390-402.

[4] Nadezhnin A. 2013. Datacenter Russian Railways. [Online] Available: https://habr.com/ru/post/204122/ (May, 2019).

[5] Sapoznikov V V, Efanov D V. 2015. Detection of dangerous errors in the working outputs of combinational logic circuits. *Automation on transport* 2195-211.

[6] Yablonsky S V. 1959. Basic concepts of cybernetics. *Problems of Cybernetics* 7, 39.

[7] Kuzmin O V, Seregina M V. 2010. Plane sections of the generalised Pascal pyramid and their interpretations. *Discrete Mathematics and Applications* 20, 377-389.

[8] Kuzmin O V, Leonova O V. 2000. Touchard polynomials and their applications. *Discrete Mathematics and Applications* 10, 391-402.

[9] Kim M, Stehr M-O, Talcott C. 2013. A distributed logic for Networked Cyber-Physical Systems. *Science of Computer Programming* 78, 2453-2467.

[10] Kurgansky V I. 2014. Relational interactive logic. From logical equations and inequalities to answers to difficult questions (Lambert Academic Publishing) p 124.

[11] Codd E F. 1970. Relational completeness of data base sublanguages. *Communications of the ACM* 13, 377-387.

[12] Anderson K. 2017. *Creating a Data-Driven Organization* (O’ Reilly) p 336.