Generic Assessment Rubrics for Computer Programming Courses

Aida MUSTAPHA, Noor Azah SAMSUDIN, Nureize ARBAIY, Rozlini MOHAMED, Isredza Rahmi HAMID
Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia
Parit Raja, 86400 Batu Pahat, Johor, Malaysia
{aidam, azah, nureize, rozlini, rahmi}@uthm.edu.my

ABSTRACT
In programming, one problem can usually be solved using different logics and constructs but still producing the same output. Sometimes students get marked down inappropriately if their solutions do not follow the answer scheme. In addition, lab exercises and programming assignments are not necessary graded by the instructors but most of the time by the teaching assistants or lab demonstrators. This results in grading inconsistencies in terms of the marks awarded when the same solution is being graded by different person. To address this issue, a set of assessment rubric is necessary in order to provide flexibility for critical and creative solutions among students as well as to improve grading consistencies among instructors and teaching assistants or demonstrators. This paper reports the development of assessment rubric for each domain in computer programming courses; cognitive, psychomotor, and affective. The rubrics were then implemented for one academic semester consisting of 14 weeks. An interrater reliability analysis based on Kappa statistic was performed to determine the consistency in using the rubrics among instructors The weighted kappa is 0.810, therefore, the strength of agreement or the reliability of the rubric can be considered to be 'very good'. This indicates that the scoring categories in the rubrics are well-defined and the differences between the score categories are clear.
Keywords: Scoring, assessment rubric, computer programming, cognitive, psychomotor, affective, Kappa statistics.

INTRODUCTION
Grading programming assignments and projects are similar to grading traditional assignments such as written essays. The primary distinctions between them are the unique keywords or constructs across different programming languages and the diverse possible solutions associated with a particular problem solving techniques. Traditional assessment for computer programming assignments and projects usually depends on an answer scheme that includes the source code as a model answer with marks allocated to specific lines of code. This model answer is then used by the instructors to allocate marks to the students' programs based on the provided source code in the answer scheme.

The problem with the traditional schema-based approach of awarding marks according to a "point-per-correct-statement" is that students are being graded based similarity of their solution to the answer scheme. This leads to little or no consideration given to creativity and originality in the student solutions. In programming, the same problem can usually be solved using different constructs but still producing the same output. Students often get marked down inappropriately if their solution is not exactly the same as the instructor's solution or alternatively marked up if their solution is similar to the provided solution. In addition, lab exercises and programming assignments are not necessary being graded by the instructors but most of the time by the teaching assistants or lab demonstrators. This results in grading inconsistencies in terms of the marks awarded when the same solution is being graded by different person. Instructors, for example, may emphasize on the design of the solutions. Demonstrators, on the other hand, may emphasize on the programming syntax.

To address this issue, a set of assessment rubric is necessary in order to provide flexibility for critical and creative solutions among students as well as to improve grading consistencies among instructors and teaching assistants or demonstrators. The literature has revealed that strategies used to grade programming assessments has evolved from grading students based on an answer scheme where marks are allocated to individual programming statements to a more holistic and inclusive methodology using rubrics. A rubric is a set of ordered categories to which a given piece of work can be compared. Scoring rubrics specify the qualities or processes that must be exhibited in order to assign a particular evaluative rating for a performance (McDaniel, 1993). As a grading tool, rubrics have successfully enable the instructors to assess the student's understanding and creativity to produce a solution in programming courses (Becker, 2003; Ahoniemi and Karavirta, 2009; Payne et al., 2012) as well as evaluating research skills in strategic management (Whitesell and Helms, 2013), ethical behavior (Carlin et al., 2011), critical thinking in engineering (Ralston and Bays, 2010; Loon and Lao, 2014), and reflective writing in medicine (Wald et al., 2012).

This study hypothesizes that rubrics provide the necessary structure and guidance that enable instructors to award marks as a whole for students' ability in problem solving, creativity, and aesthetics of any graphical user interface as well as the use of good programming practice and standards. The central focus of this research will be on creating a set of rubrics as a benchmark to measure student learning outcomes in introductory computer programming courses offered by the Faculty of Computer Science and Information Technology (FCSIT) at Universiti Tun Hussein Onn Malaysia (UTHM). At present, UTHM has to cope with very large first year classes with average of 70 students per section with multiple sections to cater four specializations of undergraduate Computer Science programs: Software Engineering, Information Security, Web Technology, and Multimedia Computing. This necessitates for more than one instructor and teaching assistants for lab sessions in each program. Due to the high number of student enrollment and diverse background of the instructors or demonstrators, grading lab assignments and group projects is particularly a challenge especially in ensuring fair delivery to all students.

The main goal for this study is to promote critical and creative thinking skills and to improve grading consistencies in programming subjects by introducing a generalized programming rubric to be used across all programming languages such as C, C++, and Java. The outcome of this research is able to increase the effectiveness in teaching and learning activities in terms of consistent assessment of the course learning outcomes. The rubric developed in this study is presented in the section following the related works. Next, the research methodology is detailed out to explain the validation process of the developed rubrics followed by the findings. Finally, the paper is concluded with some indication for future research.

RELATED WORK

The Outcome-based Education (OBE) system emphasizes the importance of a curriculum content to be driven by learning outcomes (Spady, 1994). In OBE, the learning outcomes are expressed as statements of knowledge and skills individual students should possess at the end of the course they enrolled. An OBE system offers a comprehensive approach to organize and operates an education system that is focused on successful demonstration of learning sought from students at the end of the learning cycle (Murphy and Duncan, 2007).

The OBE system has been introduced to the Faculty of Computer Science and Information Technology (FCSIT) at Universiti Tun Hussein Onn Malaysia (UTHM) since 2004. The learning outcomes of a program are set by various level of academic management team at FCSIT. There are three primary components of the OBE system; Program Educational Outcome (PEO), Program Learning Outcome (PLO), and Course Learning Outcome (CLO). The PEO expresses statements of long term objectives that describe what a Computer Science should be able to demonstrate as a result of attending its program. Clearly, the achievement of the PEO at faculty level is geared to the achievement of the vison and mission of UTHM. Table 1 shows the PEO for one of the Computer Science undergraduate program offered at FCSIT, which is the Bachelor of Computer Science (Software Engineering).

Table 1: Program Educational Outcome (PEO).

PEO 1	Apply basic knowledge, principles and skills in the field of Computer Science to meet the job specification. (Knowledge / Practical Skills)
PEO 2	Implement the responsibility for solving problems analytically, critically, effective, innovative and market-oriented. (Critical Thinking and Problem Solving / Life-long Learning and Information Management / Enterpreneurship Skills)
PEO 3	Acts effectively as an individual or in a group to convey information within the organization and community. (Team Working Skills / Communication Skills)
PEO 4	Practicing good values and ethics in a professional manner in the community and able to act as a leader. (Profesional, Social, Ethics, and Humanity / Leadership Skills)

The PEO statements are further refined to establish PLO. The PLOs highlight individual student's abilities that reflect their learning experiences at FCSIT. In addition, the management team of FCSIT is also required to consider the general learning objectives set by the Malaysian Qualifications Agency (MQA, 2008) and the Ministry of Higher Education (MOHE) in expressing the PLO. As a result, the PLO are expressed to satisfy components of MQA standards which include knowledge, practical skills, communication, critical thinking and problem solving, teamwork, life-long learning and information management, entrepreneurship, moral, professional and ethics and finally leadership. Students of the undergraduate programs at FCSIT are expected to acquire the PLO upon completion of their studies. The implementation of the PLO is he PLO is then distributed across individual courses in the undergraduate programs. Table 2 shows the PLO for Computer Science programs at FCSIT.

Table 2: Program Learning Outcome (PLO).

PLO 1	Applying knowledge and understanding of essential facts, concepts, principles and theories in the field of Computer Science Software Engineering. (Knowledge – K)
PLO 2	Implementing Software Engineering knowledge in analyzing, modeling, designing, developing and evaluating effective computing solutions. (Practical Skill – PS)
PLO 3	Communicate in spoken and written form in order to convey information, problems and solutions to the problems effectively. (Communication – CS)
PLO 4	Analyze the appropriate techniques in the field of Software Engineering to solve problems using analytical skills and critical thinking. (Critical Thinking, Problem Solving – CTPS)
PLO 5	Demonstrate teamwork skills, interpersonal and social effectively and confidently. (Team Work – TS)
PLO 6	Using the skills and principles of lifelong learning in academic and career development. (Life Learning and Information Management – LL)
PLO 7	Fostering entrepreneurship in career development. (Enterpreneurship – ES)
PLO 8	Adopt values, attitudes and responsibilities in a professional manner from ths aspects of sosial, ethics and humanity. (Moral, Professional and Ethics – EM)
PLO 9	Effectively carry out the responsibilities of leadership. (Leadership – LS)

The PLOs serve as the basis of determining the course learning outcomes (CLO) for every course offered. Each set of programming CLO in the course syllabus is mapped to the PLO of FCSIT. The mapping is known as CLO-PLO matrix. The CLO shall be constructed in such a way to accommodate the PLO. The establishment of the CLO in programming courses applies principles of Bloom's Taxonomy which covers three learning domains outlined by MQA standard: cognitive, affective, and psychomotor (Bloom et al., 1994). Table 3 presents the complete set of levels in each domain.

Table 3: Levels in cognitive, psychomotor, and affective domain based on Bloom's taxonomy.

Level	Cognitive Domain
C1	Knowledge (KN)
C2	Comprehension (CO)
C3	Application (AP)
C4	Analysis (AN)
C5	Synthesis (SY)
C6	Evaluation (EV)

Level	Psychomotor Domain
P1	Perception
P2	Set
P3	Guided response
P4	Mechanism
P5	Complex overt response
P6	Adaptation
P7	Origination

Level	Affective Domain
A1	Receiving phenomena
A2	Responding to phenomena
A3	Valuing
A4	Organizing values
A5	Internalizing values

Eventually, to measure the achievement of cognitive, psychomotor, and affective domain in each CLO, a student is evaluated using one to five assessment tools: quiz, test, laboratory assignments, project, and final exam. Each of the assessment tool is assigned to ensure positive achievement for the courses. Indeed, such information has implication on the achievement of CLO and PLO that are usually evaluated at the end of the learning process. Table 4 shows a sample of specification table to evaluate the cognitive domain in an object-oriented programming course. The specification table is designed to plan the distribution of marks based on taxonomy level mapping. Such constructive mapping is valuable to evaluate how the CLO and PLO are evaluated and related and finally implies the PEO.

Table 4: A specification table for an object-oriented programming course.

Question No.	Course Content/ Topic	Marks Distribution based on Bloom's Taxonomy						Subtotal
		KN	CO	AP	AN	SY	EV	
		Level 1		Level 2		Level 3		
Q1 (a)	Chapter 2: Primitive data types	3						24
Q1 (b)	Chapter 3: Fundamental of OO	6						
Q1 (c)	Chapter 3: Fundamental of OO	6						
Q1 (d)	Chapter 4: Object and classes					9		
Q2 (a)	Chapter 3: Fundamental of OO				12			27
Q2 b)	Chapter 3: Fundamental of OO				15			
Q3 (a)	Chapter 5: Inheritance and polymorphism		5					25
Q3 (b)	Chapter 5: Inheritance and polymorphism			20				
Q4 (a)	Chapter 4: Object and classes				5			24
Q4 (b)	Chapter 4: Object and classes					10		
Q4 (c)	Chapter 4: Object and classes					9		
Subtotal based on taxonomy (Marks)		15	5	20	32	28	0	100
Subtotal for each level (Marks)		20		52		28		40%
Cognitive level (%)		20%		52%		28%		100%
Distribution of cognitive level (%)		5%		35%		60%		100%

At FCSIT, the specification table is used to assess only the cognitive domain via quizzes, tests, and final exams. The assessment method is still using the answer scheme. However, assessments for lab assignments and projects are not necessary being graded by the instructors but most of the time by the teaching assistants or lab demonstrators. This calls for the need of a generalized rubric to cover all continuous learning assessments other than tests and final exams.

RESEARCH METHODOLOGY

A rubric is a set of categories developed based on a specific set of performance criteria. As an assessment tool, a rubric should cover all learning domains offered in computer programming courses. The purpose of such classification is to categorize different objectives that educators set for the students because educators have to focus on all three domains to create a more holistic form of delivery. In order to develop the rubric, the first step is to identify the learning outcomes at the program level followed by the course level before the types of assessments could be determined. The rubric can then be developed for a specific type of assessment such as lab assignments or group projects. In this study, the rubric development and validation process are founded on the principle of continuous feedback and improvement involving the following steps:

Step 1: Identify Program Learning Outcomes (PLO) and Course Learning Outcomes (CLO)

From the curricula, all programming courses are selected involving different languages (i.e. C, C++, Java). The PLOs and CLOs for each course were tabulated and compared. At FCSIT, UTHM, each course has three CLOs in average. Next, the assessment types were determined across all the courses and the percentage of each assessment type according to the PLO and CLO were distributed. Again, the types of assessment include tests, assignment, practical/lab, group project and final examination. Table 5 shows the mapping of PLOs and CLOs across all programming courses. The types of assessments are also indicated for each learning objective.

From the list of assessment methods provided in the table, quiz, test, and final examinations in CLO1 are graded based on traditional schema-based approach because the tools are only assessing the cognitive learning domain in computer programming. Lab assignments (CLO2) and projects (CLO2, CLO3), however, are designed to assess all three learning domains; cognitive, psychomotor, and affective. Because each CLO assess only one learning domain, the rubrics developed will be categorized according to the CLO. For each CLO, the level of domain for cognitive, psychomotor, affective are also assigned.

Table 5: Mapping of course learning outcomes to program learning outcomes across all programming courses.

		Program Learning Outcome (PLO)									
		Knowledge	Knowledge & Practical	Communication Skills	Critical Thinking & Problem Solving	Team Working Skills	Life-long Learning	Entrepreneurship Skills	Professionalism, Social, Ethics and	Leadership Skills	
Course Learning Outcomes (CLO)		PLO 1	PLO 2	PLO 3	PLO 4	PLO 5	PLO 6	PLO 7	PLO 8	PLO 9	Assessment
CLO 1	Design problem solving process based on object oriented concept.				C5						Quiz, Test, Lab, Project, Final Examination
CLO 2	Construct an object oriented computer application using Java programming language.		P4								Lab, Project
CLO 3	Demonstrate the implementation of object oriented concept using any high level programming language.						A3				Project Presentation

Step 2: Formulate the rubric

In formulating the rubric, one or more dimensions that serve as the basis for judging the student work were determined. Each CLO was broken into one or more objectively measurable performance criteria along with its sub-criteria. The basic dimension in the rubric is the assessment type, whether delivered by the students in the form of written reports or via presentation. Next, for each dimension, a scale of values from 1 to 5 on which to rate each dimension is assigned; 1 is being very poor, 2 is poor, 3 is fair, 4 is good, and 5 is excellent. Finally, within each scale, the standards of excellence for specified performance levels accompanied were provided. Table 6 to Table 8 show the rubric for CLO1 (cognitive), CLO2 (psychomotor), and CLO3 (affective), respectively.

Table 6: Rubric for CLO1. Design problem solving process using algorithm/object-oriented concepts (Cognitive – C5, PLO4 – CTPS).

Assessment	Criteria	Sub-criteria	Level	1	2	3	4	5
Report	Ability to analyze problem and identify requirements	Identify correct input/ output	C2	Unable to identify any input and output	Able to identify only one input or output	Able to identify correctly some input and output	Able to identify correctly all input and output	Able to identify correctly all input and output and provide alternative
	Ability to	Construct	C3	Unable	Able to	Able to	Able to	Able to

	demonstrate design solution	correct flowchart or pseudocode		to construct	construct but mistake on symbol	construct correctly	construct correctly and use proper elements	construct correctly, use proper elements and documentation

Table 7: Rubric for CLO2. Construct a computer application/object oriented computer application using object:- oriented concepts (Psychomotor – P4, PLO2 – Practical Skill)

Assessment	Criteria	Sub-criteria	Level	1	2	3	4	5
Report	Ability to apply required data type or data structure	Appropriate choice of variable names or data structure (i.e. array/ linked list)	P3	Unable to identify required data type or data structure	Able to identify required data type or data structure but does apply correctly	Able to apply required data type or data structure but does not produce correct results	Able to apply required data type or data structure and produce partially correct results	Able to apply required data type or data structure and produce correct results
	Ability to apply required control structure	Correct choice of sequential, selection or repetition control structure	P4	Unable to identify required control structure	Able to identify required control but does apply correctly	Able to apply required control structure but does not produce correct results	Able to apply required control structure and produce partially correct results	Able to apply required control structure and produce correct results
	Ability to run/debug	Free from syntax, logic, and runtime errors	P3	Unable to run program	Able to run program but have logic error	Able to run program correctly without any logic error	Able to run program correctly without any logic error and display inappropriate output	Able to run program correctly without any logic error and display appropriate output
	Ability to perform input validation	Validate input for errors and out-of-range data	P3	The program produces incorrect results	The program produces correct results but does not display correctly Does not check for errors and out- of- range data	The program produces correct results but does not display correctly. Does little check for errors and out- of- range data	The program works and meets all specifica-tions. Does some checking for errors and out- of- range data	The program works and meets all specifica-tions. Does exceptional checking for errors and out- of- range data
Presentation	Ability to	Comment /	P1	No	Docume	Docume	Document	Document

n	produce readable program	Description		documentation	ntation is simple comment in code	ntation is simple comments embedded in code with header separating the codes	ation is simple comments and header that useful in understanding the code	ation is well-written and clearly explains what the code is accomplishing
		Indentation / Naming Convention	P2	Unable to organize the code	The code is poorly organized and very difficult to read	The code is readable only by a person who already knows its purpose	The code is fairly easy to read	The code is extremely well organized and easy to follow

Table 8: Rubric for CLO3. Demonstrate the implementation of problem solving process/object-oriented concepts using high-level programming language (Affective – A3, PLO6 – Lifelong Learning)

Assessment	Criteria	Sub-criteria	Level	1	2	3	4	5
Presenta-tion	Ability to demonstrate program in group	Demonstrate understand-ing on program design	A3	Unable to explain program design	Able to explain a little program design	Able to explain some program design	Able to explain entire program design correctly as it is	Able to explain program design correctly and provide alternative solutions
		Organization of group presentation	A4	Materials are not organized with missing infor-mation	Materials are partially organized with missing infor-mation	Materials are partially organized with required infor-mation	Materials are highly organized with required infor-mation	Materials are highly organized with additional infor-mation
		Cooperation from all members	A2	Unable to cooper-ate in a group	Forced coopera-tion through interven-tion	Demon-strate coopera-tion after interven-tion	Demon-strate coopera-tion through personal dominance	Demon-strate coopera-tion through group hierarchy

The rubrics have been developed as a 2D grid in Microsoft Excel sheet, where each row describes one evaluation criteria and the columns indicate the level of achievement. Since the rubric is already in an Excel form, the instructors simply fill in the student performance according to the desired column and the form will add up the corresponding values to produce a final score.

Step 3: Test the reliability of the rubric

Reliability refers to the consistency of assessment scores. On a reliable test, a student would expect to attain the same score regardless of when the student completed the assessment, when the assessment was scored, and who scored the assessment. In order to measure the reliability of the rubrics, the rater reliability in the form of reliability coefficient is measured. Raters reliability refers to the consistency of scores that are assigned by two independent raters (inter-rater reliability) and that are assigned by the same rater at different points in time (intra-rater reliability) (Moskal and Leydens, 2000). According to Jonsson and Svingby (2007), the consensus agreement among raters depends on the number of levels in the rubric, whereby fewer levels lead to higher chance of agreement.

This study adopted the measurement of inter-rater reliability based on Kappa statistics (Cohen, 1960). In Cohen's kappa, values between 0.4 and 0.75 represent fair agreement beyond chance. Values ≤ 0 as indicating no agreement and 0.01–0.20 as none to slight, 0.21–0.40 as fair, 0.41– 0.60 as moderate, 0.61–0.80 as substantial, and 0.81–1.00 as almost perfect agreement (McHugh, 2012).

EVALUATIONS

The rubrics developed in this study was implemented in three programming courses are offered during the First Semester of 2015/2016. The courses were Computer Programming (BIT10303) using C programming language, Object-Oriented Programming (BIT20603) using C++ programming language, and Java Programming (BIT33803). The rubrics were consistently used for grading lab assignments and group projects throughout the 14-week period of the semester. All the assignments and projects were graded independently by two random instructor or lab demonstrator using the same rubric. Table 9 shows the total number of students works/artifacts being compiled and graded based on the rubrics.

Table 9: Summary of total written artifacts graded using the rubrics. The artifacts for lab assignments and groups projects are in the form of source codes.

Course	No. of Students (a)	No. of Instructors/ Demonstrators (b)	No. of Lab (c)	No. of Assignments (d)	No. of Projects (e)	Total Artifacts (a * (c + d + e))
BIT10303	60 (S1) + 37 (S2) = 97	2	9	1	1	1,067
BIT20603	73 (S1) + 37 (S2) = 110	2	7	1	1	990
BIT33803	76 (S1) = 76	1	5	0	1	456
					Total	2,513

*S*i* indicate section number.

Based on Table 9, all sets of scores (i.e. four sets for BIT10303, two sets each for BIT20603 and BIT33803) are then statistically analyzed for inter-rater reliability using the Cohen's Kappa (Cohen, 1960). According to this metric, a Kappa of 1 indicates a perfect agreement, whereas a kappa of 0 indicates agreement equivalent to chance. The analysis was performed using the program Statistical Package for the Social Sciences (SPSS), version 20.0. Note that the instructors or demonstrators are referred as raters in calculating the kappa values. Two raters were randomly picked to evaluate the each artifact. Table 10 presents the results for both raters on every artifact.

Table 10: Assessment results for 2,513 artifacts by two independent raters.

	Rater #2					
Rater #1	1 (very poor)	2 (poor)	3 (fair)	4 (good)	5 (excellent)	Total
1 (very poor)	364	207	0	0	0	571
2 (poor)	161	349	55	1	0	566
3 (fair)	0	6	295	108	2	411
4 (good)	0	1	18	312	109	440
5 (excellent)	0	0	3	107	415	525
	525	563	371	528	526	2,513

Based on Table 10, the total number of observed agreements is 735, which constitutes 69.04% of the observations. The number of agreements expected by chance is 509.1, which is 20.26% of the observations. The kappa value is 0.612 with 95% confidence interval from 0.589 to 0.634. Based on the kappa value, the reliability of the rubrics is considered to be 'good' based on the strength of agreement between the two raters.

However, this calculation only considered exact matches between the two raters. Since the scale of dimensions (very poor, poor, fair, good, excellent) are ordered, close matches were also being considered. This means if the first rater assessed an artifact as fair and the other as good, this is closer than if the rater assessed the artifact as poor and the other excellent. The calculation of weighted kappa assumes the categories are ordered and accounts for how far apart the two raters are. The weighted kappa is 0.810, therefore, using this approach the strength of agreement or the reliability of the rubric can be considered to be ‘very good’. This indicates that the scoring categories in the rubrics are well-defined and the differences between the score categories are clear.

CONCLUSIONS

A generic programming rubric is proposed to be used across all programming courses offered by FCSIT at UTHM involving a variety of high-level programming languages such as C, C++, and Java. The rubrics are shared with the students every time a lab exercise or assignment is assigned to help them better understand the balance of the different activities in their final grade. From the rubrics, students are able to estimate the amount of effort that are required to achieve the perfect score. In this way, students are also playing active role of becoming independent in determining their own learning objectives. In the future, the rubrics will be used in establishing benchmarks for the programming courses and analyzing student performance to improve the learning and learning process including making adjustments to the curriculum.

ACKNOWLEDGEMENT

This project is sponsored by the Contract Research Grant from the Centre for Academic Development and Training (CAD) at Universiti Tun Hussein Onn Malaysia (UTHM).

REFERENCES

Ahoniemi, T. & Karavirta, V. (2009) Analyzing the use of a rubric-based grading tool. *14th Annual ACM SIGCSE Conference on Innovation and Technology in CS Education* (pp.333-337).

Becker, B. (2003). Grading programming assignments using rubrics. *8th Annual Conference on Innovation and Technology in Computer Science Education* (pp.253-253). ACM, New York, NY, USA.

Bloom, B. S., Anderson, L., & Sosniak, L. (1994). “*Bloom’s taxonomy*: *A forty-year retrospective*”. Assessing Scholarly. Chicago: NSSE. Ball, CE (2012).

Carlin, N., Rozmus, C., Spike, J., Willcockson, I., Seifert, W., Chappell, C., Hsieh, P.-H., Cole, T., Flaitz, C., Engebretson, J., Lunstroth, R., Amos, C., & Boutwell, B. (2011). The health professional ethics rubric: Practical assessment in ethics education for health professional schools. *Journal of Academic Ethics*, vol. 9, no. 4 (pp.277-290).

Cohen, J. (1960). A coefficient for agreement for nominal scales. *Education and Psychological Measurement*, vol. 20 (pp.37-46).

Herman, J.L. (1992). *A practical guide to alternative assessment*. Association for Supervision and Curriculum Development, 1250 N. Pitt Street, Alexandria, VA 22314.

Jonsson, A. & Svingby, G. (2007) The use of scoring rubrics: Reliability, validity and educational consequences. *Educational Research Review*, vol. 2 (2007) (pp.130-144).

Loon, J.E.V. & Lai, H.L. (2014). Information literacy skills as a critical thinking framework in the undergraduate engineering curriculum. *2014 ASEE North Central Section Conference.*

McDaniel, E. (1993). *Understanding educational measurement*. Dubuque, IA: William C. Brown.

McHugh, M.L. (2012) Interrater reliability: the kappa statistic. *Biochem Med*, vol. 22, no. 3 (pp.276–282).

Moskal, B.M. & Leydens, J.A. (2000). Scoring rubric development: validity and reliability. Practical Assessment, *Research & Evaluation*, 7(10).

Malaysian Qualification Agency (MQA). (2008). Code of Practice for Programme Accreditation.

Murphy, J.J. & Duncan, B.L. (2007). Brief intervention for school problems (2nd ed.): Outcome-informed strategies. New York: Guilford Press.

Payne, N., Kolb, D., & Kotze. G. (2012). “Scheming” to optimize marking in computer programming: From memos to rubrics. *ICERI 2012* (pp.869-877).

Ralston, P. & Bays, C. (2010). Refining a critical thinking rubric for engineering. *American Society for Engineering Education*.

Spady, W.G. (1994). Outcome-based education: Critical issues and answers. American Association of School Administrators, 1801 North Moore Street, Arlington, VA 22209.

Wald, H.S., Borkan, J.M., Taylor, J.S., Anthony, D., & Reis, S.P. (2012). Fostering and evaluating reflective capacity in medical education: Developing the REFLECT rubric for assessing reflective writing. *Academic Medicine*, vol. 87, no. 1 (pp.41-50).

Whitesell, M. & Helms, M.M. (2013). Assessing business students’ research skills for the capstone project in the strategic management course. *Journal of Business & Finance Librarianship*, vol. 18, no. 1.

Modeling Behavior of Students in E-Learning Courses on the Basis of Use Interactive Animations

Martin Magdin
Department of Computer Science, Faculty of Natural Science, Constantine the Philosopher University in Nitra, Tr. A. Hlinku 1, 949 74 Nitra
mmagdin@ukf.sk

Milan Turčáni
Department of Computer Science, Faculty of Natural Science, Constantine the Philosopher University in Nitra, Tr. A. Hlinku 1, 949 74 Nitra
mturcani@ukf.sk

ABSTRACT
Authors in their contribution deal with modeling the behavior of user in e-learning course based on the use of interactive animations. Nowadays, E-learning courses form a standard part of educational process. However, it is not so easy to determine the way students work with study material, whether they make use of it in order to increase didactic effectiveness of e-course. In the contribution authors point to the non-traditional method of recording students´ activities and reverse transition to previous lessons using interactive animations, which have been implemented into the study material. The method of recording students´ activities was implemented in the academic years 2009/2010 through 2013/2014. Students were divided into two groups – experimental and reference ones. The reference group did not use interactive animations, while in the experimental group interactive animations were implemented into the study material.
Keywords: behavior of students, interactive animations, interactive matrix, transition of e-learning course.

INTRODUCTION
Computer (interactive) animations may help concretise abstract, complex concepts and phenomena in science education, thus helping students to learn more easily and more effectively (Akpinar, 2013). The questions arises as to what the reasons are which have made interactive animations a vital part of modern ecurricula, and whether there is empirical evidence to support claims that using multimedia and interactivity in e-curriculum has positive impact to cognitive development and academic achievement at students (Pinter et al., 2012). Part of study of Informatics and Information technologies in higher education, in addition to programming is eg. graphics, theory of formal languages and automata and often the different subjects with a focus on the area of computer hardware. For understand the mutual action of the individual components PC must students handle basic physical principles. This specific area of hardware is called Logical systems of computers and the students applied their knowledge not only from mathematics but also physics, acquired his studies at secondary school. Abstract and complex concepts are especially difficult for students to grasp in the traditional learning environment using traditional teaching methods. As a result, learners at different levels and ages have difficulty understanding science concepts (Chiu et al. 2002). The reasons for these difficulties have some common features such as the students' varying levels of comprehension for science concepts. This variability is true for many fundamental concepts in all branches of science such as physics, chemistry and biology (Akpinar, 2013). The thorough investigation by Sekular and Blake (1990) into how students take in information, how they learn pointed out that the learning process takes place primarily by way of sight, and since it is the most vital of our senses, it is also the most highly-developed one. It enables a person to gather information from one's surroundings, analyze these and then decide how to process based on the deduced data. Graphical representations are defined as visual aids that act as supplement to any other textual information and will concentrate learners' attention (Mayer, 1989). Such representations will have maximum effect when accompanying some learning material that is (relatively) new to the learner (Mayer and Gallini, 1990). This is especially the case with computer animation that is designed to aid long-term learning in the form of focusing learners on certain objects in the beginning (Pinter et al., 2012).

METHODOLOGY OF RESEARCH
This study investigates the effects of using interactive animations based on predict-observe-explain as a presentation tool on students' (University students) understanding of the static electricity and concepts of electronic circuits (area of computer hardware). A quasi-experimental pre-test/post-test control group design was utilized in this study. This Experiment was realized in the academic years 2009-2014 (Winter semester). The

experiment group consisted of 20 students, and the control group also of 20 students. The control group worked by normal instruction in which the teacher provided instruction by means of lecture, discussion and homework. Whereas in the experiment group, dynamic and interactive animations based on predict-observe-explain were used as a presentation tool.

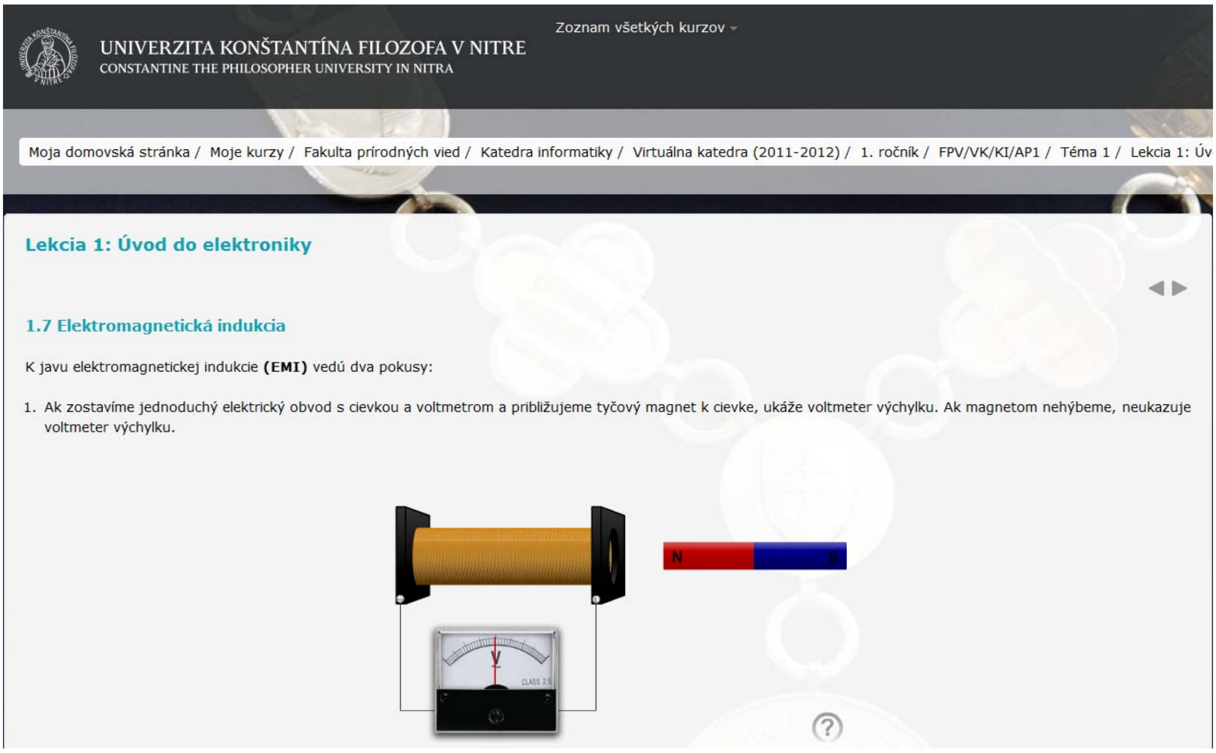

Figure 1: Example of interactive animation (changing the position of the magnet occurs misalignment of pointer the device).

Information on activities the student carried out in the e-learning course can be obtained from the records (Configuration module).
Observation of work can be divided into four main parts:

- Live signing from the last lesson,
- A report on activities,
- Records on participants,
- Statistics.

For the sake of acquisition of an idea of the real transition of all students through an e-learning course, methods of frequency and sequential analysis are mostly used. By means of these methods it is possible to set up the so-called interactive matrix (Chráska, 2007). Based on the found patterns of users´ behaviour, which are represented by sequence rules, it is possible to modify and improve the course (Munk et al, 2010). However, in order to be able to set up such type of matrix it is inevitable to filter out from the access statistics those data, which are connected with the side-show of students and they thus do not impact directly (or in a minimum possible degree) the method of acquisition of knowledge and skills. For us, such methods are for example (Nagyová, 2011):

- Initial course page view,
- Communication within the course,
- Profiles scanning.

RESULTS OF RESEARCH

Interactive matrix marked M represents a two-dimensional array of type n x n, where the number n is the number of overall activities realized by students in the course. It is possible to access the data in the matrix by means of the line number (variable i) and the column number (variable j). The matrix cells correspond to frequencies of incidence of variable j (of the given activity) after the activity i.

The creation of interactive matrix is influenced mainly by the selection of individual activities, which form header of the matrix. Activities depicted by the interactive matrix can represent, for example, the transition through individual chapters identically arrayed in line I and column j. In the following tables we present activities of students representing transitions between individual chapters of the study material in the e-learning course Architecture of computers in the academic years 2009/2010 (Winter semester) until 2013/2014 (Winter semester).

Table 1: Interactive matrix of transitions between individual lessons in the academic year 2009/2010 (control group).

	Start study	L1	L2	L3	L4	L5	L6	L7	L8	L9	End study
Start study	0	2450	852	356	124	258	689	346	734	428	45
L1	892	0	1987	556	87	219	324	222	318	110	23
L2	634	554	0	2041	918	796	369	567	216	257	51
L3	176	652	347	0	1321	821	221	705	599	375	74
L4	841	869	490	1458	0	1878	478	756	311	338	36
L5	654	512	591	428	998	0	2887	568	850	151	111
L6	317	974	627	898	370	350	0	1655	347	185	34
L7	268	498	623	495	580	915	1331	0	1201	100	174
L8	954	825	829	461	613	558	471	434	0	1637	190
L9	438	604	268	947	864	466	420	623	350	0	1255
End study	526	249	315	185	277	216	265	170	57	46	0

Table 2: Interactive matrix of transitions between individual lessons in the academic year 2009/2010 (experimental group).

	Start study	L1	L2	L3	L4	L5	L6	L7	L8	L9	End study
Start study	0	1629	360	847	356	785	147	324	897	489	121
L1	980	0	1322	123	784	324	472	246	146	732	13
L2	999	2401	0	1325	258	753	159	456	321	798	27
L3	589	125	1324	0	2471	125	245	587	523	348	11
L4	359	547	125	756	0	687	225	586	152	245	58
L5	458	365	852	456	1247	0	122	111	252	584	98
L6	221	456	247	247	328	122	0	122	212	523	102
L7	556	128	258	258	654	122	125	0	578	236	54
L8	768	745	265	136	369	356	578	1125	0	1456	24
L9	452	257	132	458	147	369	785	145	1184	0	1471
End study	321	253	457	563	235	236	227	123	115	111	0

Table 3: Interactive matrix of transitions between individual lessons in the academic year 2010/2011 (control group).

	Start study	L1	L2	L3	L4	L5	L6	L7	L8	L9	End study
Start study	0	1005	123	241	235	632	568	512	122	145	11
L1	23	0	1247	220	215	666	553	596	215	233	123
L2	57	1147	0	2343	512	213	621	232	232	515	12
L3	123	111	1238	0	1247	233	258	223	562	923	45
L4	357	254	233	266	0	3568	222	465	212	232	95
L5	159	475	253	156	556	0	1247	213	113	952	78
L6	654	44	452	696	321	265	0	2582	875	213	68
L7	789	458	563	668	545	546	1024	0	2562	565	65
L8	257	754	126	160	456	546	546	555	0	1220	23
L9	369	351	165	161	516	815	566	546	872	0	1235
End study	57	123	124	245	264	214	235	11	63	24	0

Table 4: Interactive matrix of transitions between individual lessons in the academic year 2010/2011 (experimental group).

	Start study	L1	L2	L3	L4	L5	L6	L7	L8	L9	End study
Start study	0	1609	720	662	792	558	822	521	880	830	43
L1	955	0	1595	435	849	874	729	912	420	218	62
L2	949	1674	0	1455	896	411	862	325	930	538	30
L3	958	449	1674	0	1355	297	808	375	358	994	42
L4	221	706	667	1721	0	1279	831	480	814	266	53
L5	551	656	742	505	378	0	1004	458	609	292	20
L6	156	795	302	804	928	1429	0	1108	351	203	175

L7	663	694	251	846	956	892	676	0	1184	161	42
L8	982	356	826	703	629	710	615	1123	0	1523	262
L9	519	546	334	590	495	554	863	294	187	0	1358
End study	259	177	142	236	112	190	127	90	134	217	0

Table 5: Interactive matrix of transitions between individual lessons in the academic year 2011/2012 (control group).

	Start study	**L1**	**L2**	**L3**	**L4**	**L5**	**L6**	**L7**	**L8**	**L9**	**End study**
Start study	0	2687	235	214	268	235	789	124	125	247	13
L1	74	0	1987	234	157	547	249	652	735	621	56
L2	147	2410	0	2343	457	652	234	475	578	256	14
L3	478	245	235	0	1247	145	125	221	154	152	16
L4	245	592	265	263	0	3568	262	215	262	110	24
L5	212	812	256	124	454	0	1247	512	155	823	48
L6	142	323	262	296	265	265	0	2582	652	854	78
L7	25	548	546	556	263	215	256	0	2562	158	98
L8	57	485	152	152	125	287	152	158	0	1220	91
L9	21	156	145	458	542	225	486	267	225	0	2347
End study	47	4	15	215	25	50	54	52	87	12	0

Table 6: Interactive matrix of transitions between individual lessons in the academic year 2011/2012 (experimental group).

	Start study	**L1**	**L2**	**L3**	**L4**	**L5**	**L6**	**L7**	**L8**	**L9**	**End study**
Start study	0	1578	258	152	785	225	211	125	125	215	65
L1	125	0	2145	15	158	524	223	215	152	185	23
L2	325	1360	0	2254	254	258	258	872	155	375	25
L3	252	154	2451	0	1524	151	284	582	152	815	15
L4	152	522	145	1545	0	1552	216	415	562	812	12
L5	58	528	485	458	3542	0	5222	132	521	217	244
L6	158	785	212	556	569	511	0	2725	215	25	552
L7	548	542	252	541	965	232	51	0	1247	223	54
L8	215	85	895	961	215	548	514	4325	0	2251	85
L9	85	57	12	23	514	584	145	12	2152	0	3332
End study	485	595	95	558	45	54	82	48	48	21	0

Table 7: Interactive matrix of transitions between individual lessons in the academic year 2012/2013 (control group).

	Start study	**L1**	**L2**	**L3**	**L4**	**L5**	**L6**	**L7**	**L8**	**L9**	**End study**
Start study	0	3564	52	562	596	592	114	596	325	147	45
L1	256	0	1045	51	155	214	124	154	128	875	5
L2	784	256	0	1384	135	357	158	152	258	761	26
L3	136	863	4578	0	2004	201	741	357	258	208	45
L4	789	853	121	182	0	3007	257	722	225	167	56
L5	568	259	158	637	475	0	1367	277	248	365	45
L6	14	972	223	951	430	457	0	1473	256	152	15
L7	26	782	182	892	942	211	247	0	4236	255	54
L8	189	784	253	261	885	555	445	115	0	2544	25
L9	288	123	127	357	226	168	496	957	5687	0	2347
End study	12	256	213	686	145	556	562	215	25	4368	0

Table 8: Interactive matrix of transitions between individual lessons in the academic year 2012/2013 (experimental group).

	Start study	**L1**	**L2**	**L3**	**L4**	**L5**	**L6**	**L7**	**L8**	**L9**	**End study**
Start study	0	1032	152	56	545	589	325	591	102	147	22
L1	874	0	2567	256	262	258	894	255	235	482	56
L2	852	2223	0	1078	251	487	365	811	254	278	21
L3	472	152	157	0	1125	963	578	215	754	221	32
L4	465	254	271	1472	0	5687	145	878	241	811	247
L5	863	505	562	255	1254	0	2354	922	152	158	82

L6	552	225	522	364	125	3587	0	2235	751	235	12
L7	124	526	821	482	142	122	6211	0	4210	257	54
L8	989	222	121	212	224	127	752	1235	0	1247	25
L9	222	121	474	125	758	223	352	121	2225	0	1985
End study	22	25	48	25	58	52	23	48	16	21	0

Table 9: Interactive matrix of transitions between individual lessons in the academic year 2013/2014 (control group).

	Start study	L1	L2	L3	L4	L5	L6	L7	L8	L9	End study
Start study	0	1027	258	245	215	122	171	825	150	147	15
L1	452	0	2522	148	847	512	152	821	122	215	82
L2	256	1270	0	7511	253	895	122	148	152	472	47
L3	548	472	4851	0	2233	223	872	335	522	417	42
L4	582	415	845	581	0	3231	485	512	145	213	23
L5	852	481	485	527	535	0	1233	222	852	354	51
L6	562	212	485	215	415	147	0	1845	152	212	32
L7	851	151	562	561	185	758	669	0	4154	512	21
L8	75	123	213	357	152	152	154	585	0	1522	15
L9	125	145	21	478	84	15	128	482	1223	0	4844
End study	25	58	15	21	72	15	54	14	54	4122	0

Table 10: Interactive matrix of transitions between individual lessons in the academic year 2013/2014 (experimental group).

	Start study	L1	L2	L3	L4	L5	L6	L7	L8	L9	End study
Start study	0	1984	245	275	365	956	214	482	147	120	32
L1	84	0	5458	523	182	482	145	124	142	754	12
L2	145	1223	0	4235	458	125	215	582	287	122	21
L3	752	123	255	0	4203	523	581	152	851	522	35
L4	635	154	452	553	0	2536	152	845	485	235	89
L5	545	152	472	851	1234	0	2347	582	215	215	12
L6	375	512	152	152	123	152	0	1208	264	823	45
L7	754	123	158	254	502	123	5145	0	1369	556	52
L8	421	215	852	125	258	921	122	1522	0	2049	59
L9	215	145	495	528	555	145	552	142	3547	0	2102
End study	69	72	125	57	82	15	56	15	82	75	0

Based on interactive matrices we can observe the frequency of incidence of j- sequentiality (activity) after the activity depicted in line i. Values, which are highlighted in colours, represent maximum values within the line and column, and at the same time, in both interactive matrices are highlighted those values, which significantly influence the transition of one activity into another (all numbers in cells above the value 1000). Maximum value in the column expresses the fact that students realized the given activity most frequently and then they proceeded in another activity with the highest maximum value situated in the nearest column. In case that in the column of the interactive matrix appears more than one maximum value, it means that student during his study returned to this activity after a while. This phenomenon can be observed in both interactive matrices (each academic year).

DISCUSSION

By modelling the behaviour of users based on their activity we succeeded in defining the real transition through the e-learning course using interactive matrices. Experiment, which was carried out in academic years 2009/2010 through 2013/2014, was focusing on defining the effectiveness of utilization of interactive animations in the e-learning course. The method we used can be considered an indirect one. Based on the results of interactive matrices in individual academic years it is clear that the experimental group, which kept the e-learning course with implemented interactive animations at their disposal for the whole period of study, kept returning to the previous study lessons at any time. This fact proves our presumption that by implementing interactive animations into the e-learning course its didactic effectiveness as well as the one of its utilization were increased. That is the following rule of proportion applies: the more frequently the students used the back transition, the more frequently they employed interactive animations and the e-learning course itself. High figures ranging between 3000 and 5000 presented in interactive matrices represent places to which the students returned based on the written test announced in advance and also the places, to which the students returned after completing the test in order to verify the correctness of what they had written into the test. It is interesting that the students in the reference group, who did not use the implemented interactive animations, employed the previous lessons only in a minimum way, despite the announced written tests. When using a questionnaire, as a research and evaluation pedagogic tool for finding information on the knowledge, opinions or attitudes of students we found that students of the reference group were frequently frustrated and irresponsible and took the study slovenly. Since the students, who were divided into both the experimental and reference groups were offered commonly and in the same form realized lectures, the only reason for their failure can be seen just in the distinct provision of the study material.

CONCLUSIONS

The results show evidence that interactive simulation contents can be very effective tools in the learning process. It can deliver information in a very attractive way, which also can be advantageous in assembling curricula for the students who have different skill levels and learning styles. Besides that, it can help learners to understand scientific topics, with presenting important conceptual relationships (Pinter et al., 2012).

In case of implementing interactive animations into the study material we obtain not only attractive form of providing the knowledge to the students, but also the possibility to determine the way the students use to work with this material. However, in the contribution we pointed also to another fact in case of using interactive animations, which is returning to previous lessons. This step is very important within the educational process, since by means of it there comes to the confirmation and stabilization of the contents of lessons. In case of the experimental group, which employed the implemented study materials, there came to the reverse transition and thus there is a presumption that the students attempted at putting the concepts acquired by means of interactive animations into context with the concepts previously taken within the study material. In the contribution we focused only on determining the activity of the students when using interactive animations and modelling the transition through the e-learning course. Study results of both the reference and experimental groups were not evaluated in this contribution. However, based on the partial evaluation we can state that differences between the experimental and the reference groups were marginal.

Similarly, as we do, Pinter came to the conclusion: However, results also show that there is a tendency of decreasing the difference between those learners who had used the animation and those who had not. Is this because there is an increasing number of such and similar e-curricula available to students, and this kind of attractive multimedia presentations are no longer motivate students as they used to before. However according to the Felder–Silverman (Richar and Rebeca, 2005) learning style model, the animations containing a lot of visual elements, such as pictures, diagrams, flow charts etc. are preferred for the visual learning profile, while written and auditory explanations are effective with the verbal type of student. And to mention another example: students with an active profile prefer the simulation (interactive animation) which allows experimenting with the system parameters. (Pinter et al., 2012).

REFERENCES

Akpinar, E. (2013). The use of interactive computer animations based on POE as a presentation tool in primary science teaching. *Journal of Science Education and Technology*, 1-11.

Azar, A. (2001) Analysis of university students' misconceptions in electricity subject [Universite ogrencilerinin elektrik konusunda kavram yanılgılarının analizi]. Yeni Binyılın Basında Turkiye'de Fen Bilimleri Eg˘itimi Sempozyumu Bildiri Kitabı, pp 345–350. Istanbul

Baser, M., Geban, O. (2007) Effect of instruction based on conceptual change activities on students' understanding of static electricity concepts. *Res Sci Technolo Educ* 25(2):243–267

Chambers, SK., Andre, T. (1997) Gender, prior knowledge, interest, and experience in electricity and conceptual change text manipulation in learning about direct current. *J Res Sci Teach* 34(2): 107–123

Chiu, ML., Chiu, MH., Ho, CY. (2002) Using cognitive-based representations to diagnose students' conceptions of the characteristics of matter. *Proc Natl Sci Counc* ROC (D) 12(3):91–99

Chráska, M. (2007) *Metody pedagogického výzkumu*, Praha: Grada Publishing.

Guruswamy, C., Somars, MD., Hussey, RG. (1997) Students' understanding of the transfer of charge between conductors. *Phys Educ* 32:91–96

Lee, Y., Law, N. (2001) Explorations in prompting conceptual change in electrical concepts via ontological category shift. *Int J Sci Educ* 23(2):111–149

Mayer, R.E., Gallini, J.K. (1990). *When is an illustration worth ten thousand words?*, Journal of Educational Psychology, 1990, Vol. 82, No. 4,715-726

Mayer, RE. (1989). *Systematic Thinking Fostered by Illustrations in Scientific Text*, Journal of Educational Psychology, 1989, Vol. 81, NTo. 2, 240-246

Munk, M., Kapusta, J., Švec, P and Turčáni, M. (2010) 'Data advance preparation factors affecting results of sequence rule analysis in web log mining', *E a M: Ekonomie a Management*, vol. 13, no. 4, pp. 143-160.

Nagyová, I. (2011) 'Modelování výukových procesů Petriho sítěmi', *Metódy modelovania a analýzy dát v informačných systémoch*, Nitra, pp. 49-65.

Pinter, R., Radosav, D., Cisar, S.M. (2012). Analyzing the Impact of Using Interactive Animations in Teaching. *Int. J. of Computers, Communications & Control*, Vol. VII (2012), No. 1 (March), pp. 147-162

Richar F., Rebeca, B. (2005.)*Understanding Student Differences*, Journal of Engineering Education, 94 (1), 57-72

Sekular, R., and Blake, R. (1990). *Perception, Second Ed.*, New York, McGraw-Hill, 1990

Sharon, KC., Thomas, A. (1997). Gender, prior knowledge, interest and experience in electricity and conceptual change text manipulations in learning about direct current. *J Res Sci Teach* 34(2):107–123

Tsai, C-C. (2003). Using conflict map as an instructional tool to change student alternative conceptions in simple series electric-circuits. *Int J Sci Educ* 25(3):307–327