



ISSN: 2350-0328

**International Journal of Advanced Research in Science,
Engineering and Technology**

Vol. 7, Issue 1 , January 2020

Software Solution Accuracy Automatically Testing System and Its Features

Raxmatova Baxtigul Axatovna, Nasilloev Sanjar Baxtiyor o'g'li.

Assistant of Department "Technology and Fabric Design" of Bukhara Engineering and Technology Institute, Bukhara, Uzbekistan.

Student, Tashkent University of Information Technology named after Muhammad al-Khwarizmi, Tashkent, Uzbekistan.

ABSTRACT: This article shows the general characteristics of an automated system, the general structure of the system, and the algorithm for working with teachers to evaluate students. In addition, the areas of application of this automated system are described directly.

KEY WORDS: User Interface, users, database, solution block, testing, initialization.

I. INTRODUCTION

At present, the Association for Computing Machinery (ACM) has become increasingly popular among students of universities in the programming Olympiad, which is held under the protection of the International Computer Society. It exists since 1947 year. This organization is engaged in the development and development of the skills of professionals and students in the field of Information Technology. Since 2002, the teams of Uzbekistan have also participated in this championship. Enough success has been achieved over the past years. In this case, the results of students and students are rising from year to year. Many universities are holding internal (local) championships in programming. This serves as a preparation for the inter-team World Cup in programming. In the process of preparation, teachers are faced with such questions as the development of software for holding competitions, the organization of preparatory classes and the use of it on the Internet. Carrying out preparatory work with the use of such verification systems creates the ground for taking high places.

II. THE MAIN PART

The main goal of the development of automatic testing systems for software solutions is to create the opportunity to organize the process of preparing students and students for the Olympiad and various competitions in programming. But this is not the only field of use of the system. The control automated system can be used to analyze the results of scientific research, evaluate them and also in the process of simple learning. It is possible to use the services of a system that automates verification, when the teacher does not have the opportunity to check the knowledge of the students in a short period of time. In this case, in order to check the solutions in advance, it will only be required to check them automatically with the help of pre-prepared tests. Preparation of tests is a much more complicated process, which requires very high skill. In the tests, there should be no errors in the examination of the issue. But once prepared tests can be used as long as they are desired. Therefore, the teacher can also use the tests of others without preparing his own tests. But for this you must of course collect a database of issues with a set of tests that are more than enough. In fact, collecting a database of such issues is times with the development of a course of Informatics, and this work is not an easy task.

Checking the knowledge of students who have already started to learn programming from the Informatics course is one of the most difficult issues (correcting and Testing written programs). The next examination is to verify the correctness of the performance of the assignments given by this teacher. This is also a difficult issue and requires further development of the assessment methodology. When examining the solution of theoretical issues, the main attention is paid to the analysis of the solution and the correctness of the response received, while in practical matters the situation is the opposite. Checking the correctness of the program, despite the importance of work, but also the timely implementation of the development of an algorithm for solving the given issue in them is required. One of the last stages of a written software solution is testing it. It is to examine the solution using a specially developed set of

tests for this issue. One of its convenient aspects is the assignment of its incoming and outgoing parameters to the condition of the issue. This will help the reader to verify that the issue is going through properly.

Recognizing the effectiveness and effectiveness of the work of testing the program, teachers believe that the results of testing will never be able to accurately assess the quality of the program and its student.

However, for many years, testing has been used around the world as a tool for assessing solutions to various problems, leading to improvements in test systems and the emergence of new computer technologies that meet the requirements of modern programming methodologies. Such testing systems can be useful for teachers. Because it is always better to look for errors in a student's program listing than to check for correctness.

III. GENERAL CHARACTERISTICS OF THE SYSTEM.

If the process of testing the application is carried out intuitively and without a clear plan, then this process will be called artificial. If the testing process is carried out on control examples with pre-thoroughly prepared data, if the testing process is carried out sequentially and on time, then it is possible to take testing as a science.

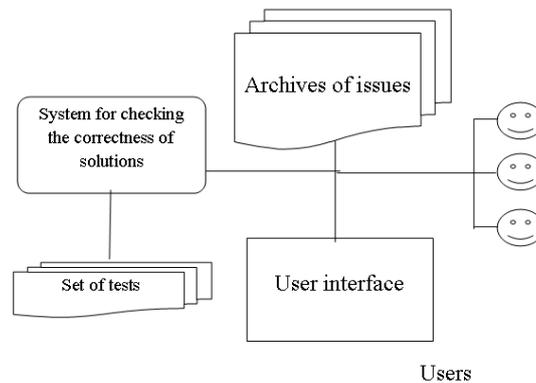


Fig 1. Automatic system scheme checking software solution accuracy

It will be necessary to pass the test of the training program, in order to check its proper performance in all possible cases. It is also designed to evaluate its effectiveness. In this case, based on the teacher's point of view, the student's program is considered a "black box". In order to carry out the verification of such a program without the participation of the reader, it is necessary to put the terms of the issue clearly and completely.

In Figure 1, we will consider an automatic system scheme that checks the correctness of the software solution:

The basis of an automatic system that checks the correctness of a software solution is a block that checks the correctness of the solution. This is the issue of formalization and verification of the software solution that is sent to the system by the user. Using a convenient graphical interface, the user needs to select the issue to be resolved from the archive of the issues. Chooses which programming language to write the solution. And then, using the same graphical interface, it sends the issue resolution in the form of a file, or its initial kodini resolution, to the system that verifies the correctness. The system that verifies the solution performs the following steps to determine the result: compiles the initial kodini of the program that corresponds to the programming language that the user chooses, executes the file received using the pre-prepared test package and analyzes the results. The user gets the verification result after a few minutes or even seconds after sending the solution.

While the verification system provides everyone with equal opportunities and criteria of authenticity, it requires participants to look carefully at the terms of the issue. Especially it is necessary to pay attention to the format of incoming and outgoing data. To participants who are familiar with such requirements, this does not cause difficulties. Unfortunately, schoolchildren fall into such a state of neglect or lack of a clear approach to the terms of the issue. It is possible to introduce invariable rules of programming in production areas, especially in case of compliance with the requirements of the customer. An optional professional programmer should know how to work with a clear approach to the specification. The extreme can impose a single restriction on the system that verifies this solution. But as accepted in international practice, there should not be assembly inserts in the program code.

IV. THE GENERAL STRUCTURE OF THE SYSTEM.

It is possible to determine the scheme of operation of the system based on the general requirements imposed on it before the implementation of a system that verifies software solutions(fig 2.). It should be noted that the overall scheme of the system consists of three main blocks: a system that verifies Solutions, a system that stores data and an interface that interacts with an automated system. Summarizing it all, it can also be called an automated system that checks the correctness of the software solution.

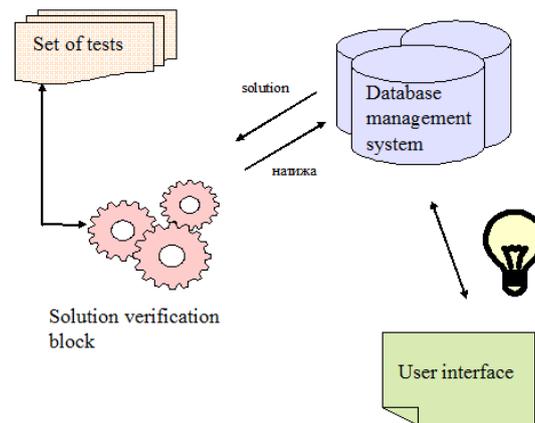


Fig 2. Overview of the testing system

The user who wants to check their knowledge through the WEB interface will select the desired issue from the archive. The user must write the chosen issue in the programming language that supports the system that tests the software solution. After preparing the software solution of the issue, the user enters the software solution of the issue by selecting the programming language and the issue identifier(name)using a special dispatch form and clicks the "Otravit" ("Send") button. The software solution code sent by the user is stored in a special folder on the file server under a unique identifier. And this identifier determines the software code all its data in the database at the same time, that is, the solution is written in what language it is written, by whom, when it is sent, etc.the G.). After that, the code of the user's software solution passes to the line that reflects the results of the verification and is checked with the help of tests. It is usually enough to pass 10-15 tests to verify that the program works correctly on the solution. The participant knows the results of the check in a few seconds after sending the dasturiy software solution of the issue.

It is also possible that the main components of the automated system (the system that verifies the solution, the database management system and the user interface) can be located on absolutely different servers. This makes the operation of the automated system more convenient.

An automated system that verifies the accuracy of software solutions can be used on optional local area networks built on IP/IP protocols.

V. THE SOFTWARE SOLUTION IS A SYSTEM PERFORMANCE ALGORITHM THAT AUTOMATICALLY TESTS THE ACCURACY.

The main goal from the creation of a complex automated system is to decompose (turn out) a given issue into several tiny and light Part issues. It is necessary to design the process of system operation in such a way as to the modules that are logically separated from each other.

Let's see in Figure 3 the general algorithm of the operation of the software solution controller system.

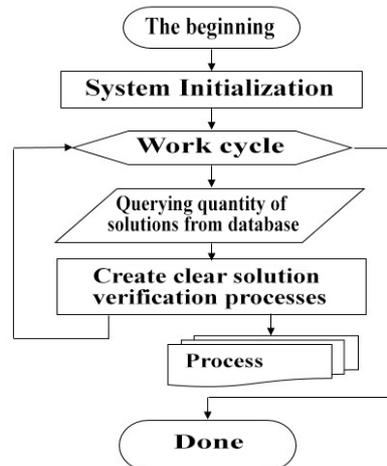


Fig 3. General algorithm of software solutions controller system performance

1. The configuration of the software solution sent at the stage of initialization of the system is determined. The result is determined all the necessary and necessary values of the software solution kodini. For example, the parameters associated with the database, the programming language, the ways of loading compilers corresponding to programming languages, etc.the G.. Also at this stage, the process of connecting to the database server is carried out. If there is some kind of error at the stage of initiation, the system should stop its work. In this case, all data on the time and status of the error generated error will be stored in the log file (a file containing a detailed report on the performance of the system).

2. After the stage of initiation, the main cycle of work begins. This phase is also called the basic life cycle of a software solution verification system. This will stop its work as soon as the system comes out of the cycle. The exit status of the cycle can occur for two reasons: the condition of stopping the program in case of an accident, as well as a critical unexpected execution error. A description of these error cases is quickly placed in the report file.

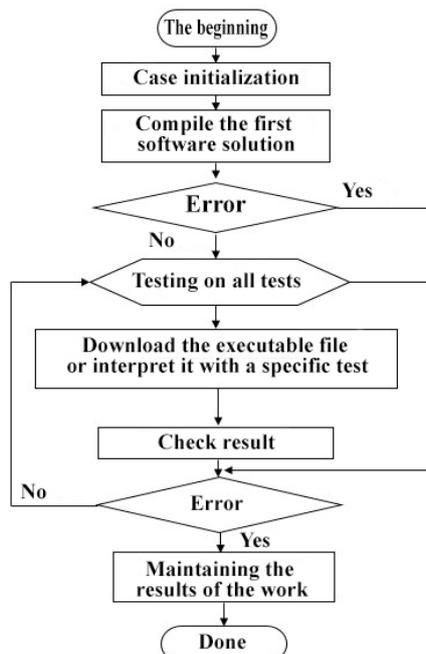


Fig 4. Software solutions controller module performance algorithm

Until the testing process is complete, verification to check for new issues is a regular reference to the database during system operation. It therefore determines the number of idle processes that design issues to load issues in this process. If all processes are busy, the next punk of the work will be missed and the system will wait for at least one

process to finish, that is, 2 will return to the Punkt. Also, if there are no software solutions that can be checked in the database, the system returns to the 2 Punkt of the algorithm.

1. If there are solutions that require verification in the system – then they load the verification process into another stream. That is, a separate environment is created, which does not depend on the process of running the system. It is necessary to consider separately the performance of each process.

2. The performance cycle is completed and a new cycle begins. The software solution verification system will be in such a state throughout its entire life cycle.

In the process of the entire life cycle of the control system, the control module performs its main functions directly software solution. The algorithm of this module is shown in Figure 4.

We will consider in detail the stages of operation of the module of direct inspection of the solution.

VI. CONCLUSION AND FUTURE WORK

The system that automatically tests software solutions is also free from defects. This is only in his analysis of solutions on a test basis.

In general, the development of the system is aimed at solving issues that are relevant. At the next stage, we will continue our research on software solutionsualual analysis system, which will eliminate the shortcomings of the automatic testing system and determine the exact correctness of the result.

REFERENCES

1. Basics of algorithms and programming: a study guide for colleges / A.R Azamatov; Ministry of Higher and secondary education of the Republic of Uzbekistan, Center for secondary special, vocational education. Fourth edition. — T:Chulpon named publishing house-printing creative house, 2013.- 232p.
2. Yusupbekov N.R and others. Technological process management systems archives-project topics - Tashkent: Teacher. It's 1997.-704 p.
3. Yusupbekov N.R , Muhammedov B.I , Gulomov Sh.M. Control and automation of technological processes. - Tashkent: Teacher. The 2011. - 576 p.
4. Lukas V. A. Theory of automatic control. - Moscow: Nedra, 1990. – 416 p.
5. O. Y. Ageeva. Mathematical logic and algorithm theory. Moscow, 2011.

AUTHOR'S BIOGRAPHY



Raxmatova Baxtigul Axatovna Assistant of Department “Technology and Fabric Design” of Bukhara Engineering and Technology Institute, Bukhara, Uzbekistan. Bukhara city, K.Murtazaev street 15



Nasilloev Sanjar Baxtiyor o'g'li Student, Tashkent University of Information Technology named after Muhammad al-Khwarizmi, Tashkent, Uzbekistan.