

Modeling and computer simulation of real process – solution of Mastermind board game

Stepan Hubalovsky

Abstract—One of the most important methods in current scientific and technological research as well as in research of strategy algorithm and programming is modeling and computer simulation of real systems and real processes. System approach, modeling and simulation are discipline with its own theory and research methodology.

The paper focuses to the theory of modeling, simulation, visualization, model validation and verification of feedback controlled real process – solution of the Mastermind board game. Multidisciplinary approach is point out too.

Step by step there will be shown the process of system identification, mathematical analysis and strategy of the solution. Conceptual model of the process is realized by process chart. Conceptual model is supplemented by the simulation model realized in Visual Basic for Excel. Visualization is a part of the simulation model. Validation of the conceptual model as well as verification of the simulation model is shown in the paper too.

Keywords—Algorithm development, Mastermind, algorithmic thinking, education, system approach, multidisciplinary learning, computer simulation and modeling.

I. INTRODUCTION

CCOURSES in 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 [1], [2], [3].

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 first briefly introduce the theory of system

approach, modeling and simulation as a method of multidisciplinary and system approach to education of algorithm development.

Secondly we introduce a case study illustrating step by step process mathematics analysis, strategy and algorithm development and computer simulation of a real process – *solution of Mastermind board game*.

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 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 equation;
- 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. 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 and spreadsheets (MS Excel).

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 is simulated in Visual Basic for Application and visualized in MS Excel spreadsheet.

C. Simulation of feedback regulated systems and process

Many real systems as well as real processes are constructed so that they are able control themselves by means of the feedback.

Feedback is a mechanism that is looped back to control a system / process within itself. Such a loop is called a feedback loop. In the system or process containing the feedback loop the input characteristics or signals are influenced by output characteristics or signals.

The computer simulation is essential method not only for representation of the conceptual model but can also be used for investigation of the feedback affected processes. In this case the simulation allows:

- realization of the numbers of different steps of process that cannot be investigated by other way;
- setting the parameter of the feedback process and its optimization;

- visualization of the behavior of the feedback regulated process.

Feedback affected behavior of the real process will be shown in the paper.

D. 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.

E. 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 – *solution of Mastermind board game* – 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 Fig. 1.

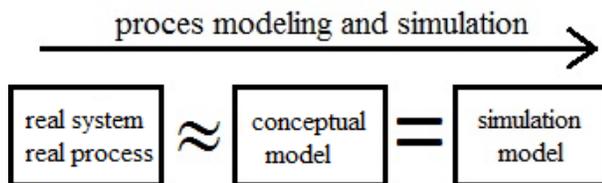


Fig. 1 Process modeling and simulation

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).

F. 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. CASE STUDY -

SOLUTION OF THE MASTERMIND GAME AS REAL PROCESS

A. Principles and rules of the Mastermind game

Mastermind is a code-breaking game for two players.

The game is played using:

- *decoding board*, - shield at one end covering a row of four (or five) holes (matrix), and twelve (or ten, or eight) additional rows containing four (or five) holes next to a set of four (or five) small holes;
- *code pegs* of six (or eight) different colors;
- *key pegs*, black and white.

The code-maker chooses a pattern of four (or five) code pegs from six (or eight) possible ones. Duplicates are allowed. The chosen pattern is placed in the four (or five) holes covered by the shield, visible to the code-maker but not to the code-breaker.

The code-breaker tries to guess the pattern, in both order and color. The code-maker provides feedback by placing from zero to four (or five) key pegs in the small holes of the row with the guess. A black key peg is placed for each code peg from the guess which is correct in both color and position. A white peg indicates the existence of a correct color peg placed in the wrong position (taken from [13]).

B. Identification of the process of solution as real system

Although the Mastermind game is a real system, we identify this system as a process of finding the strategy and algorithm of the solution. Figure 2 shows identification of the process of the solution as system with the elements, element interactions, input and output.

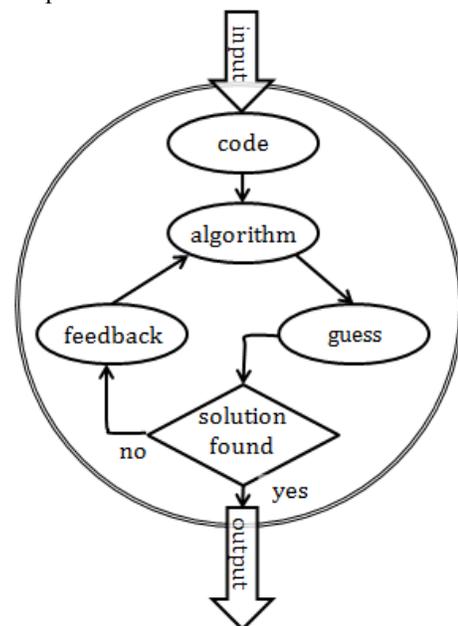


Fig. 2 Identification of the process of solution as system

IV. PROBLEM DEFINITION

A. Definition of the initial condition

Before the strategy of solving the game Mastermind will be presented it is important to set the initial condition of our particular solution.

1st condition: in this paper we will focus for the Mastermind variation with 8 colors (pegs) and 5 holes;

2nd condition: we will assume more simple solution in which the duplications of code pegs are not allowed, so it means that code pegs include only one color;

3rd condition: solution has to be found in the final steps, but not in the fastest way. Instead we will put the emphasis on easy and clear algorithm.

The second and the third condition are simplifications which allow using this case study in learning of the algorithm development. Finding the algorithm where duplications are allowed as well as the algorithm for the fast strategy is not suitable for the educational purposes. The algorithm of the fast strategy where duplications are allowed can be found e.g. in [12].

B. Analysis of the problem

The analysis of the problem solution and algorithm development under the conditions listed in above paragraph can split to the following parts:

- 1) Creation of the mathematical model – mathematical analysis;
- 2) Strategy analysis;
- 3) Algorithm and computer simulation program development.

V. ANALYSIS OF THE PROBLEM

A. Mathematical representation of the system Mastermind

To Instead of real Mastermind game (plastic board with colored pegs) the mathematical model of the game has to be the colored pegs and the five dimensional ordered vectors will be used. The code vector is $\mathbf{c} = (c_1, c_2, c_3, c_4, c_5)$, where coordinates of the vector has values $c_i = \{1, 2, 3, 4, 5, 6, 7, 8\}$. The guess vector is $\mathbf{g} = (g_1, g_2, g_3, g_4, g_5)$, where coordinates has again values $g_i = \{1, 2, 3, 4, 5, 6, 7, 8\}$. One particular example of the vector \mathbf{c} is i.e. $\mathbf{c} = (2, 5, 8, 1, 3)$.

The response pegs can be represented by two dimensional ordered vector $\mathbf{r} = (b, w)$, where coordinate $b = \{0, 1, 2, 3, 4, 5\}$ represents the number of correct colors in correct positions and coordinate $w = \{0, 1, 2, 3, 4, 5\}$ represents the number of correct colors that are part of the code but not in the correct positions. Particularly, the vector $\mathbf{r} = (2, 3)$ means that the guess vector has all five correct colors, but only two of them are in the correct positions.

B. Number of combinations of code vector

The number of the variations of the code pattern under the 1st and the 2nd condition (we select 5 different elements from 8 different ones) is given by variation without repetition.

The number of k -variations without repetition from a given set of n elements is often denoted in elementary combinatorics texts by $V(n, k)$ and the formula for the calculation of the number is:

$$V(k, n) = \frac{n!}{(n-k)!} \quad (1)$$

In our particular study case $n = 8$ and $k = 5$. The number of the possible variations equals:

$$V(5, 8) = \frac{8!}{(8-5)!} = \frac{40320}{6} = 6720 \quad (2)$$

The probability that the first guess will be correct (that the player choose correct color pegs in correct order) is:

$$P_{V1} = \frac{1}{6720} = 1.49 \times 10^{-4} \quad (3)$$

This probability is rather low.

Let's calculate the number of the possible combinations of the code pattern of 5 different patterns from 8 ones, but without the ordering. The number is given by k -combinations without repetitions from a given set of n elements and in the combinatorics it is denoted as $C(n, k)$. The formula for the calculation is as follows:

$$C(k, n) = \frac{n!}{k!(n-k)!} \quad (4)$$

In our particular case, where $n = 8$ and $k = 5$, the number of the combinations is:

$$C(5, 8) = \frac{8!}{5!(8-3)!} = \frac{40320}{720} = 56 \quad (5)$$

The probability that the first guess of the chosen color pegs will be correct is:

$$P_{C1} = \frac{1}{56} = 1.78 \times 10^{-2} \quad (6)$$

This probability is more than hundred times higher than probability P_{V1} where the ordered pegs have been considered. This result will be taken in the account in the following part, where the strategy of the algorithm development will be discussed.

Let's assume, that we have already found all five unordered colors but regardless, i.e. the guess vector $\mathbf{g} = (g_1, g_2, g_3, g_4, g_5)$ has the same values of the coordinates as vector $\mathbf{c} = (c_1, c_2, c_3, c_4, c_5)$, but the order can be different. Number of possible ordering of the coordinates is given by permutation $P(n)$:

$$P(k) = k! \quad (7)$$

In our particular study case $k = 5$. The number of the possible permutation is $P(5) = 5! = 120$.

C. Number of combinations of key vector

Let's now discuss numbers of responses given by black pegs (number of correct colors in correct positions) and white pegs (number of correct colors but not in the correct positions). The total number is given by k -combination with repetitions from a set of n elements $C'(n, k)$. The common formula for the calculation is given by following formula:

$$C'(k, n) = \binom{n+k-1}{k} = \frac{n(n-1)\cdots(n+k-1)}{k!} \quad (8)$$

In our particular case we choose from set of 3 elements (black peg, white peg and no peg), so the $n = 3$. The number $k = 5$, because we the response pattern has 5 holes. Total number of combination is:

$$C'(5,8) = \binom{7}{5} = 21 \quad (9)$$

Because the fact the response combinations depends on the relation between the guess and code pattern some of the combination has to be removed from the total number 21. The total possible number is 16, which is clear from the Table 1, where impossible combinations are shown.

	Colors					No	value of $r = (b, w)$
1						No	(0,0)
2	W					No	(0,1)
3	B					No	(1,0)
4	B	B	B	B	W	No	(4,1)
5	B	W	W	W	W	No	(1,4)

Table 1 Impossible combinations of key pegs

The combinations 1, 2 and 3 are impossible, because, if at least 2 guess pegs have to match with the code pattern (the maximum incorrect guess pegs is 3 and rest 2 has to be correct). The combination 4 and 5 are impossible, because if the 4 pegs are in position the rest one has to be also in the correct position.

Detailed mathematical analysis of the Mastermind solution can be found e.g. in [10], [11].

VI. STRATEGY ANALYSIS – CONCEPTUAL MODEL

In this paragraph we will discussed the most appropriate strategy for algorithm development and computer simulation programming of this problem. Developed algorithm should fulfill the criteria and initial condition we specify in the above part of the paper.

The most optional solution is that both colors and code pattern is guessed at the same time. The algorithm of this strategy is rather difficult and can be found e.g. in [12].

In this paper the different strategy will be presented. The solution will be divided to two steps – two independent algorithms.

Based on the first algorithm the code colors are to be

decoded regardless to its ordering. The second algorithm decodes the color order.

A. Principle of the decoding of the colors

Process of the color decoding will be shown and explained based on the two examples shown in the Table 2 and Table 3. Instead of the colors pegs of the code pattern and guess patterns the representation via vectors c and g_i is used. Instead of response feedback pegs the vector r_i is used. Index i indicates the step of the decoding process.

Code c	2	4	5	1	8		
Step i	Guess $g_i = (g_1, g_2, g_3, g_4, g_5)$					Response $r_i = (b, w)$	$R_i = b + w$
1	1	2	3	4	5	0	4
2	6	2	3	4	5	0	3
3	1	7	3	4	5	0	3
4	1	2	8	4	5	0	5

Table 2 Process of color decoding – Example 1

The first example shown in the Table 2 represents start and basic principle of the decoding process. Regarding the fact, that in the first step there is nothing known about the code c , the coordinates of the guess vector g can be set arbitrarily, e.g. $g_1 = (1, 2, 3, 4, 5)$.

The last column of the table represents response summation $R = b + w$ of the coordinates of the feedback vector, which gives the most important information for process of the color decoding – the number of correct coordinates (regardless its ordering). The response summation can reach values $R = \{2, 3, 4, 5\}$ (see Table 1).

The action in the second step depends on the value of R_1 . There are only two possible results.

1. If $R_1 = 5$, then the process finished – guess vector $g_1 = (1, 2, 3, 4, 5)$ equals to code vector c .
2. If $R_1 \neq 5$, then the process has to continue.

Let's go back to our example shown in the Table 2. The vector g_1 matches the vector c in for coordinates and doesn't match in one coordinate. It means that value of one of the coordinates of g_1 (i.e. 1, 2, 3, 4 or 5) has to be replaced by one value "6", "7" or "8". It is clear from the Table 1, that value "3" has to be replaced by value "8", but this information is during the real playing of the game unknown. This means, that the first coordinate of g_1 (i.e. value 1) is to be replaced by i.e. value "6" – the vector $g_2 = (6, 2, 3, 4, 5)$ is created. Comparing vectors g_2 and c the value of R_2 can be calculated. In our particular case from the Table 2 we reach value $R_2 = 3$, $R_2 < R_1$.

This result means, that the number of correct coordinates decreases and vector c includes coordinate with value "1" and doesn't include coordinate with value "6". It follows that in the next steps the vector g_i has to include value "1" and doesn't include value "6".

The action in the third step depends on the value of R_1 and

R_2 . In the case of the example of the Table 2 the vector $g_3 = (1, 7, 3, 4, 5)$, where the first coordinate was replaced back from value 6 to value 1, and the second coordinate was replaced from value “2” to value “7”. By calculation response summation $R_3 = 3$ ($R_3 < R_1$) the similar result as in step 2 is obtained – vector c includes value “2” and doesn’t include value “7”.

The action in the next step is the same as in the previous step. Vector $g_4 = (1, 2, 8, 4, 5)$ is created by replacing back the value of the second coordinate from “7” to “2” and by replacing the value of the third coordinate from value “3” to rest value “8”. Response summation is $R_4 = 5$ in this step. This result means that:

1. Vector c doesn’t include value “3” and includes value “8”.
2. Regarding the fact, the $R_4 = 5$, the process of the color decoding ends – vectors $g_4 = (1, 2, 8, 4, 5)$ and $c = (2, 4, 5, 1, 8)$ has the same coordinate values, but regardless to their ordering.

Code c	7	2	3	8	5			
Step i	Guess $g_i = (g_1, g_2, g_3, g_4, g_5)$					Response $r_i = (b, w)$		$R_i = b + w$
1	1	2	3	4	5	1	2	3
2	6	2	3	4	5	1	2	3
3	1	6	3	4	5	1	1	2
4	7	2	3	4	5	2	2	4
5	7	8	3	4	5	2	2	4
6	7	2	8	4	5	3	1	4
7	7	2	3	8	5	2	3	5

Table 3 Process of color decoding – Example 2

The first two steps are the same as in the previous Example 1. The values of response summation $R_1 = 3$ and $R_2 = 3$ are the same. Equality of R_1 and R_2 means two possible results:

1. Both values of the coordinates (“1” and “6”) are included in the code vector c ;
2. Both values of the coordinates (“1” and “6”) are not included in the code vector c .

The check, which of the above possibility is valid has to be done in the next step. The realization is clear from the Table 3. In the third step the value of first coordinate of g_3 is let “6” and the value of the second coordinate is replaced from “2” to “1”. By calculating the response summation the three independent results can be obtained:

1. If the value of the R in this step is less than the values of R in the previous steps, then both values of the checked coordinates doesn’t include to vector c (it is the case shown in the step 3 of our example);
2. If the value of the R in this step is higher than the values of R in the previous steps, then both values of the checked coordinates includes in vector c ;
3. If the value of the R in this step equals to the values of R in the previous steps, then other checks by means of the

replacement of next coordinate of the vector g has to be realized. The situation is shown in the steps $i = 5, 6, 7$, where step by step the coordinate values “2”, “3” and “4” are replaced by value “8”.

Response summation $R_7 = 5$ in the step 7. This result means that:

1. Vector c doesn’t include value “4” and includes value “8”.
2. Regarding the fact, the $R_7 = 5$, the process of the color decoding ends.

Summary and generalization of the process color decoding

Summary and generalization of the above mentioned strategy of finding color code regardless to the color ordering can be divided to the following more less independent parts – procedures.

Procedure 1 – The First step:

This procedure creates the vector $g_1 = (1, 2, 3, 4, 5)$ and provides the check if response summation $R_1 = 5$. If not, the process continues by Procedure 2.

Procedure 2 – The Second step:

This procedure replaces value of the first coordinate of vector g_1 by value 6 and calculates the response summation R_2 .

Based on comparison values of R_1 and R_2 the process continues in one of the three following procedures, where $i = 2$.

Procedure 3 – Decrease of value of R_i :

This procedure is executed if value of the last response summation R_i is less than $\text{Max}(R_j)$, where $j = 1, 2, \dots, i-1$. The procedure provides two actions:

- The value of the g vector coordinates, replaced in the previous step, is returned back;
- Next coordinate of the vector g is replaced by next value, which was not yet included to the replacement.

Procedure 4 – Increase of value of R_i :

This procedure is executed if value of the last response summation R_i is higher than $\text{Max}(R_j)$, where $j = 1, 2, \dots, i-1$. The procedure provides one action:

- Next coordinate of the vector g is replaced by next value, which was not yet included to the replacement.

Procedure 5 – Equality of value of R_i :

This procedure is executed if value of the last response summation R_i equals to $\text{Max}(R_j)$, where $j = 1, 2, \dots, i-1$. The procedure provides one actions:

- The value of the g vector coordinates, replaced in the previous step, is returned back;
- Next coordinate of the vector g is replaced by the same value used for replacement in the previous step;
- If value of response summation in this step R_{i+1} equals to R_i ($R_{i+1} = R_i$) the Procedure 5 is executed again;
- If value of response summation in this step R_{i+1} is higher than value of R_i ($R_{i+1} > R_i$) then the replacement of given coordinate and new valid coordinate were found;
- If value of response summation in this step R_{i+1} is less

than value of R_i ($R_{i+1} < R_i$) then the actual replacement is returned back and given coordinate is replace by next value, which was not yet included to the replacement.

The above procedures run in the following order:

- Procedure 1, Procedure 2 and repetitively based on the described conditions Procedure 3, Procedure 4 and Procedure 5. The process ends if value of the response summation equals first time to 5. Flowchart of the algorithm of color decoding is shown on the Figure 3.

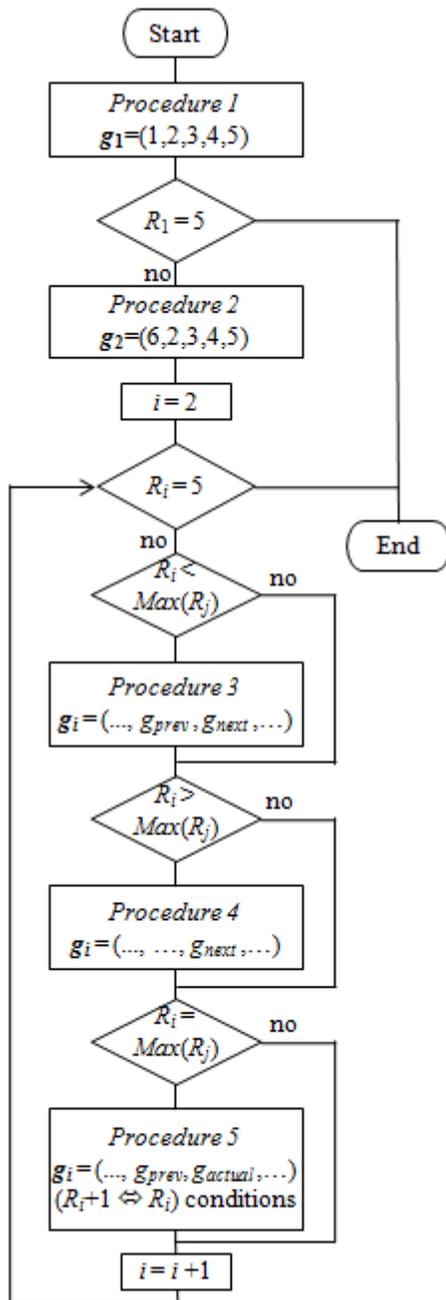


Fig. 3 Flowchart of the algorithm of color decoding

B. Principle of the decoding of the colors order

Process of the order decoding will be shown and explained based on the example shown in the Tables 4. Instead of the colors pegs of the code pattern and guess patterns the representation via vectors c and g_i is used. Instead of response feedback pegs the coordinate b_i vector of the r_i is used. Index i indicates the step of the decoding process.

Code c		2	4	3	5	1	
Step i		Guess $g_i = (g_1, g_2, g_3, g_4, g_5)$					b_i
1	Matrix G_1	1	2	3	4	5	1
2		5	1	2	3	4	0
3		4	5	1	2	3	0
4		3	4	5	1	2	1
5		2	3	4	5	1	3
6	Matrix G_2	5	4	3	2	1	3
7		1	5	4	3	2	0
8		2	1	5	4	3	1
9		3	2	1	5	4	1
10		4	3	2	1	5	0
11	Matrix G_3	2	1	3	5	4	3
12		4	2	1	3	5	0
13		5	4	2	1	3	1
14		3	5	4	2	1	1
15		1	3	5	4	2	0

Table 4 Process of position decoding

The first row of the Table 4 represents code vector $c = (2, 4, 3, 5, 1)$ - positions of the coordinate values. Based on the equation (7) the total number of possible permutation of values of the vector c is $5! = 120$.

The last column represents the values b_i for each permutation of the values of the guess vector g_i . The values of b_i can reach the values $b_i = (0, 1, 2, 3, 5)$.

In our particular case shown in the Table 4 the process of the order decoding is as follows.

First, the cyclic permutation of the vector $g_i = (1, 2, 3, 4, 5)$ are created and corresponding values b_i are calculated. The result is shown in the first five rows ($i = 1, \dots, 5$) of the guess part of the table like matrix G_1 . It is evident, that the summation of values b_i for these five cyclic permutations is $1 + 0 + 0 + 1 + 3 = 5$.

No final result can be evaluate from this first five permutation, except the case if one of the vector g_i equals to vector c , so in the case when one of $b_i = 5$.

The process continues by the same way – see rows 6 – 10 of the Table 4, where are shown cyclic permutations of the vector $g_i = (5, 4, 3, 2, 1)$ - matrix G_2 and corresponding values b_i (summation of b_i is again $3 + 0 + 1 + 1 + 0 = 5$).

Sometimes the final result can be evaluated by the following process of evaluation of these first two matrixes. Let's take the first permutation of matrix G_1 , where at least one position is correct – this permutation is in the row 1. We will take first

element – “1” and assume (for a moment), that its position is correct. But comparing it with the same element “1” and same column of matrix G_2 (permutation in the row 7) we get result, that element “1” cannot be in the first position. Total analysis of all permutation of matrixes G_1 and G_2 is shown in the Table 5.

Possible positions of the elements		2	3	4	
	3	4	5		
	2			5	1
Semi - final result 1	3	2	5	4	1
Semi - final result 2	2	4	3	5	1

Table 5 Process of matrixes evaluation

It is evident from the Table 5 that we reach two results – the second one corresponds to the code vector positions.

In the case of ambiguous result the next matrix G_3 has to be generated by the same way like previous two ones (G_1 and G_2). The matrix G_3 is shown on the Table 5 in the rows 11 – 15. This matrix is generated by cyclic permutation of vector $g_i = (2, 1, 3, 5, 4)$.

Process evaluation of the matrixes G_1 , G_2 and G_3 is evident from the Table 6.

Possible positions of the elements			3		
	3	4			
	2			5	1
Final result	2	4	3	5	1

Table 6 Process of matrixes evaluation

Summary and generalization of the process position decoding

Above mentioned description of the process of decoding the color order can be divided into two procedures:

Procedure 1 – Creation of the matrix:

This procedure creates the matrix G (5×5) of the five cyclic permutations of the vector g and calculates corresponding values b .

Procedure 2 – Evaluation of the matrixes:

This procedure evaluates the matrixes G and finds all possible results based on above description.

Procedures 1 and 2 are executed repetitively until one final result is reached. The maximum of the repetition found by testing of this process on the computer simulation program is three. The flowchart of the algorithm of order decoding is shown on the Figure 4.

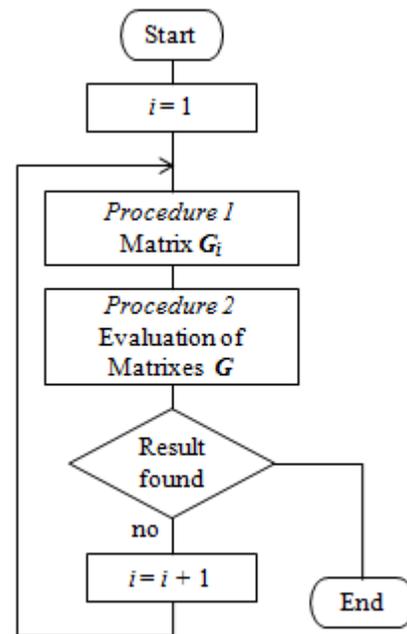


Fig. 4 Flowchart of the algorithm of order decoding

VII. COMPUTER SIMULATION OF THE ALGORITHM IN MS EXCEL

The algorithms presented in the flowcharts on the Figures 3 and 4 represent the conceptual model. Based on this model the computer simulation program can be created in standard programming language – Visual Basic for Application.

A. Programming in Visual Basic for MS Excel

All MS Office application like MS Excel, MS Access, MS Word, MS PowerPoint, MS Outlook and others, have implemented a programming language *Visual Basic for Application (VBA)*. It is an object-oriented programming language, which is based on Visual Basic.

VBA contains the same commands as Visual Basic and extra commands that operate with objects of their own parent application (with cells, selection, sheets, etc. in MS Excel). This can be regarded as the main advantage, especially in teaching of programming. Every user of MS Office has a programming language, and s/he can try to use it as s/he likes, s/he can create his/her own “little useful application”, which can become the basis on which a further interest in programming develops.

VBA, although it is an object-oriented language, allows handling the basic algorithmic tasks based on fully structured, sequential approach, which is especially helpful in the initial stages of algorithm development and programming.

B. VBA Integrated development environment

The integrated development environment (IDE) of VBA is available in MS Excel 2007 in the ribbon tab “Developer”. This tab is not displayed by default. The tab “Developer” in version of MS Excel 2007 can be activated in the “Excel Options” dialog box (which is reached by clicking on the

Excel Option button in the main menu) in the folder "Popular" – see Figure 1. The "Show Developer tab in the Ribbon" checkbox has to be checked.

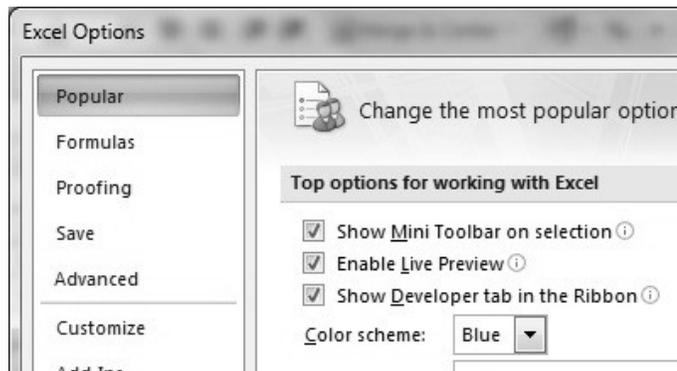


Fig. 5 Activation of Developer tab

To be able to exploit this IDE, the macro security has to be set to low. In MS Excel 2007, this setting can be performed on the "Developer tab", "Macro Security" command, where the option button "Enable all macros" has to be checked.

The IDE can be opened in two ways - either through the "Developer tab", click on "Visual Basic" command, or by pressing the keys Alt-F11.

The IDE has the Project Explorer in its left part, which shows a list of currently open projects - Workbooks, Worksheets, Modules and Forms.

In the case of sequential structured programming, the programs are saved in and run from separate *Modules*. One module can contain more programs.

In the case of object-oriented programming, the *Graphical user interface elements (GUI)* are inserted in the *Form*. VBA includes a reduced number of GUI like *Command_Button*, *Text_Box*, *Label*, *Option_Button* and *Check_Box*.

C. Inputs / Outputs in VBA for MS Excel

The VBA for MS Excel gives different possibilities of Inputs and Outputs of variables.

Inputs:

- 1) The command `A = InputBox(„Input A“)` opens the input dialog box into which the user inputs the value of the variable A.
- 2) Command `A = Cells (R, C)` is the value of variable A read from the cell in row R and column C.
- 3) The `TextBox` can be used for reading the data in the object oriented approach of programming.

Outputs:

- 1) The command `MsgBox(A)` opens the output dialog box in which the value of variable A is output.
- 2) Command `Cells (R, C) = A` saves the value of variable A in a cell in row R and column C. This value can be used in following program executions.
- 3) The `TextBox` can be used for writing the data in the object oriented approach.

D. Variables of array type

Unlike the conventional programming languages like C #, Delphi, Visual Basic, etc., the VBA for MS Excel enables two ways of working with data stored in the form of vectors (one-dimensional variables) or matrixes (two-dimensional variables).

The first option is the same as for other languages, where the vector and matrix variables are declared in a section of variable declarations:

```
Dim A(1 To 26) As String      'Vector
```

```
Dim B(1 To 26, 1 To 26) As Long 'Matrix
```

The second option is to store one- and two-dimensional data directly into cells in Excel worksheets. The command

```
Cells(2,3) = 10
```

can be understood as an element of the matrix with the row coordinates 2 and column coordinates 3 with the value 10. The advantage of this approach with the vector and matrix data is that the values of individual elements of the vector / matrix are visible to the user.

E. Computer simulation model of color decoding

Computer simulation model created in VBA is visualized in *Color decoding* MS Excel sheet.

Vector data (variables of array type) are saved into the cells:

- The input for the simulation model - code vector c is entered and saved directly to cells in the range B3:F3;
- The output - guess vectors g_i are saved in the columns B:F below from the row 5;
- Corresponding feedback – response R_i is calculated and saved in the cells of the column J.

The simulation program is divided into to three sub-procedures:

- **Guess_Code**: this procedure creates new vector g_i based on the conditions given by value of response R_i ;
- **Feedback**: calculates the value of response R_i ;
- **Automation_Run**: repetitively calls the procedures **Guess_Code** and **Feedback**.

The sub-procedure **Guess_Code** has the following code:

```
Sub Guess_code()
R = AR 'Identification of actual step
If R = 2 Then 'step 2
    For C = 1 To 5
        Cells(R, C) = C
    Next C
End If
If R > 2 Then 'Other steps
    No = Cells(R - 1, 7) 'Actual change coordinate value
    Poz = Cells(R - 1, 8) 'Actual position of change
    Sh = Cells(R - 1, 9) 'Identification Ri = Max(Rj)
    ****Guess if Ri = Max Rj
```

```

If (Cells(R - 2, 6) > Cells(R - 1, 6)) And (R > 3) Then
  Dwn = Cells(R - 2, 9) 'Identification Ri < Max(Rj)
  For C = 1 To Poz - 1
    Cells(R, C) = Cells(R - 2 - Dwn, C)
  Next C
  Cells(R, Poz) = No
  For C = Poz + 1 To 5
    Cells(R, C) = Cells(R - 1 - Dwn, C)
  Next C
  ***
  ***Guess if Ri < Max(Rj)
Else
  If Sh <> 0 Then
    Row = Cells(R - 1, 10)
    For C = 1 To Poz - 2
      Cells(R, C) = Cells(R - 1, C)
    Next
    Cells(R, Poz - 1) = Cells(Row, Poz - 1)
    Cells(R, Poz) = No
    For C = Poz + 1 To 5
      Cells(R, C) = Cells(R - 1, C)
    Next C
  End If
  ***
  ***Guess if Ri > Max(Rj)
  If Sh = 0 Then
    For C = 1 To Poz - 1
      Cells(R, C) = Cells(R - 1, C)
    Next C
    Cells(R, Poz) = No
    For C = Poz + 1 To 5
      Cells(R, C) = Cells(R - 1, C)
    Next C
  End If
  ***
End If
End If
End Sub

```

The sub-procedure Feedback has the following code:

```

Sub Feedback()
R = AR 'Identification of actual step
Cells(R, 6) = 0
  ***Calculation of Ri
For C = 1 To 5
  For CC = 1 To 5
    If Cells(R, C) = Cells(1, CC) Then
      Cells(R, 6) = Cells(R, 6) + 1

```

```

End If
Next CC
Next C
  ***
  ***Identification of condition between Ri and
Max(Rj)
Sh = Com(R)
  ***Evaluation if Ri = Max(Rj)
If Sh = 1 Then
  Cells(R, 7) = Cells(R - 1, 7)
  Cells(R, 8) = Cells(R - 1, 8) + 1
  Cells(R, 9) = Cells(R - 1, 9) + 1
End If
  ***
  ***Evaluation if Ri > Max(Rj)
If Sh = 2 Then
  Cells(R, 7) = Cells(R - 1, 7) + 1
  Cells(R, 8) = Cells(R - 1, 8) + 1
  Cells(R, 9) = 0
End If
  ***
  ***Evaluation if Ri < Max(Rj)
If Sh = 0 Then
  Cells(R, 7) = Cells(R - 1, 7) + 1
  Cells(R, 8) = Cells(R - 1, 8) - Cells(R - 1, 9)
  Cells(R, 9) = 0
End If
  ***
  ***Evaluation if g = c
If Cells(R, 6) = 5 Then
  For I = 1 To 5
    Cells(20, I) = Cells(R, I)
  Next I
End If
  ***
End Sub

```

Above sub-procedures can be run either manually under the control of the user during the solution process or automatically by execution the sub-procedure Automation_Run:

```

Sub Automation_Run()
Do While Cells(20, 1) = ""
  Call Guess_code
  Call Feedback
Loop
End Sub

```

Figure 6 shows visualization of the color decoding simulation model.

	A	B	C	D	E	F	G	H	I	J	K	L	
1	Color decoding sheet												
2	Code vector c :						Response vector r_i :		Response summation:		Code		
3		4	8	5	7	2					Man Input		
4	Step:	Guess vectors g_i :					\hat{b}	w	$R_i = \hat{b} + w$			Rand Input	
5	1	1	2	3	4	5	0	3	3			Hide	
6	2	6	2	3	4	5	0	3	3				
7	3	1	6	3	4	5	0	2	2				
8	4	7	2	3	4	5	0	4	4				
9	5	7	8	3	4	5	1	3	4			Automatic	
10	6	7	2	8	4	5	0	5	5			Manual	
11													
12													

Fig. 6 Visualization of color decoding simulation model

F. Computer simulation model of order decoding

Computer simulation model created in VBA is visualized in *Order decoding* MS Excel sheet.

Vector data are saved into the cells:

- The input – code vector c (where the values of coordinates are already decoded) is entered and saved directly to cells in the range C3:G3;
- The output – matrixes G_i are saved in the ranges C5:G9, C10:G14 and C15:G19;
- Final evaluation is provided in the range C22:G24 and final result is shown in the range C25:G25.

The simulation program is divided into to two sub-procedures:

- **Permutation:** this procedure creates matrixes G_i by cyclic permutation of the vector g_i ;
- **Evaluation:** evaluates matrixes G_i and find all possible results.

The sub-procedure Permutation has the following code:

Sub Permutation(M)

****Creation of the first row matrixes G1

If M = 1 Then

For I = 1 To 5

Cells((M - 1) * 5 + 3, I) = Cells(2, I)

Next I

End If

****Creation of the first row matrixes G2

If M = 2 Then

For I = 1 To 5

Cells((M - 1) * 5 + 3, I) = Cells(2, 6 - I)

Next I

End If

****Creation of the first row matrixes G3

If M = 3 Then

Cells((M - 1) * 5 + 3, 1) = Cells(2, 2)

Cells((M - 1) * 5 + 3, 2) = Cells(2, 1)

Cells((M - 1) * 5 + 3, 3) = Cells(2, 3)

Cells((M - 1) * 5 + 3, 4) = Cells(2, 5)

Cells((M - 1) * 5 + 3, 5) = Cells(2, 4)

End If

****Creation of permutation of matrixes G_i

For P = 1 To 4

Cells((M - 1) * 5 + 3 + P, 1) =

Cells((M - 1) * 5 + 2 + P, 5)

For Q = 2 To 5

Cells((M - 1) * 5 + 3 + P, Q) =

Cells((M - 1) * 5 + 2 + P, Q - 1)

Next Q

Next P

****Function provided Position evaluation

Call Position Check

End Sub

Below is shown an example of the code of the sub-procedure Evaluation:

Sub Evaluation_3()

RE = 24

For R1 = 3 To 7

If Cells(R1, 6) > 0 Then

RE = RE + 1

For R2 = 8 To 12

If Cells(R2, 6) > 0 Then

For R3 = 13 To 17

If Cells(R3, 6) > 0 Then

For C = 1 To 5

If (Cells(R1, C) = Cells(R2, C))

And (Cells(R1, C) = Cells(R3, C)) Then

Cells(RE, C) = Cells(R1, C)

End If

Next C

End If

Next R3

End If

Next R2

End If

Next R1

****Procedure that checks if final result is found

Call Final_Evaluation

End Sub

Above sub-procedures run repetitively either under manually control of the user or automatically.

Figure 7 shows visualization of the order decoding simulation model.

	A	B	C	D	E	F	G	H	I	J	K	
1	Position decoding sheet											
2			Code vector c :									
3			4	8	5	7	2					
4	Steps:		Guess vectors g_i :					Response:				
5	1	Matrix G_1	7	2	8	4	5	b_i				
6	2		5	7	2	8	4	0				
7	3		4	5	7	2	8	1				
8	4	Matrix G_2	8	4	5	7	2	3				
9	5		2	8	4	5	7	1				
10	6		5	4	8	2	7	0				
11	7	Matrix G_3	7	5	4	8	2	1				
12	8		2	7	5	4	8	1				
13	9		8	2	7	5	4	0				
14	10		4	8	2	7	5	3				
15	11	Matrix G_3	2	7	8	5	4	0				
16	12		4	2	7	8	5	1				
17	13		5	4	2	7	8	1				
18	14		8	5	4	2	7	0				
19	15		7	8	5	4	2	3				
20												
21	Final Evaluation:											
22	Possible positions		4				8					
23					5	7	2					
24				8								
25	Result:		4	8	5	7	2					

Fig. 7 Visualization of order decoding simulation model

VIII. MODEL VERIFICATION

Validation of the conceptual models as well as verification of the simulation models of the process of solution of Mastermind game are realized by running of the simulation program.

Firstly, the validation and verification of the algorithm of the process of color decoding was done by means of execution the simulation program for all 56 possible input combinations (see (5)) of the code vector c .

Similarly, the validation and verification was realized for process of order decoding – the input was replaced by all 125 (see (7)) permutations.

It has been also discovered that result of the color decoding is reached on average in 5.2 steps and result of the order decoding is reached on average in 13.9 steps.

IX. 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. The system approach can be set as the default paradigm for a wide integration of the principles of the algorithm development into education. The paper emphasizes the fact that the algorithm development of the real processes should be supported by computer simulation in any programming language.

Solution of the Mastermind board game 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 Science of University of Hradec Kralove No. 2103.

REFERENCES

- [1] S. Hubalovsky, E. Milkova, "Modelling of a real situation as a method of the algorithmic thinking development", in *Proc. 6th WSEAS/IASME International Conference on Educational Technologies (EDUTE'10)*, WSEAS Press, Kantoui, Sousse, Tunisia, 2010, pp. 68–72.
- [2] S. Hubalovsky, E. Milkova, P. Prazak, "Modeling of a Real Situation as a Method of the Algorithmic Thinking Development and Recursively Given Sequences", *WSEAS Trans. on Information Science & Applications*, Issue 8, vol. 7, Aug. 2010, pp.1090-1100.
- [3] S. Hubalovsky, J. Sedivy, "Algorithm for determination of estimates of parameters of regression straight polyline with objective elimination of outliers", in *Proc. Information technology (ICIT 2010)*, Gdansk : Politechnika Gdanska, 2010, pp. 263-266.
- [4] V. Jehlicka, "Programming as a Method of Interdisciplinary Relations in Learning", *International journal of applied mathematics and informatics*, Vol. 4, No. 4, 2010, pp. 111-120.
- [5] S. Hubalovsky, "Modelling of real kinematics situation as a method of the system approach to the algorithm development thinking", *International journal of applied mathematics and informatics*, Vol. 4, No. 4, 2010, pp. 83-91.
- [6] S. Hubalovsky, M. Musilek, "Automatic cryptanalysis of the monoalphabetical substitution as a method of the system approach in the algorithm development thinking", *International journal of applied mathematics and informatics*, Vol.4, No.4, 2010, pp. 92-102.
- [7] J. Bailer, M. Daniela, "Tracing the Development of Models in the Philosophy of Science", *Magnani, Nersessian and Thagard*, 1999, pp. 23-40.
- [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, pp. 77–100.
- [9] J. A. Sokolowski, C. M. Banks, "Principles of Modeling and Simulation – A Multidisciplinary Approach", *Wiley Publication*, New Jersey, 2009, pp. 121-141.
- [10] V. Chvatal, "Mastermind", *Combinatorica*, Vol.3, 1983, pp. 325-329.
- [11] M. Mitchell, *MasterMind[®] Mathematics*, Emeryville: Key Curriculum Press, 1999.
- [12] M. Koyama, T. Lai, "An Optimal Mastermind Strategy", *Journal of Recreational Mathematics*, Vol.25, pp: 251–256.
- [13] "Mastermind (board game)". *Wikipedia, free encyclopedia* [www]. Available: [http://en.wikipedia.org/wiki/Mastermind_\(board_game\)](http://en.wikipedia.org/wiki/Mastermind_(board_game)).

Stapan 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 assistant professor on University of Hradec Kralove His scientific activities are system approach, modeling and simulation.