Automated skills control of food production specialists using a training system

Ivan S. Polevshchikov[1,2*], and *Rostislav* A. Kravchenko[1]

[1]Russian Biotechnological University, 11, Volokolamskoe shosse, Moscow, 125080, Russian Federation
[2]Moscow State University of Technology and Management named after K.G. Razumovsky (First Cossack University), 29, ulitsa Zemlyanoy Val, Moscow, 109004, Russian Federation

Abstract. The article is devoted to the development of information technologies for monitoring knowledge and skills in the training of engineering specialists in the field of food production. A methodology for monitoring the initial skills of these specialists has been developed based on a training and teaching subsystem. The technique consists of automatically generating and evaluating practical tasks using mathematical models of objects of professional activity. The use of an automated information system in the educational process, which includes implementing the specified training and education subsystem functions, will allow to reduce the teacher's time for preparing and checking assignments; increase the accuracy of assessment of assignments during control, reduce the subjectivity of assessment (compared to checking an assignment by a teacher); reduce the time a student waits for an assessment. The considered methodology will be implemented in the form of automated information system software modules for monitoring knowledge and skills during the training of specialists. The automated information system prototype is currently used at ROSBIOTECH and MSUTU n.a. K.G. Razumovsky when training students in areas of training in the field of automation, informatization, robotization of technological and business processes.

1 Introduction

In connection with the development of intelligent information technologies in all areas of activity (in particular, in the food industry), the rapid change in the socio-economic situation, and the solution of food security problems, an important task is the high-quality training of engineering personnel for food production (food production technologists, specialists in automation, informatization, robotization of technological and business processes).

The fulfillment of this task is largely facilitated by such a trend in modern education as the development and use of automated educational process management systems. Among the scientific works relevant to the problems of development and application of these systems in training the above specialists, one can highlight works in the field of e-learning and distance

[*] Corresponding author: i.s.polevshchikov@gmail.com

technologies [1-3], automation of assessment of knowledge and skills [4-6], modeling of processes of knowledge and skills formation [7], creation of virtual laboratories [8-10].

To date, a prototype of an automated information system (AIS) has been created to control knowledge and skills in the training of specialists in automation, informatization, and robotization. This prototype is implemented as a web application and partially implemented in ROSBIOTECH and MSUTU named after K.G. Razumovsky when teaching students in the relevant fields [11]. The main function of the AIS prototype at the moment is the control of knowledge and skills through remote checking by the teacher of work (laboratory, practical, tests, etc.) completed by students and sent for verification through the AIS.

Previously, the works [11, 12] described approaches to automating the control of knowledge and skills in vocational training, allowing to minimize the labor-intensive manual work of the teacher in preparing and checking practical assignments. An urgent task, the results of which are presented below, is the development of these methods and means of automating the control of knowledge and initial skills based on modeling objects of professional activity in the educational process.

2 Methodology for monitoring initial skills using a training system

A method for monitoring the initial skills of engineering specialists in the field of food production is proposed through the automatic generation and evaluation of small practical tasks. Such tasks can be used, for example, when conducting tests using the training and teaching subsystem (TTS) of the AIS (in particular, the test can include test tasks to test theoretical knowledge, as well as one or more practical tasks), to protect laboratory reports and practical work.

One of the objects of professional activity of the above specialists (in particular, specialists in automation, information, robotization of production processes) is algorithmic and software for automated information processing and control systems. During the control of knowledge and skills, tasks may be provided: on compiling program code; on software code analysis (in particular, when testing software modules); on building visual models that describe the structure and operating algorithms of software systems.

This technique is based on the clarification, systematization and development of a number of studies [11, 12]. As examples, consider the tasks:

- on the analysis and construction of a flow graph [12] when studying base path testing as one of the methods for testing software modules of automated control systems (ACS) for production processes;
- on the analysis of UML use case diagrams used to visually represent requirements for automated control system software.

The formation and evaluation of such tasks can be formalized and automated to a large extent.

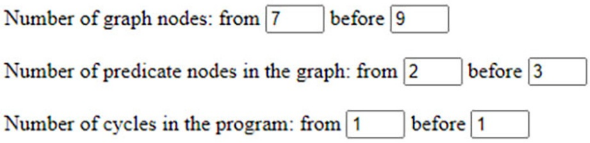
Let us present this technique in a sequence of steps. At step 1, the teacher develops a practical task, which includes the following components:

1) Setting up options for completing the task. An effective solution is to customize a task template, which includes setting values for a set of object properties that the student needs to analyze or construct. For example, in a task on analyzing a flow graph, the teacher sets the values of some properties of the graph (total number of nodes, number of predicate nodes, etc.), based on which an individual version of the graph will be automatically generated when the student completes the task.
2) Definition of individual subtasks that are part of the task. For example, in a task on flow graph analysis, the student may be offered subtasks to determine the specified characteristics of the flow graph (for example, “number of predicate nodes”,

"cyclomatic complexity", "upper bound on the number of test cases"), as well as theoretical test questions on the given topic (for example, knowledge of basic concepts, formulas, facts on the base path testing method).

3) Setting up a palette of elements for tasks that involve building an object (in particular, a visual model). In the process of setting up the task, the teacher creates a palette of elements that the student can then use when completing the task of constructing some object (in particular, a visual model, an example of which is a flow graph). The set of valid elements for forming a palette should be stored and replenished over time in the AIS database. Depending on the level of complexity of the task, the palette may differ within the same topic being studied (the more palette elements and settings of their properties are available to the student, the more complex the task).
4) Setting up parameters for assessing the completed task. The weight of the task is determined (relative to other possible tasks within, for example, a test). A time limit for completing a task can be set. For each subtask included in the task, a weight is assigned that reflects the complexity and importance of this subtask when assessing the completed task. Additional settings are acceptable, for example, selecting one of the possible assessment algorithms, differing in the degree of accuracy and detail.

Let us look at the layouts of the developed AIS web interfaces with examples of setting up some tasks. Figure 1 shows the setup of a flow graph template for a task to analyze it (an individual version of the graph will be generated based on the settings).



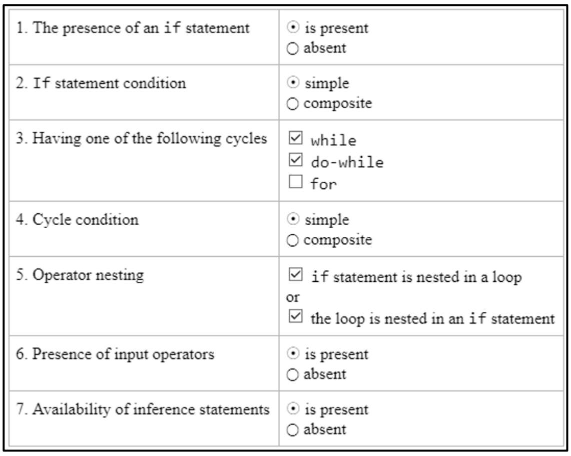
Number of graph nodes: from 7 before 9

Number of predicate nodes in the graph: from 2 before 3

Number of cycles in the program: from 1 before 1

Fig. 1. Setting up a flow graph template.

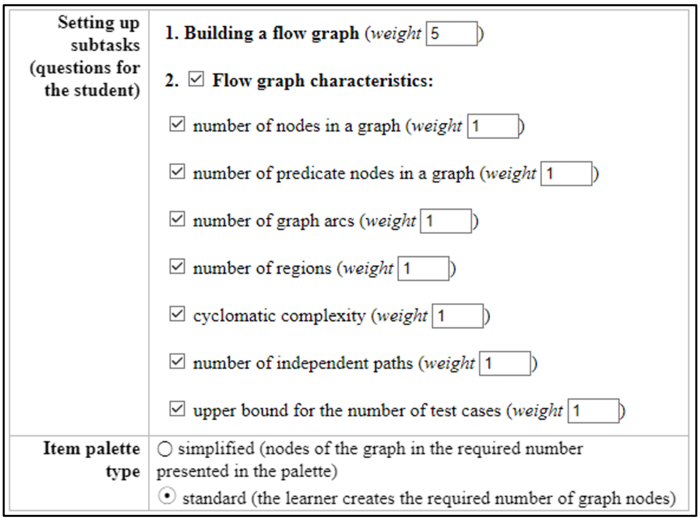
Figure 2 shows the setup of a program code template in Java for the task of constructing a flow graph based on this code (according to the settings, an individual version of the program code will be generated).



1. The presence of an `if` statement	● is present ○ absent
2. `If` statement condition	● simple ○ composite
3. Having one of the following cycles	☑ `while` ☑ `do-while` □ `for`
4. Cycle condition	● simple ○ composite
5. Operator nesting	☑ `if` statement is nested in a loop or ☑ the loop is nested in an `if` statement
6. Presence of input operators	● is present ○ absent
7. Availability of inference statements	● is present ○ absent

Fig. 2. Setting up a code template.

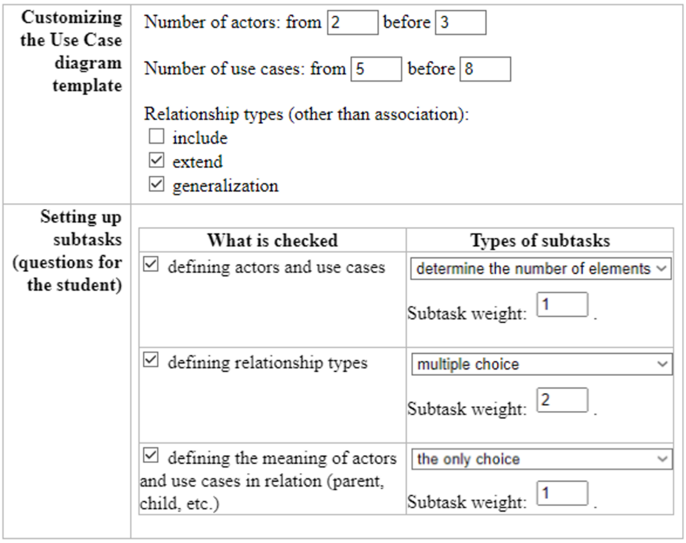
Figure 3 shows the setup of the subtasks that are part of the task of constructing and analyzing a flow graph (the setup includes selecting the required subtasks and indicating their weights for evaluation). Figure 3 also shows the choice of the type of palette of elements for constructing a graph (the level of complexity of the task depends on the selected type of palette).



Setting up subtasks (questions for the student)	**1. Building a flow graph** (*weight* 5) **2.** ☑ **Flow graph characteristics:** ☑ number of nodes in a graph (*weight* 1) ☑ number of predicate nodes in a graph (*weight* 1) ☑ number of graph arcs (*weight* 1) ☑ number of regions (*weight* 1) ☑ cyclomatic complexity (*weight* 1) ☑ number of independent paths (*weight* 1) ☑ upper bound for the number of test cases (*weight* 1)
Item palette type	○ simplified (nodes of the graph in the required number presented in the palette) ● standard (the learner creates the required number of graph nodes)

Fig. 3. Setting up subtasks for the task of constructing and analyzing a flow graph and choosing the type of element palette.

Figure 4 shows the setup of a UML diagram template and test questions for a student to analyze this diagram. The student must answer a list of automatically generated questions based on the individual diagram provided.



Customizing the Use Case diagram template	Number of actors: from 2 before 3 Number of use cases: from 5 before 8 Relationship types (other than association): □ include ☑ extend ☑ generalization
Setting up subtasks (questions for the student)	

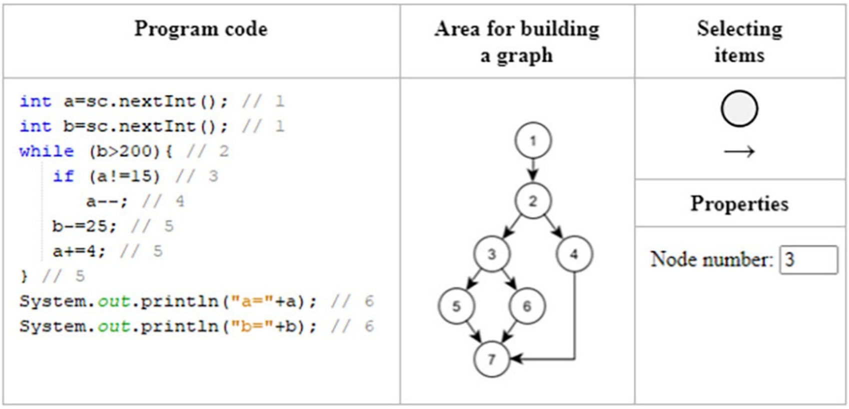
What is checked	**Types of subtasks**
☑ defining actors and use cases	determine the number of elements Subtask weight: 1 .
☑ defining relationship types	multiple choice Subtask weight: 2 .
☑ defining the meaning of actors and use cases in relation (parent, child, etc.)	the only choice Subtask weight: 1 .

Fig. 4. Setting up a UML diagram template and student diagram analysis questions.

https://doi.org/10.1051/e3sconf/202344306011

At step 2 the student completes an individual version of the task, formed based on the teacher's settings. The student constructs some object and/or answers a series of test questions.

Figure 5 shows the layout of the web interface for the student to complete the task of constructing a flow graph based on program code (in Java) using a special graphic editor. The program code was generated according to the teacher's settings (Figure 2). To construct a graph, the student has access to a drawing area, a palette of elements (in this example, it includes two elements - a node and an arc of the graph), and the ability to set the number of the selected graph node in the properties.

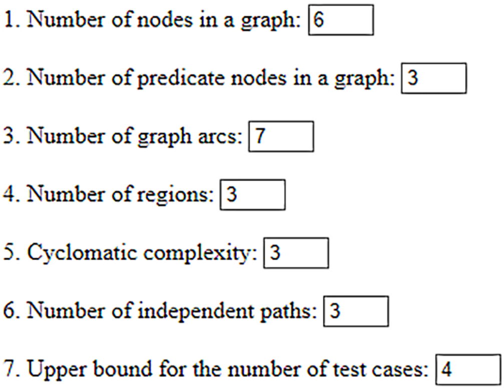


```
int a=sc.nextInt(); // 1
int b=sc.nextInt(); // 1
while (b>200){ // 2
  if (a!=15) // 3
    a--; // 4
  b-=25; // 5
  a+=4; // 5
} // 5
System.out.println("a="+a); // 6
System.out.println("b="+b); // 6
```

Node number: 3

Fig. 5. The student completes the task of constructing a flow graph.

Figure 6 shows the layout of the web interface for the student to complete the task of analyzing a flow graph. The student must determine the characteristics of the graph by filling out the text boxes.



1. Number of nodes in a graph: 6
2. Number of predicate nodes in a graph: 3
3. Number of graph arcs: 7
4. Number of regions: 3
5. Cyclomatic complexity: 3
6. Number of independent paths: 3
7. Upper bound for the number of test cases: 4

Fig. 6. The student performs a task on analyzing a flow graph.

Figure 7 shows an example of an individual version of a Use Case UML diagram displayed in the student's web interface to answer questions about analyzing the diagram. This version of the diagram was generated according to the teacher's settings.

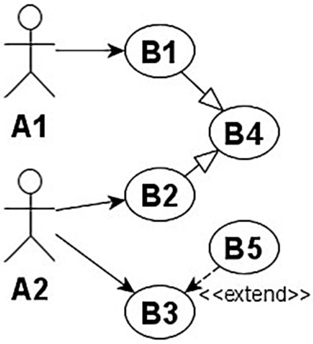
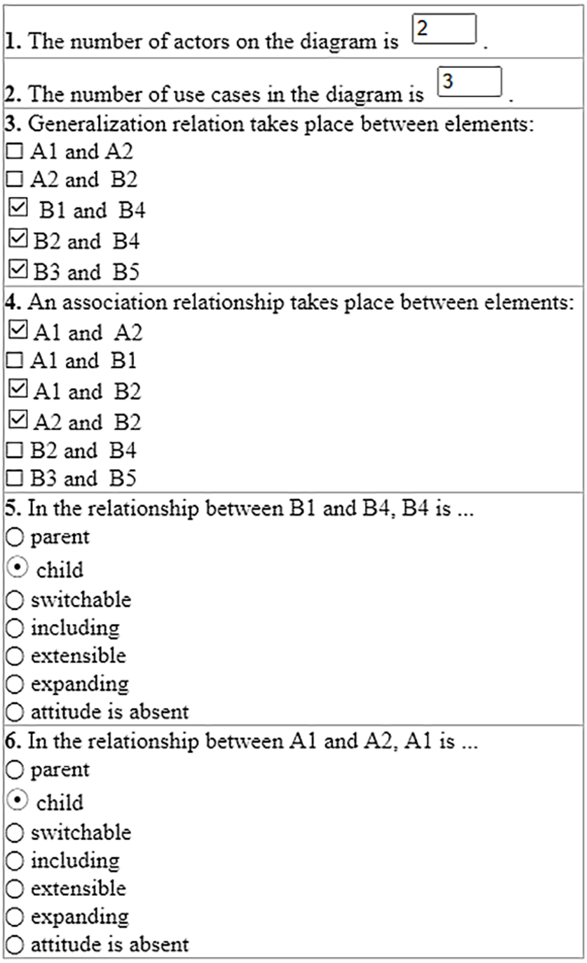


Fig. 7. Example of a custom Use Case UML diagram.

Figure 8 shows the layout of the student's web interface for answering questions on the analysis of an individual version of the Use Case UML diagram. The list of questions was generated according to the teacher's settings (Figure 4).



1. The number of actors on the diagram is 2 .

2. The number of use cases in the diagram is 3 .

3. Generalization relation takes place between elements:
□ A1 and A2
□ A2 and B2
☑ B1 and B4
☑ B2 and B4
☑ B3 and B5

4. An association relationship takes place between elements:
☑ A1 and A2
□ A1 and B1
☑ A1 and B2
☑ A2 and B2
□ B2 and B4
□ B3 and B5

5. In the relationship between B1 and B4, B4 is ...
○ parent
● child
○ switchable
○ including
○ extensible
○ expanding
○ attitude is absent

6. In the relationship between A1 and A2, A1 is ...
○ parent
● child
○ switchable
○ including
○ extensible
○ expanding
○ attitude is absent

Fig. 8. Example Questions on Use Case UML Diagram Analysis.

At step 3, an automatic assessment of the task solved by the student is carried out. The assessment of the task solved by the student is carried out according to the equation:

$$K=\sum_{i=1}^{N} w_i K_i \Big/ \sum_{i=1}^{N} w_i ,$$

where w_i is the weight of the i-th subtask (N - total subtasks), reflecting its importance and/or complexity within the task; $K_i \in [0;1]$ – evaluation of the i-th subtask (the closer the value K_i is to 1, the higher the correctness of the solution to the subtask).

Various algorithms for calculating grades are acceptable, aimed at establishing full or partial correspondence between the correct solution to the task and the student's solution. When assessing a practical task (or a subtask within the task) to construct an object, two mathematical (usually set-theoretic) models are compared, namely the model of the task solution prepared by the student and the model of the reference (correct) solution (an example of a model is the representation of arcs and vertices flow graph in the form of a set). The correct model is determined automatically based on the settings or set by the teacher (for example, a flow graph is determined automatically based on the program code).

The considered examples of generating and assessing practical tasks K_i can be generalized for various types of exercises with the aim of training specialists in the food industry to construct and analyze mathematical models of objects and processes in the form of graphs.

The use of AIS based on this methodology in the educational process will allow:

- to reduce the teacher's time for preparing and checking assignments;
- to increase the accuracy of assessment of assignments during control, reduce the subjectivity of assessment (compared to checking an assignment by a teacher);
- to reduce the time a student waits for an assessment.

It should be noted that it is possible to automatically check only small practical tasks, the evaluation process of which can be completely formalized. Checking many tasks for a number of types of control is impossible without the participation of a teacher. These types of control include checking reports on laboratory work, coursework, and final qualifying papers.

3 Conclusion

A methodology has been developed for monitoring the initial skills of engineering specialists in the field of food production based on a training and teaching subsystem. The technique consists of automatically generating and evaluating practical tasks using mathematical models of objects of professional activity.

The use of AIS in the educational process, which includes implementing the specified TTS functions, will allow: to reduce the teacher's time for preparing and checking assignments; increase the accuracy of assessment of assignments during control, reduce the subjectivity of assessment (compared to checking an assignment by a teacher); reduce the time a student waits for an assessment.

The methodology considered will be implemented in the form of AIS software modules for monitoring knowledge and skills during the training of specialists. The AIS prototype is currently used at ROSBIOTECH and MSUTU named after K.G. Razumovsky when training students in areas of training in the field of automation, informatization, robotization of technological and business processes.

As a further development of research, it is planned to make the considered methods and corresponding AIS software modules more universal, taking into account the specific features of most subject areas (according to the disciplines studied) when forming and assessing the correctness of the student's performance of practical tasks.

The study was carried out with financial support from the Russian Science Foundation within the framework of scientific project No. 23-79-10162.

References

1. L.S. Lisitsyna, M.S. Senchilo, E.A. Efimchik, Smart Innovation, Systems and Technologies **188,** 171-180 (2020)
2. A.P. Gagarin, E.V. Ivanov, Modern information technology and IT education **13(3)**, 61-71 (2017)
3. D. Bouhnik, G. Carmi, Journal of Information Technology Education: Research **11,** 201-219 (2012)
4. R. Babo, L. V. Babo, J. T. Suhonen, M. Tukiainen, Journal of Information Technology Education: Innovations in Practice **19**, 001-029 (2020)
5. A. Gero, Y. Stav, I. Wertheim, A. Epstein, Global Journal of Engineering Education **21(2),** 139-144 (2019)
6. Z. Kovacic, J. Green, Journal of Information Technology Education: Innovations in Practice **11,** 53-70 (2012)
7. E.A. Nazoykin, A.V. Ionov, A.S. Nosenko, I.G. Blagoveshchenskiy, Food industry **6,** 36-39 (2015)
8. J. Ramírez-Romero, D. Rodriguez, S. Rivera, Global Journal of Engineering Education **22(2),** 123-129 (2020)
9. M. Alkhedher, O. Mohamad and M. Alavi, Global Journal of Engineering Education **23(1)**, 55-61 (2021)
10. Yi-Han Liu, Yu-Tong He, Di Tian, Run-Long Fan, Li Yao, Computer Applications in Engineering Education **26(5),** 22053 (2018)
11. M.M. Blagoveshchenskaya, I.S. Polevshchikov, E.R. Ponasenkov, R.A. Kravchenko, AIP Conference Proceedings **2627,** 070001 (2023)
12. I.S. Polevshchikov, E.B. Krokha, Automated control of initial skills in the training of IT specialists, Informatization of education and e-learning methods: digital technologies in education: materials of the V International Scientific. conf., (Krasnoyarsk, September 21–24, 2021) pp. 270-275