

# E-learning for Visual Programming Language

Ing. Zdena Dobešová, Ph.D.  
Department of Geoinformatics, Faculty of Science  
Palacký University  
Olomouc, Czech Republic  
zdena.dobesova@upol.cz

**Abstract** — Students at study branch Geoinformatics use several software products for the creation and analyzing spatial data. Software products belong to the special information systems named Geographic Information System (GIS). The practical lectures take place in the computer laboratory. Courses are supported by e-learning portal based on open source learning platform Moodle. E-learning portal of Department of Geoinformatics supports both bachelor study “Geoinformatics and Geography” both master study “Geoinformatics”. Some lectures have been created for lecturing visual programming languages. Visual programming languages in GIS are aimed for the batch automatic data processing in GIS. Skills with visual programming languages are important for advanced users in GIS. Visual programming languages are more comprehensible than textual programming languages. Students in second and third year are ready to learn it. They are also familiar with e-learning courses and blended learning. The presented e-learning lectures for visual programming language confirm the usability of the platform Moodle. The described experiences with the visual language programming supported by e-learning portal are based on several years of lecturing at Palacký University in Olomouc in the Czech Republic.

**Index Terms**— e-learning, Moodle, geoinformatics, visual programming, assignments, testing.

## I. INTRODUCTION

Utilization of Geographic Information Systems (GIS) is the main topic of study branch of Geoinformatics. Processing spatial data can be done interactively from a software interface by choosing operations step by step. Opposite solution is automatic repetition of steps by batch process in the program. Creation of the batch program as an advanced technique is counted among the expecting student knowledge in the field of geoinformatics. Batch processing could be designed by visual programming language or textual programming language. The course “Programming 2” is aimed to gain knowledge visual and textual programming. All course topics are supported by study materials under Moodle learning management system.

This article brings an overview of experiences with lecturing a visual programming language that have been lectured for seven years. Students put knowledge into effect in other followed courses.

## II. THE COURSE “PROGRAMMING 2”

The course “KGI/PRG2 Programming 2” is an obligatory subject in the fourth semester of a bachelor study

“Geoinformatics and Geography” [1]. Formerly it was the course “KGI/SKRI Scripting”. The scope of the subject is firstly creation models in ArcGIS ModelBuilder. Software ArcGIS by company Esri is one of most spread GIS software in practise. ModelBuilder is component for creation models [2]. Models are diagrams that represent in a graphical way a batch of processes for spatial analysing (Fig. 1). The second main theme of the course is programming in Python for the same product ArcGIS. Creation of models in ModelBuilder can be considered as a visual language. Initial theme - visual programming in ModelBuilder prepares student for textual programming in Python very well. Learning of programming in Python follows in the second part of the course.

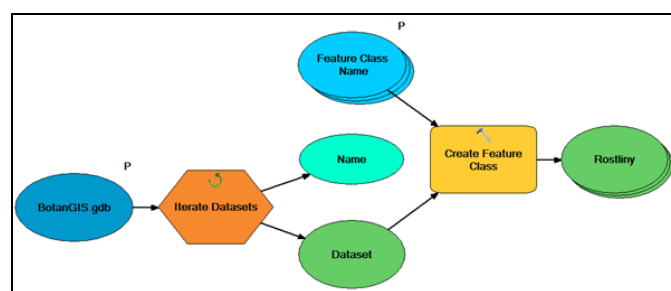


Figure 1. Example of visual program from ArcGIS ModelBuilder

The course is supported by a set of themes in a department e-learning portal that is based on the platform Moodle. There are about 22 different courses at this department e-learning portal that is used by teachers and students. The URL is <http://gislib.upol.cz/vyuka/>. Students of study branch of Geoinformatics are experienced in using Moodle early in the first semester. In following years, they use it regularly [3, 4].

Lectures, training data, tasks for home works and assignment module for submissions of home works are arranged in Moodle course with the same name “Programming 2”. The lecturing is divided to the three themes. Each of them consists of theoretical introduction that is created in Microsoft Power Point presentation. Several practical examples of models are created under each topic. Training spatial data (point, line and polygon feature classes) are downloadable in the form of Esri Geodatabase. The pages also contain the solutions of each practise in a form of Toolbox file. The final solutions – functional models are for comparison with student self-solutions. The final solutions of models help especially

them who use courses only in the distance learning (students under ERASMUS program, which stay abroad).

The course is open for public. Everybody can enrol as a guest. Some students of the master study who have a bachelor study level at another university in the Czech Republic can supplement this type of skills by a self-study with this e-learning portal. Several students spend one or two semesters abroad under ERASMUS exchange programme each year. The accessing all study material via e-learning portal is very appropriating for distance learning and preparing for the exam. Fast technology changes required gaining new knowledge after the graduation at the university. Further education is a necessity. Open Educational Resources (OER) can be the tools for its realization [5]. Therefore, e-learning course “Programming 2” can also use graduate students.

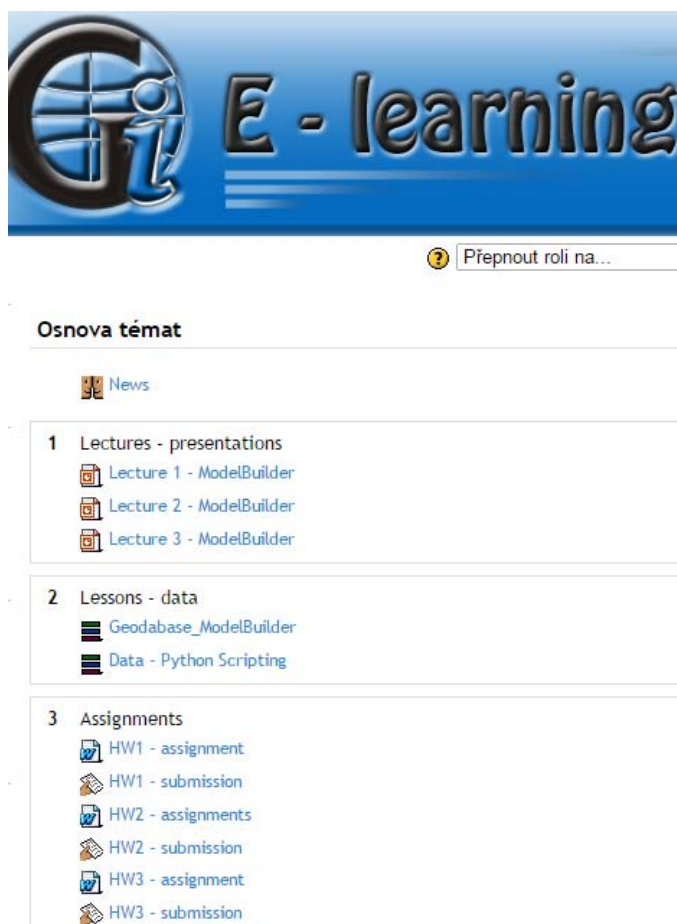


Figure 2. The content of the e-learning course for course “Programming 2”

### Visual programming in ModelBuilder

Visual programming is a way when the steps of the algorithm are designed in a graphical way. The opposite of visual of visual learning is a textual programming when an algorithm is written by textual instruction. Textual programming has very often a complicated syntax and semantics. Visual languages are very suitable for a novice programmer. The perception and cognition are very quick. The

construction of visual program is supported by a graphical editor. Connectors join graphical elements with arrow endings to express the orientation of dataflow.

Component ModelBuilder present at software ArcGIS from 2004. The component is improved especially in newer versions. New functionality as iterators, labelling was added. My task as a teacher is to teach students to be able design functional model with various functionalities.

### III. THE COURSE STRUCTURE

The teaching of ModelBuilder consists of three lectures. The frequency is one lecture per week. Each lecture takes 180 minutes at computer laboratory. About 25 students attempted to the course in academic year 2013/2014. The lectures about ModelBuilder are at the beginning of 13-weeks long summer semester. The lectures take three weeks. The following topics are textual programming in Python programming language for software ArcGIS in this course. The practical initial work in visual language ModelBuilder is a fundamental stage for following textual scripting.

The form of lecturing respects pedagogical principles. Firstly, detailed explanation presents the construction of the model. The practical part follows. The aim is the construction of functional models. New data as output data are stored in the source geodatabase.

An understanding of the model, reading and comprehension requires practise and repetition. It was the reason for additional home-works.

The Assignment module in Moodle is used for home works. An assignment has a “due date“, after which teachers can choose not to accept submissions any more [6]. It was set before the next lesson in the course “Programming 2”. The “Assignment 1” practices all new information from the “Lecture 1”. The due date of the “Assignment 1” is before the “Lecture 2”. The “Assignment 2” practices skills from the “Lecture 2” and once again the due date is before the “Lecture 3”. The workout of homework can be assumed as repetition before starting new lectures, to repeat all skills.

Only one disadvantage is for the teacher. Each year is necessary to set correct due date of assignments according to the time table of course “Programming 2”. The setting proper date is only small modification of course.

#### Assignments

The three homework assignments were added to the course in the two last year. The big advantage of Moodle is to arrange place for upload homework assignments. Students can upload their homework before the deadline (due date). Formally, I thought that the homework is not necessary for the student at university. However, the repetition of new skills is needful. The tasks in each homework are the nearly the same at the lecture but with different data and practise different spatial

functions. Each home works consist of three of four tasks – creation of three of four models.

Three home works are built as chain of tasks that follows one another. The second homework is based on some results from the first homework. The second homework extended some models from the first homework. Finally, the third homework extends the result models from the second homework. The outline of home works follows the idea that the models are extended and adopt new features of visual programming (parameters, iteration and variables).

The baseline of home works is creation and processing of a digital elevation model. Students have some knowledge about interpolation method for surface creation from the course “Digital Elevation Models“. The first homework is only about simple basic models. Firstly, students create a model that generates a digital elevation model by IDW method from the point input layer. Secondly, students extend the model by calculation of aspect and slope of digital elevation model (surface). Next model is a model that changes inputs to the parametrical model. Parametrical model is functional for various input data.

The second homework “Assignment 2” practices the using of a variable. Variable (%Input%) serve for the automatic naming of various output data to express the association of input and output data. Moreover, the new function iterator was added to calculate the same amount of function for a set of point input layers. The model goes through all input data in the geodatabase and automatically creates a set of surfaces for different input data (different area).

Finally, students practice different interpolation methods in the third “Assignment 3”. The result is a large model with 12 functions (12 yellow boxes) that creates surface by four different interpolation methods (Spline with tension, Regularized spline, Kriging, IDW). For each surface morphometric characteristics are calculated (aspect, slope, orientation). The previous skills are automatically used form “Assignment 1, 2”. They are parametric input and using of inner variable

Time management is an important issue in distance learning and fulfil of home works [7]. In the case of the first homework, some students had a problem with upload their homework before the deadline. The evaluation of homework by teacher and feedback including grades (high score) had positive motivation for student. The time management was better in case of the second and the third assignments. The homework has a good influence on study habits.

The homework with the better grade achievement has influence on positive fixation of knowledge. Positive fixation is necessary from the psychological view. The teacher must be a good instructor and use some supporting psychological tools as a positive fixation.

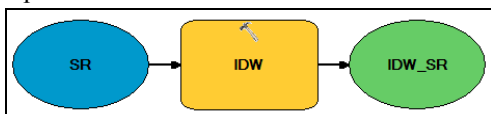


Figure 3. One result model from the Assignment 1

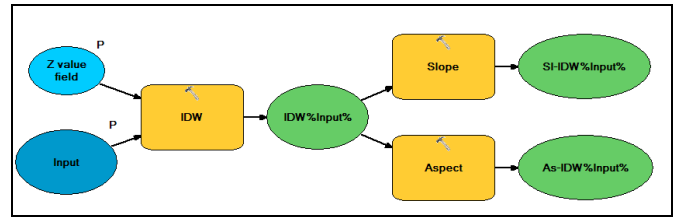


Figure 4. One result model from the Assignment 2

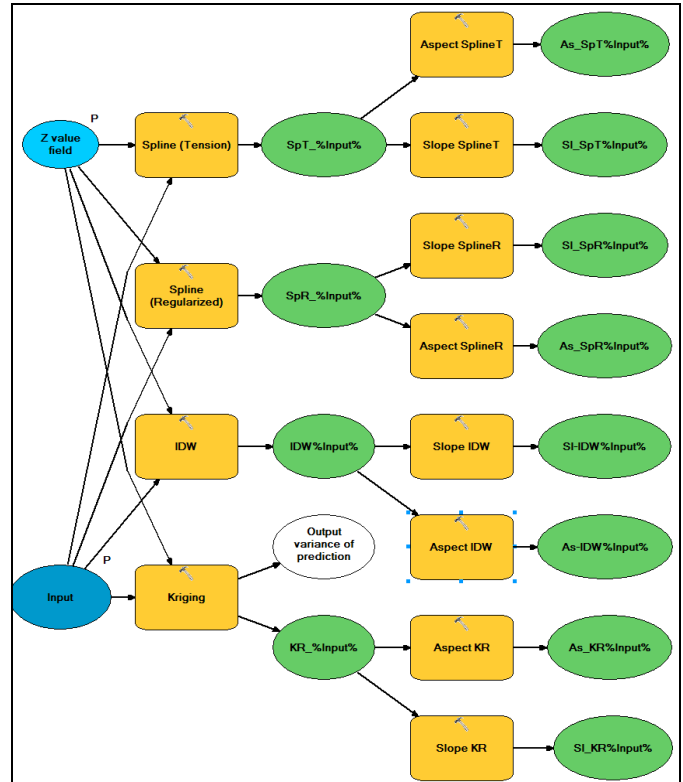


Figure 5. Result model from the Assignment 3

*Textual programming*

Robinson mentioned that programming courses are generally regarded as difficult [8]. Especially the textual programming is hard for novice programmers.

The programming for software ArcGIS has one interesting opportunity. User can automatically convert simple models designed in ModelBuilder to the Python programming language. Example of converted model is in Fig. 6. The conversion of the functional model is the way how to start the learning of programming [9]. This technique was utilized in the process of teaching of programming in the course “Programming 2”.

There were two ways how implement it:

*The first way:*

- Start with the creation of a simple model in ModelBuilder and convert it immediately to program in Python. Teacher explains techniques for creation model

and programming in Python simultaneously. There is a mixture of visual and textual programming for one task.

The second way:

- Firstly complete all three lectures about ModelBuilder and train creation of models very well. After finishing lecturing visual programming, start with converting the model to Python program. Continue only with lecturing of textual programming with the help of converted models.

Both ways were practically tested in the education process.

The second way was classical way in several initial years of lecturing. Experimentally the first way was tested for two years. To decide which way is better, the students fill the course evaluation by questionnaire at the end of the course “Programming 2”. Students assumed the first way as a worth way. The student comments mentioned difficulties in comprehension of visual and textual techniques simultaneously. The second way have been returned to the teaching process.

A converted model into a program is good inspiration how to start the textual programming [9]. The symbol iterator (orange hexagon in Fig. 1) introduces the idea about cycles, one of the basic ideas of programming. Equivalent textual command is “for:” or “while:” in Python.

```
# -----
# 3_Model_BuffRail_ParamVariable.py
# Created on: 2014-02-28 09:48:00.00000
# (generated by ArcGIS/ModelBuilder)
# Usage: 3_Model_BuffRail <Rail> <RB_Out> <Dist>
# Description:
# -----

# Import arcpy module
import arcpy

# Script arguments
Rail = arcpy.GetParameterAsText(0)
-if Rail == '#' or not Rail:
    Rail = "RailrdL"
    # provide a default value if unspecified

RailB_Out = arcpy.GetParameterAsText(1)
-if RailB_Out == '#' or not RailB_Out:
    RailB_Out = "D:\\RailrdBuff.shp"
    # provide a default value if unspecified

Distance_ = arcpy.GetParameterAsText(2)
-if Distanc_ == '#' or not Distance:
    Distance = "500 Meters"
    # provide a default value if unspecified

# Process: Buffer
-arcpy.Buffer_analysis(Rail, RailB_Out,
                      Distance, "FULL",
                      "ROUND", "NONE", "")
```

Figure 6. Program in Python automatically generated from ModelBuilder

#### IV. RESEARCH OF VISUAL NOTATION

Besides the lecturing, the research of cognitive level of graphical notation of dataflow diagrams in ModelBuilder was tested. Students and teachers assessed the graphical notation according to the theory "Physics" of Notations" [10, 11]. The author of the theory is Daniel Moody [12]. This theory consists of nine principles for graphical notations and brings recommendation for improving the notation. The reason for testing was the influence of cognition on the learning process. If the cognitive level is effective, the learning, comprehension and understanding is quicker and correct.

An alternative testing of graphical notation was processed in the department eye-tracking laboratory. The test consisted of twelve pictures of models (diagrams) that were showed to testers. Tester answered to the questions that were connected to each diagram. Respondent has to find a graphical element in diagram e.g. “Click on the parametric input in ModelBuilder.”, “Count the number of output data elements.”, “How many processes are in the diagram?”. The number of the correct answers, dwell time for each diagram expresses the comprehension of diagrams by respondents indirectly.

Gaze plot (scan path), attention (heat map), AOI (Area of Interest), time of answers are basic methods of the eye-tracking data investigation and visualization [13, 14]. The investigation of gaze plot discovered some reading habits of diagram. The left-right orientation of data flow is the same reading orientation as reading text in a book. The reading is very quick for that type of orientation. The main investigation was the assessment of graphical notation of ModelBuilder and comparison with results received by the theory "Physics" of Notations”.

A group of 30 under gradual students take part in that research testing. The time, correct answers, mouse click and gaze plot were recorded. The participating in research is interesting for students and motivation for deeper understanding of graphical notation and eye tracking testing. The result of the eye-tracking test could also be used as feedback for the improvement of the process of lecturing.

The theory "Physics" of Notations” discovered small visual differences between the symbols “Derived Value” and “Input Value”. Both symbols are ovals, and the color fill is nearly the same blue-turquoise color (Fig. 7). This similarity causes some difficulties in utilisation of inner variables in models that students mentioned in the evaluation questionnaire.

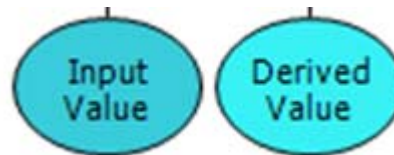


Figure 7. Graphical symbols for “Input Value” and “Derived Value” in ModelBuilder

The graphic elements of the diagram could be arranged and aligned to the grid. The button “Auto Layout“ aligns the



graphic symbol in ModelBuilder. Moreover, the connected lines are set to the same length. The diagram is more symmetrical and transparent. Resulting time and number of correct answers verified in eye-tracking test that a good arranging of diagram (aligning to the grid) is better in the process of reading than diagram without aligning. This result of the eye-tracking test was also reflected in the learning process. The teacher reminded the student to use “Auto Layout” button repetitively in the process of diagram creation. Auto aligning helps to create more readable diagram. The using of “Auto Layout” button must be a habit for student. The Fig. 8 shows a diagram without alignment. Diagram is a first draft before alignment. The perception of diagram is very chaotic. All other figures show diagrams with aligned graphical symbols (Fig. 1,3, 4, 5).

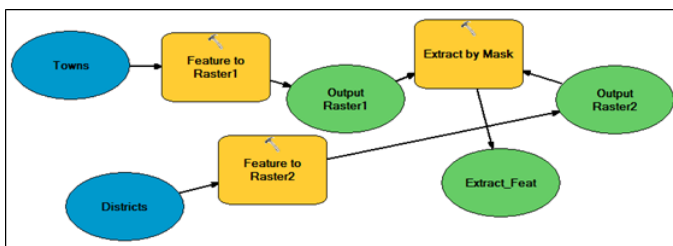


Figure 8. Diagram without alignment of symbols

## V. COURSE FINAL EXAMS

The final course exam consists of two parts. The first is the knowledge test. The second part is a practical programming on the computer when students have to create one model in ModelBuilder and one script in Python. They must pass both parts of the exam successfully.

The Moodle learning management system assists also in the knowledge testing that is the first part of the exam. Teachers can create a huge question bank in Moodle. Each test contains a set of questions from the question bank. Moodle offers about ten types of question [15]. The most appropriate types of questions were “Yes/No question”, “Multiple choice question” and “Essay question” in the course “Programming 2”. The essay type of question allows students to answer by writing text into the textbox. Moodle automatically grades the first two types. Teacher must manually grade the answers of the essay question.

Our experience proves that the mixture of all three types of questions is the best in one test. When the test contains only “multiple choice” questions, students only try to guess the answers. The essay question better verifies the student understanding to selected themes. Each exam test contains about 20 questions. The test lasts about 40 minutes. The time is not the crucial limitation for answers. The speed is not so important in the testing process; students have plenty of time to think about answers.

The mixture of question types is also advantaged in the grading by teacher. Moodle grades the Yes/No and multiple

choice questions automatically. Teacher goes only through answers of essay questions. The final sum of points is also calculated by Moodle.

The testing module is very helpful for the teacher. Time-consuming is a creation of the question bank. The questions must be understandable and punctual. The advantage is that questions can be used repetitively for next lecturing year.

## VI. OTHER VISUAL LANGUAGES IN GIS

Visual programming is not limited only for software ModelBuilder in ArcGIS. Student also used other visual languages in some software. Students utilize Workflow Designer in AutoCAD Map, Spatial Model Editor for ERDAS IMAGINE, Macro Modeler for IDRISI and Processing Modeler in QGIS in other geoinformatics courses [16]. The graphical notation is totally different in other visual languages that ModelBuilder in ArcGIS. However, the basic idea of automatic batch processing is the same for all of them.

The previous experience in ModelBuilder helps them to learn another visual language more quickly. The structure and teaching method of mentioned course “Programming 2” is fundamental for followed acquire skills in the area of new visual languages. When students obtain good practices and experience how to construct and arrange the graphic symbols in visual languages then they easily apply them to a new visual language. The most important experience is automation of spatial analyzing by batch process in visual language. From that point, the elaborated lecturing of ModelBuilder is crucial for future learning of other visual languages.

## VII. SUMMARY

Visual programming in GIS is a very useful way how to automatically process spatial data. The elaborated course “Programming 2” supports the quality of knowledge and student skills. The Moodle learning management system is used for dissemination of Power Point presentations, training data and solutions. In the phase of repetition, the assignment module in Moodle is used for submission of the three home works. The Moodle is also used for final testing in exam. Students took part in research of graphical notation in eye-tracking laboratory. The results of eye-tracking testing attracted students in deep acquiring the skills in visual programming languages.

## ACKNOWLEDGEMENT

Work was supported by the project **CZ.1.07/2.3.00/20.0166**.

## REFERENCES

- [1] Department of Geoinformatics. Bachelor programme Geoinformatics and Geography, Available at: <http://www.geoinformatics.upol.cz/ebestudy.php>
- [2] Esri Support. ArcGIS Help 10.2 - What is ModelBuilder? Available at: <http://resources.arcgis.com/en/help/main/10.2/index.html#/002w0000001000000>

- [3] Pechanec, V., Vávra, A. Education portal on climate change with web GIS client, *Journal of Cases on Information Technology*, Vol 15 (1), 2013, pp. 51-68.
- [4] Machar, I., Pechanec, V. Applying GIS to Conservation Education: Case Study Litovelske Pomoravi PLA , Ed. Lee, G: *Advances in Education Research*, 3rd International Conference on Education and Education Management (EEM), Singapore, 2013, Vol. 28, pp. 67-72.
- [5] Drozdova, M., Dado, M., Malcik, M., Mechlova, E. Open education at universities, quo vadis, *IEEE 11th International Conference on Emerging e<sup>l</sup>earning Technologies and Applications (ICETA)*, 2013, Stara Lesna, pp. 73 – 78, DOI: 10.1109/ICETA.2013.6674407.
- [6] Moodle: Assignment module, Available at: [https://docs.moodle.org/27/en/Assignment\\_module](https://docs.moodle.org/27/en/Assignment_module)
- [7] Çakiroğlu Ü. Analyzing the effect of learning styles and study habits of distance learners on learning performances: A case of an introductory programming course, *International Review of Research in Open and Distance Learning*, Vol. 15/ 4, 2014, pp. 161-185.
- [8] Robins, A., Rountree, J., Rountree, N. Learning and Teaching Programming: A Review and Discussion, *Computer Science Education*, Vol.13, No.2, Swets & Zeitlinger, pp. 137-172, 2003.
- [9] Dobesova, Z. Visual programming for novice programmers in geoinformatics, *Proceedings of SGEM 2012, 12th International Multidisciplinary Scientific GeoConference, Volume III*, STEF92 Technology Ltd., Sofia, Bulgaria, 433-440s. ISSN 1341-2704, DOI: 10.5593/sgem2012
- [10] Otevřelová A. Eye tracking evaluation of salience and interactive work in visual programming languages (Eye tracking vyhodnocení salience a interaktivní práce ve vizuálních programovacích jazycích), diploma thesis, Department of Geoinformatics, Palacký University, 2014, Olomouc.
- [11] Zieglerová J. Evaluation of eye tracking testing for visual programming languages, (Vyhodnocení eye tracking testování vizuálních programovacích jazyků), diploma thesis, Department of Geoinformatics, Palacký University, 2014, Olomouc.
- [12] Moody, D. L., The "Physics" of Notations: Toward a Scientific Basis for Constructing Visual Notations in Software Engineering. *IEEE Transactions on Software Engineering*, 2009. 35(6): p. 756-779.
- [13] Brychtová, A., Popelka, S., Dobesova Z. Eye-tracking methods for investigation of cartographic principles. *SGEM 2012 Conference Proceedings Volume II STEF92 Technology Ltd.*, Sofia, Bulgaria, 2012, pp.1041-1048, ISSN 1341-2704.
- [14] Holmquist, K., Nystrom, M., Andersson, R., Dewhurst, R., Halsyka, J. & Van de Weijer, J. (2011). *Eye tracking: A comprehensive guide to methods and measures*. Oxford University Press, 560 p.
- [15] Moodle: Question types, Available at: [https://docs.moodle.org/28/en/Question\\_types](https://docs.moodle.org/28/en/Question_types)
- [16] Dobesova Z. Data Flow Diagrams in Geographic Information Systems: A Survey, *Conference Proceedings SGEM 2014, 14th International Multidisciplinary Scientific GeoConference Volume I*, STEF92 Technology Ltd. Sofia, Bulgaria, 17-26 June 2014, pp. 541-548, ISBN 978-671-7105-10-0, DOI: 10.5593/sgem2014B21