Modeling and simulation of real process – passing through the labyrinth as a method of development of algorithm thinking and programming skills

S. Hubalovsky

***Abstract*—** One of the primary aims of teaching algorithm development and programming at the secondary school is to give students not only practical skills but also the theoretical knowledge and methods to solve various problems. In this context, the authors emphasize the importance of the problematic of problem analysis as well as its implementation in the teaching even before the basic principles of compiling algorithm that solves a specific problem. The whole process of problem analysis, including its solution, is demonstrated on the example of a robot model passing through the labyrinth. In this paper, problem analysis is implemented in question of labyrinth form, design of robot model and finally dealing with potential states during passage through the maze. This work is finished by sample of solution of the problem using the flowchart.

***Keywords*—** Algorithm Development, Algorithmic Thinking, Labyrinth, Lego® Mindstorms®, Modeling and simulation, Multidisciplinary Approach.

I. Introduction

THERE are many things in the world, which people want to express. If the things are static, the expression is simple. But, most of those change their state in the time. Some rules or patterns can make the changes and people want to express them in the well-arranged and understandable form [1].

The ability to represent the processes and actions in such form and the ability to understand them is the aim of the education in algorithms and algorithm development.

Courses of the algorithm development and programming are currently provided in the Czech Republic at secondary schools, technical colleges and universities in the subjects focusing on informatics. The ability to create algorithms develops logical thinking skills and imagination and is an inseparable part of study skills of prospective and undergraduate teachers specializing in "Informatics" at the Faculty of Education.

Algorithms are encountered in all practical activities without being realized. An algorithm generally involves defining the rules and giving the sequences of steps necessary for any activities. The most common examples of algorithms in everyday life can be found e.g. in [2], [3].

The expression of the algorithms in the form of schemes and figures - flowcharts can be hardly understandable for the beginners. An abundance of particular shapes can be confusing.

The term algorithm can be also connected with the terms like *system*, *model*, simulation, *multidisciplinary approach*, which are important in current approach to scientific, technological and professional practice. Many universities are realizing that modeling and simulation is becoming an important tool in finding the strategy for solving and understanding numerous and diverse problems. The examples can be found e.g. in [4], [5], [6].

In this paper we introduce a case study illustrating step by step process problem solution, strategy and algorithm development and computer simulation of a real process – *pass through the labyrinth*. Computer simulation of this process will be presented by robotics system LEGO® MINDSTORMS® programmed in LEGO® MINDSTORMS® NXT-G and NXC languages.

II. Modeling and simulation of the systems and processes

A. Modeling

Modeling is a method that is often used in professional and scientific practice in many fields of human activity.

The main goal of modeling is not only describing the content, structure and behavior of the real system representing a part of the reality but also describing the processes.

The process can be understood as series of transformations that changes the input values to output values. From the system point of view the process is dynamic system in which the values of the characteristic of the system elements are changed under the influence of the external elements.

The models are always only approaching of the reality, because the real systems are usually more complex than the models are. The system homomorphism is applied in the process of modeling, which means that each element and

Stepan Hubalovsky is working at University of Hradec Kralove, Department of informatics, Faculty of Science, Hradec Kralove 500 38, Rokitanskeho 62, Czech republic, stepan.hubalovsky@uhk.cz.

interaction between the elements of the model corresponds to one element and interaction of the modeled real system or real process, but the reverse is not true. The model is always to be understood as simplification of the original. If the relation of isomorphism is between the model and real system the original model we could not distinguish between the model and the original, which is discussed e.g. in [7].

The first step in the process of computer simulation is creation of conceptual model of the studied real system / real process. Conceptual model can be represented in different way. The most used representations are:

- Mathematical equitation;
- Process charts.

Mathematical equitations establishes mathematical model of the studied real system. The model can be obtained either theoretically based on basic physical properties of the system, or numerically by means of the measured values. Determination of parameters of theoretical model developed from empirical data is called system identification.

Process charts establishes process model of the real process. The process models can be described by different way; the most common are flowcharts that described the algorithm of the modeled process.

The conceptual model must adequately describe the dependency system outputs on its inputs. Models of real process system will be shown in the following paragraphs of this paper.

B. Simulation

The process of modeling is closely related to the simulation. Simulation can be understood as process of executing the model. Simulation enables representation of the modeled real system or real process and its behavior in real time by means of computer or other ICT tool (e.g. LEGO Mindstrom). The simulation enables also visualization and editing of the model.

A typical simulation model can be written both through specialized programming languages that were designed specifically for the requirements of simulations, or the simulation model can be created in standard programming languages.

From the above considerations, it is clear that simulation is a process that runs on the computer. In some publications, therefore, can be found the term "computer simulation". It generally is valid that computer simulation is a computer-implemented method used for exploring, testing and analysis of properties of the conceptual (mathematical or process) models that describe the behavior of the real systems or real process which cannot be solved using standard analytical tools, se e.g. [8].

The simulation models represented by executable computer program have to be isomorphic with the conceptual model that is a representation. It means that the mathematical model and simulation model have to represent the real system, its elements, internal interactions and external interaction with the environment in the same way.

In our paper the real process *pass through the labyrinth* and computer simulation of this process will be presented *by robotics system LEGO® MINDSTORMS® programmed in LEGO® MINDSTORMS® NXT-G and NXC languages*..

C. Significant function of the simulation

Simulation has from the scientific point of view several functions – see e.g. [8].

We will focus in this paper two of them and they are:

- replacing the real process;
- development of educational process.

1) Replacement of the real process

This is an important and indispensable feature of simulations and simulation model because it allows realize a situation of the process that cannot be investigated conventionally. The main advantage of simulations is that simulations model allows providing rather big number of the process steps in relatively short time, changing of input parameters and its visualization and optimization of the process.

2) Development educational process

The simulation is very useful from educational point of view. Using the simulation model and visualization of simulation results on the screen, students can better understand the basic features of the processes and systems and develop their intuition. It is also essential that the teaching by means of simulation is much cheaper and faster than the teaching carried by real experiment. In some cases providing the real experiment cannot be feasible.

D. Model verification and validation

Verification and validation are important aspects of the process modeling and simulation. They are essential prerequisites to the credible and reliable use of a model and its results [9].

1) Verification

In modeling and simulation, verification is typically defined as the process of determining if executable simulation model is consistent with its specification – e.g. conceptual model. Verification is also concerned with whether the model as designed will satisfy the requirements of the intended application. Verification is concerned with transformational accuracy, i.e., it takes into account simplifying assumptions executable simulation model. Typical questions to be answered during verification are:

- Does the program code of the executable simulation model correctly implement the mathematical model?
- Does the simulation model satisfy the intended uses of the model?
- Does the executable model produce results when it is needed and in the required format?

2) Validation

In modeling and simulation, validation is the process of determining the degree to which the model is an accurate representation of the real system / real process. Validation is concerned with representational accuracy, i.e., that of representing the real system / real process in the conceptual model and the results produced by the executable simulation model. The process of validation assesses the accuracy of the models. The accuracy needed should be considered with respect to its intended uses, and differing degrees of required accuracy may be reflected in the methods used for validation. Typical questions to be answered during validation are:

- Is the mathematical model a correct representation of the real system?
- How close are the results produced by the simulation executable model to the behavior of the real system?
- Under what range of inputs are the model's results credible and useful?

Validation and verification are both ultimately activities that compare one thing to another. Validation compares real system / real process and conceptual model. Verification compares conceptual model and executable simulation model. Sometimes validation and verification are done simultaneously in one process.

Validation of the conceptual model as well as verification of the simulation model of our real process – *passing through the labyrinth* – are to be done simultaneously by running simulation computer model.

The whole process of transformation from a real system, the simulation model and its visualization is shown in Figure 1.

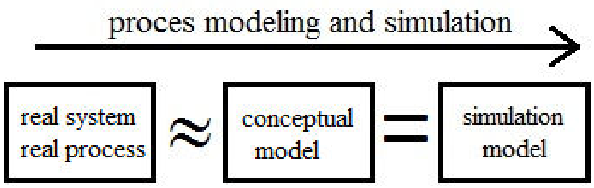

Fig. 1 Process modeling and simulation

Here again let us summarize that the mathematical model that reflects the real system / real process has some limitations and simplifying assumptions (the real system / process and conceptual model are in homomorphic relation).

In contrast, the simulation model is only the computer expression of the conceptual model (the conceptual model and simulation model are in isomorphic relationship).

E. Multidisciplinary approach

Another important benefit associated with the modeling and simulation of real processes is a multidisciplinary approach, without which the identification of the real processes using conceptual and simulation model and cannot be realized. This is also emphasized in this paper.

Multidisciplinary approach generally means that specialized disciplines are applied in a study of real process. These disciplines provide partial analysis of the process. These mono-disciplinary analyses are integrated to overall solution by integrating the solver who has basic multi-disciplines knowledge.

In our case study four disciplines are integrated, namely, algorithm development, programming and mathematics.

III. Problem Formulation
Process of Passing through the Labyrinth

Problems come in a variety of shapes and sizes. For example, in motoring, we can use the word 'problem' to cover: a car breakdown, high running costs or locking keys inside. All these could be termed problems, yet they are very different and require different approaches to deal with them. Some involve decisions. Some involve a whole range of problems from which priorities must be chosen. Some may not be completely soluble and may have to be coped with. There is no one way that will solve all problems. There are various approaches, or 'tools', which will help to solve certain types of problems. Most of them are only ordered common sense, but this is precisely what is lacking in many intuitive attempts to tackle problems [9].

Problem analysis should be convenient for solvers, so there are many tools and techniques to try to capture any given problem. According to [9], we can identify five general practices in problem analysis:

- Identifying priorities
- Analyzing symptoms to find causes
- Developing alternatives
- Decision making
- Follow up

Not all of these techniques must necessarily be used to analyze each problem. Their use is based on the characteristics of the problem, where it makes sense. Some of them are applied as part of the analysis in the below mentioned problem of passing through the labyrinth.

The solution of the process of the passing through the labyrinth from algorithm point of view is very popular – see e.g. [10]. Different variants of this problem provide interesting aspects not only in specific design of the robot, but also in the specific solutions within the labyrinth. In this article the specific problem of a robot constructed from the LEGO® MINDSTORMS®, which should be able to successfully complete the passing through the labyrinth.

The analysis of the problem includes the division identified the main problem into sub-problems [10]. This will allow easier and more convenient solution in case of the selected sub-problem than at the original main problem. The individual minor problems may have relations– see Figure 2.

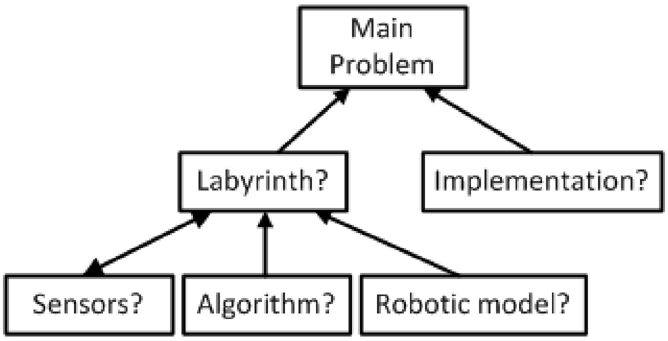

Fig. 2 Map of problem

Based on previous analyze, the detailed specifications of problem has to be set, i.e. the following questions has to be answered:

- What is the shape of the labyrinth?
- Which sensors will be used?
- What is the construction of a robot?
- Which algorithm will control the robot?
- How the algorithm will be implemented?

These issues are necessarily interrelated. For example, if we consider the sensors, it is necessary to have an idea about labyrinth form, the algorithm of the problem solution can be designed until all the previous questions will be answered, etc.

A. Form of the labyrinth

First, the labyrinth form has to be defined. As seen in the Fig. 1, form of labyrinth is the main limiting factor to solution this specific problem. Individual problems as sensors, robot construction and algorithm depend on it. There is a mutual relationship, between the labyrinth and sensors, which explains the need to use appropriate materials for incident-free detection by available sensors.

Some labyrinths are, for simplicity, only represented by a line, others only by walls, or the labyrinth can be defined by both above possibilities. Such approach is used, for example, for orientation in space on the boxes, which are usually defined by black line. Our solution will also use both approaches.

The black line will facilitate the orientation of the robot model in space that will form the axis of every possible way even the blind. The robotic model will keep the black line during its movement and no accidental deviation occurs from the desired direction during the rotation of model on angle of 90 degrees.

The end of the labyrinth can be arranged differently colored mark – e.g. green square.

For proper detection of the color space is necessary to provide such a floor material that is flat and dull as possible because of possible reflection beam from the color sensor back in the sensor. Any reflected color component can cause an error in detecting the correct color – see Figure 3.

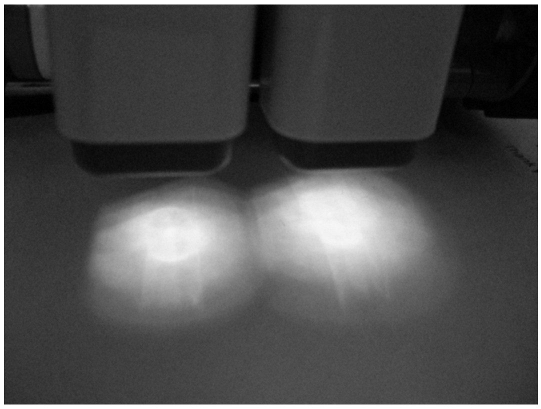

Fig. 3 Color sensors

The possible form of the labyrinth is shown on the Figure 4.

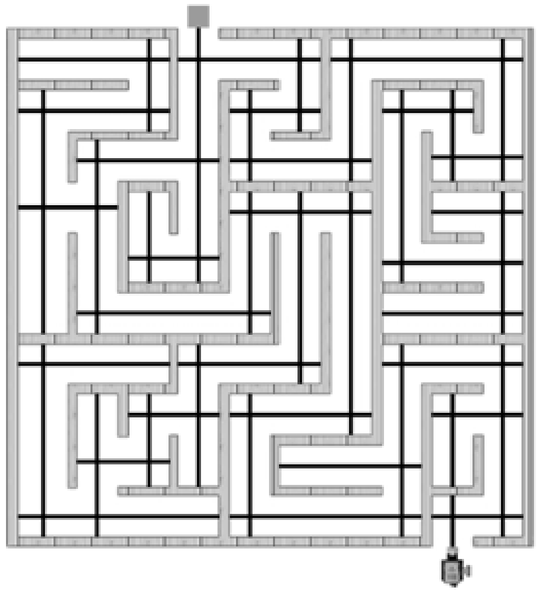

Fig. 4 Example of labyrinth

IV. Problem Solution

In this section the problem solution will be presented. First, the construction of the robot model will be shown, then the analysis of the algorithm of passing the robot will be discussed and finally the simulation program in NXT-G will be shown.

A. Construction of the robot model

Construction elements of LEGO kits offer almost unlimited possibilities of compilation of various constructional solutions. Through the gears, belts, beams, axles and other elements it is possible to construct sophisticated mechanical models which must be adequately fixed. Sufficiently reinforced model works more precise; on the other hand, it is good to think about the weight and dimensions of the robot itself.

1) Sensors

Generally the term sensor represents a specialized information source of the outside world for the controlling system. Biological sensors play an important role in human life in form of the senses, which human is able to perceive and which are further processed in the brain. In LEGO® construction set, then sensors perform similar function. These sensors transmit information into NXT intelligent brick using electronic “nervous” system.

As mentioned above, the robot will operate in a labyrinth form of both - walls and black line. To detect the barrier and black line the proper sensors has to be used.

Developing alternatives

Generally, various approaches exist for developing alternatives, for example, brainstorming or lateral thinking. In our case, the set of admissible variations is limited to a list of suitable sensors that we have available.

The following sensor can be used for detection of *black line*:

- color sensor;
- light sensor.

The following sensor can be used for detection of *barrier*:

- EOPD sensor;
- ultrasonic sensor.

In the following the limits and possibilities of above mentioned sensors will be discussed and based on decision making process the optimal solution will be found.

Color sensors versus light sensors

The light sensor is suitable for most applications to monitor the line due to the contrasting background. By detecting of the reflected light it is possible to distinguish shades of color objects. The measured intensity of the colors and the light itself is represented as a percentage, where 0% is considered the absolute darkness (color or black) and 100% indicates the maximum intensity of light (or white).

Color sensor handles feature of the light sensor, but it can also evaluate not only the primary colors of RGB (red, green, blue), but also black, yellow and white in the given range. Moreover color sensor works as a color lamp, emitting red, green, or blue light. We are not so dependent on shades of gray and objective, we can simply choose a specific color such as green, which is located approximately in the middle of the color spectrum.

Light and color sensors are shown on Figure 5.

Fig. 5 Light sensor and color sensor [11]

EOPD sensors versus ultrasonic sensors

EOPD sensor enables indication of the distance between the sensor and object or boundaries but with higher accuracy than the ultrasonic sensor. The HiTechnic EOPD is similar to a standard light sensor. The principle of operation is based on the pulse light for elimination of the surrounding lights and can be used for measurement of the distances up to 20cm. This sensor measures the distance based on the intensity of reflected light in raw (unscaled) values from 0 to 1023 or in scaled values from 0 to 100.

With ultrasonic sensor, the robot can react not only on distance of the boundaries, but also on their movement. Ultrasonic sensor uses similar principles as a submarine sonar or bat‘s orientation in space. Transmission part of the sensor sends a small dose of ultrasonic waves at a frequency of 40kHz, that is reflected from the obstacle and subsequently is captured by detection part of the sensor. Then, the distance of remote object is evaluated from the known speed of acoustic waves and calculating the time that takes for the reflected ultrasonic wave to go back. This sensor measures the distance from 0.0 m to 2.55 m and with an accuracy of ± 0.03m. Distance can be measured in either metric system and in centimeters, or inches.

EOPD and ultrasonic sensors are shown on Figure 6.

Fig. 6 EOPD sensor and ultrasonic sensor [11]

Decision making process - Sensors

Available decision making tools, such as weighted points and decision trees, involve setting out the facts in a logical way so that the overall decision is made as a result of many minor ones. Many specialized mathematical decision making techniques exist, but they are really tools of the operational research specialist. In our case, the decision-making process is based on easy choice of an optimal alternative of solutions of some minor problems.

Light sensor is sufficient for black line detection. But this sensor does not enable detect in the desired time the colored mark which indicates the end of the labyrinth. This is due to the transition between light and dark color on the interface. The sensor is not able to distinguish between the detected colors in shades of gray - the values from 0 to 100. If any shade of grey is used for the identification of the end of the labyrinth, it is very likely that the robot ends up still inside the labyrinth during the detection of two colors, i.e. white background and black guide lines. This weakness makes light sensor unusable for solution our specific maze. Color sensor can distinguish up to 8 colors in light mode; therefore this sensor is suitable for our purpose.

The weakness of EOPD sensor is its sensitivity to texture and color of the object. This causes the robot will work differently in principle identical labyrinth - robot will evaluate distance of walls of maze incorrectly. This sensor measures the distance based on the intensity of reflected light, not by length measurement system. Depending upon the above-mentioned weaknesses, with changing conditions we will find a relationship between the intensity of the reflected light and the actual distance only with difficulties. There was reported theoretical maximum distance of 20 cm under optimal conditions in sensor specification. It is not enough due to the size of the robot and maze in most cases. Although the ultrasonic sensor detects the thin wall or round objects with the difficulties, for the purpose of the labyrinth it is sufficient.

2) Robot construction

Compactness, functionality and especially taking into account the purpose for which the robot is built up has to be ensured in developing the robot model. The robot has to be able to detect boundaries wall in both directions – straight direction and one of lateral direction. This can be realized by one rotating sensor or for simplicity by two static and fixed sensors.

To follow the black line it is needed at least one color sensor. For smoother scanning of black line and mainly for additional detection of junction, it is advisable to use two color sensors.

Moving of the robot model is provide by two motors (one one left side and one on right side). The rubber band is used for better stability.

Robot model with the above mentioned requirements is shown on Figure 7.

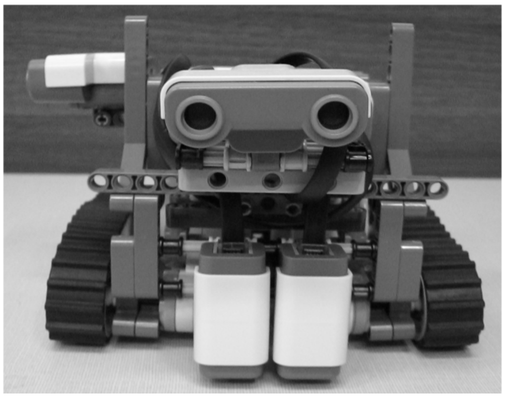

Fig. 7 Robot model construction

B. Algorithm development of robot passing through the labyrinth

The algorithm is generally clear procedure of solution of given problem. It has to meet certain rules. The result of algorithms can be already divided into several stages from problem formulation to debug of the program. The flowchart is one of the algorithmic language and a universal schematic description of the problem solution and is important support for the implementation of the process of problem solution to a particular programming language.

In the following we will discuss algorithm development of the process of passing through the labyrinth.

1) Analysis of the possible states

Well-functioning autonomous system carries not only minimal functionality, but is also able to deal with exceptional situations that may arise. In our case study the robot model has to be able to move in labyrinth - it has to be able to decide the right moment, when and how to move, or to recognize the "bad" maze.

Identification of intersection

The starting point for orientation in the labyrinth is a place where change of the direction of movement of the robot is expected. In practice, this place is realized by crossing of two black lines. For our robot model, this means that both color sensors detect black color.

Identification of wall

Depending on the construction design and dimensions of the robot the size of the labyrinth as well as the sensitivity of the sensors used for detection of the walls. In practice this means that the robot detecting the barrier should have sufficient space and time to react to this fact. Ultrasonic sensors used for this purpose is manually calibrated to 20cm, which is the approximate length of our assembled robot.

Identification of the end mark

The robot model recognizes the end of the labyrinth based on the green mark (by using color sensor). After detection of such green, the robot says "Good Job" and then the program is to be finished.

Identification of the absence of black line

The program will be terminated if robot identifies the wall barrier (by ultrasonic sensors) before black line (by color sensor).

2) Analysis of the robot motion

The eight possible situations can be found in labyrinth that robot has to solve to reach safely reach the end of the labyrinth. In the following these situations is to be discussed.

Straightaway – Figure 8:
detection:
- color sensors detect black color line, according to which varies the direction of travel;
- both ultrasonic sensors do not detect barrier;

action:
- the robot follows a line, so it goes straight.

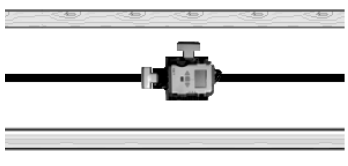

Fig. 8 Straightaway direction

Turn left – Figure 9:
detection:
- color sensors detect the crossing of the black lines;
- both ultrasonic sensors detect barriers;

action:
- the robot turn left 90 degrees;

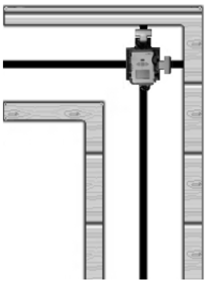

Fig. 9 Turn left

Turn right – Figure 10:
detection:
- color sensors detect the crossing of the black lines;
- front ultrasonic sensor detects barrier;
- right ultrasonic sensor does not detect barrier;

action:
- the robot turn right 90 degrees;

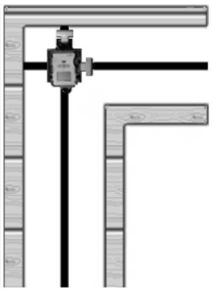

Fig. 10 Turn right

Intersection left – Figure 11:
detection:
- color sensors detect the crossing of the black lines;
- front ultrasonic sensor does not detect barrier;
- right ultrasonic sensor detects barrier;

action:
- the robot goes straight;

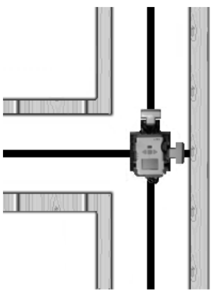

Fig. 11 Intersection left

Intersection right – Figure 12:
detection:
- color sensors detect the crossing of the black lines;
- both ultrasonic sensors do not detect barrier;

action:
- the robot turn right 90 degrees;

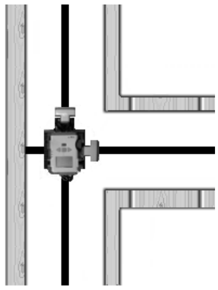

Fig. 12 Intersection right

Intersection left / right – Figure 13:
detection:
- color sensors detect the crossing of the black lines;
- front ultrasonic sensor detects barrier;
- right ultrasonic sensor does not detect barrier;

action:
- the robot turn right 90 degrees;

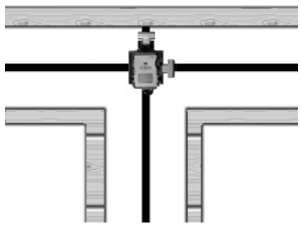

Fig. 13 Intersection left / right

Intersection complete – Figure 14:
detection:
- color sensors detect the crossing of the black lines;
- both ultrasonic sensors do not detect barrier;

action:
- the robot turn right 90 degrees;

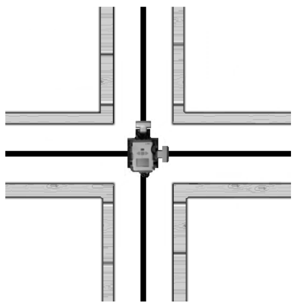

Fig. 14 Intersection complete

Blind lane – Figure 15:
detection:
- color sensors detect black color line, according to which varies the direction of travel;
- both ultrasonic sensors detect barrier;

action:
- the robot turn back 180 degrees;

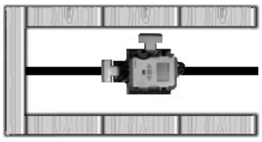

Fig. 15 Blind lane

3) Algorithm of solution of the

The algorithm of the problem in the form of a flowchart (shown on the Figure 16) repeats in an endless loop. The flowchart of problem, which solves the maze, consists of three basic flowchart. In current practice, symbol terminator is used to describe the begin and end of the algorithm, symbol process for the command(s) and symbol decision for branching algorithm depending on the possible conditions. As the robotic model uses multiple sensors whose information is used for decision making, this is displayed in so-called nested branching.

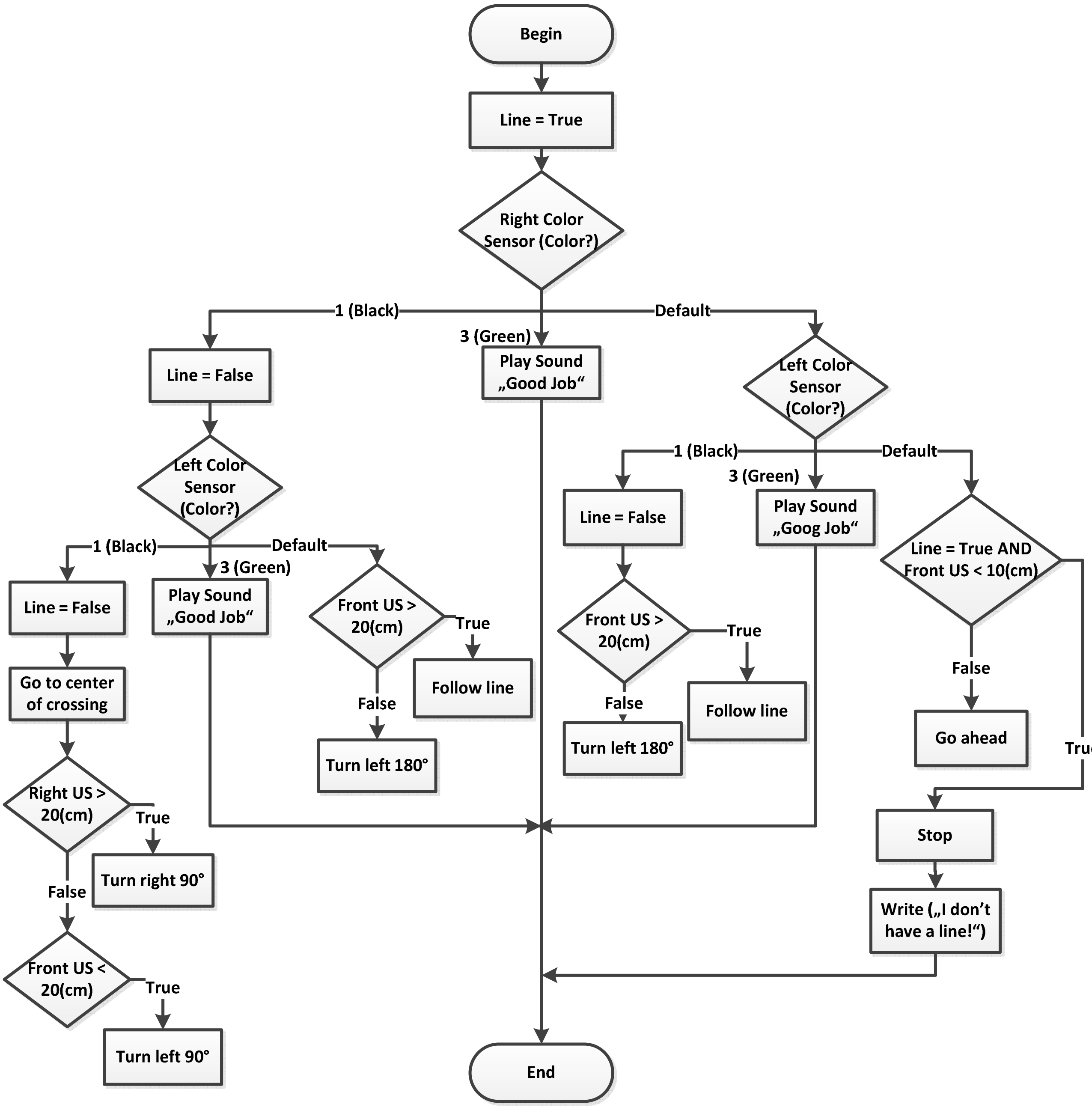

Fig. 16 The algorithm of the problem in the form of a flowchart

Verbal expression of the algorithm

Immediately after starting the program, the logical variable „Line“ stores the value „True“. This variable, with the help of color sensors, has the task of detect respectively store information on whether the robot captured line. Value „True“ means that the robot noticed line, value „False“ conversely. It sounds illogical, but in this case the inverted logic easier to ensure proper functionality. This mechanism solves the problem of collision the robot, if the robot finds itself in the "bad" maze without line at the beginning. If the robot does not recognize a line, it will go straight until it will not recognized the presence of obstacles in the distance of less than 10cm, then it will stop, it will display a message on the screen and the program will terminate.

If one of the sensors captures the green color, indicating target, NXT brick will play sound "Good Job" and then the program will exit.

If both color sensors detect the line, it means that it was found the intersection, so the robot will move to the center of the intersection and then it will be examined the path by ultrasonic sensors. Primarily it is used right sensor for orientation so if it is free on the right side, robot will turn 90 degrees to the right and will continue to move. If both sensors detect a color other than black or green and there is space in front of robot, it will go directly, if not, it will turn to the left about 90 degrees and then it will continue driving.

If the robot enters into a dead end, ie the front sensor records the barrier without being detected intersection, the robot will turn to the left by 180 degrees. If the front sensor does not detect an obstacle, the robot will follow a line as it will move forward.

The algorithm has been implemented in graphical programming language NXT-G. Part of the NXT-G program is shown on the Figure 17.

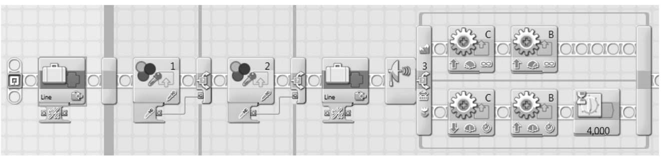

Fig. 17 Part of the NXT-G program

V. Conclusion

There are various approaches how to provide training in algorithms, how to introduce and develop basic algorithmic thinking of students.

The paper offered one of the kinds of the possible teaching / learning strategies using the system approach including problem analysis. The system approach can be set as the default paradigm for a wide integration of the principles of the algorithm development into education. The authors highlight importance of utilizing systematic problem-solving approaches, like problem analysis, for tackling problems in algorithmic tasks in teaching algorithm development and programming. The paper emphasizes the fact that the algorithm development of the real processes should be supported by computer simulation.

Solution of the process of passing through the labyrinth has been chosen as one possible example of how to present the application of the system approach.

Acknowledgment

This research has been partially supported by the Specific research project of the Faculty of Education of University of Hradec Kralove No. 2134.

References

[1] O. Horák, L. Mitrovič, “Description of the Basic Algorithm Blocks and Structures Representation in Courses of Algorithm Development”, *WSEAS Trans. on Information Science & Applications on Advances in Engineering education.* vol. 9, No. 2. 2012.

[2] S. Hubalovsky, J. Jelinek, J. Sedivy, „Mathematical modeling and computer simulation of optimal reaction time of the Lupine protein hydrolysis using fermented whey”, *International Journal of Mathematical Models and Methods in Applied Sciences*. vol. 6, No. 2., 2012.

[3] J. Sedivy, S. Hubalovsky, “Mathematical foundations and principles in practice of computer aided design simulation”, *International Journal of Mathematics and Computers in Simulation.* vol. 6, No. 1. 2012.

[4] S. Hubalovsky,” Modeling and computer simulation of real process – solution of Mastermind board game”, *International Journal of Mathematics and Computers in Simulation.* vol. 6, No. 1. 2012.

[5] J. Šedivý, **“**Multimedia support of parametric modeling”, in *Proc. 9th WSEAS International Conference on Engineering Education (EDUCATION '12).* WSEAS Press, 2012.

[6] K. Dvořák, J. Šedivý, “CAx application in the teaching of engineering subjects”, in *Proc. 9th WSEAS International Conference on Engineering Education (EDUCATION '12).* WSEAS Press, 2012.

[7] J. Bailer, M. Daniela, “Tracing the Development of Models in the Philosophy of Science”, *Magnani, Nersessian and Thagard*, 1999.

[8] S. Hartmann, “The World as a Process: Simulations in the Natural and Social Sciences”, In R. Hegselmann, et al., *Modelling and Simulation in the Social Sciences from the Philosophy of Science Point of View*, Theory and Decision Library. Dordrecht: Kluwer, 1996.

[9] IRM Training (ACN 007 219 589). Available: http://www.irm.com.au/papers/Problem_Analysis_Techniques.PDF (Retrieved 12/09/02).

[10] L. Mitrovič, Š. Hubálovský, “Simulation of the pass through the labyrinth as a method of the algorithm development thinking“, in *Proc. Visualization, imaging and simulation (VIS '12).* WSEAS Press, 2012.

[11] LEGO Education. Available: http://www.legoeducation.us/eng/product**.**

Stepan Hubalovsky was born in 1970 in Czech Republic. He obtained master degree in education of mathematics, physics and computer science in 1995 and doctor degree in theory of education in physics in 1998 both in Faculty of Mathematics and Physics, Charles University in Prague, Czech Republic. He works as associated professor on University of Hradec Kralove His scientific activities are system approach, modeling and simulation.