Development of A Learning Model on Software Design Pattern Selection for Novice Developers

Masita Abdul Jalil
Faculty of Ocean Engineering Technology and Informatics, Universiti Malaysia Terengganu
21030 Kuala Terengganu, Terengganu, Malaysia
+6096683274
masita@umt.edu.my

Nurul Azarina Abd.Rahman
Faculty of Ocean Engineering Technology and Informatics, Universiti Malaysia Terengganu
21030 Kuala Terengganu, Terengganu, Malaysia
+6096683274
mszaryna@gmail.com

Noraida Hj. Ali
Faculty of Ocean Engineering Technology and Informatics, Universiti Malaysia Terengganu
21030 Kuala Terengganu, Terengganu, Malaysia
+6096683260
aida@umt.edu.my

Shahrul Azman Mohd Noah
Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia
43600 Bangi, Selangor, Malaysia
+ 60389216812/6343
shahrul@ukm.edu.my

Noor Maizura Mohamad Noor
Faculty of Ocean Engineering Technology and Informatics, Universiti Malaysia Terengganu
21030 Kuala Terengganu, Terengganu, Malaysia
+6096683355
maizura@umt.edu.my

Fatihah Mohd
Faculty of Entrepreneurship and Business, Universiti Malaysia Kelantan, City Campus,
16100 Pangkalan Chepa, Kelantan, Malaysia
+6097717000
mpfatihah@gmail.com

ABSTRACT
Design pattern is still actively discussed in software engineering academic research. The highlight of research that still gains attention is the use of design patterns as a learning medium to improve object-oriented design skills among fresh developers or novices. This paper focuses on the design of a learning model for design pattern selection. The main objective of the proposed model is to reduce the learning curve on design pattern application. Selected cognitive methods are implemented to minimize the cognitive complexity throughout the pattern selection process. This is aimed to assist novices in learning the process of matching the design problem to design pattern. This learning model will be utilized as a practice for novices to gain expert design skills from diverse design approaches through design patterns. The learning model simplifies the pattern selection process which comprises of three sub-processes; 1) Identify design strategy 2) Identify design scope and 3) Identify design intention. In each sub-process, potential words indicating design flaws are highlighted to guide novices in identifying the underlying design issues in the attended problem. The keywords highlight feature enables novices to highlight correct information within the design problem that leads to the identification of the right solution from design patterns.

ICEIT 2020, February 11–13, 2020, Oxford, United Kingdom

ACM ISBN 978-1-4503-7508-5/20/02…$15.00
https://doi.org/10.1145/3383923.3383966

CCS Concepts
• Applied Computing→Education→Computer-assisted instruction

Keywords
Cognitive theory; design pattern; learning model; novice; design pattern selection.

1. INTRODUCTION
Design pattern has been incorporated into software engineering education as an exposure to the real practice of software analysis and design. Other than teaching design pattern through collaborative learning [3,10,12,24], some studies focus on developing training tool that students can utilize to gain object-oriented design skills through design patterns.

It is widely accepted that design pattern has great potential in enhancing design skills for fresh developers or novices [1,4,6,7,20]. However, the ability to understand the patterns is limited by novices' experiences. Design pattern manifests high level object-oriented design skills. To be able to use them, one has to be familiar with complex design cases that are commonly encountered in large scale systems. Without past experiences dealing with complex design, novice could not justify the solution presented in design patterns thus unable to utilize them.

In design pattern application, the complexity is mainly attributed to the abstract feature of design patterns and inability to identify the underlying design issues. Many studies address the complexity of design patterns [8], but no study is conducted to address complexity in understanding the underlying issues in the problem domain. As the complexity of pattern application is contributed from both domains, intervention on both design pattern and design problem context is necessary.

In this study, we develop a learning model for design pattern application to guide novices to develop skills in applying design patterns, by minimizing cognitive complexity on both solution domain and problem domain. A web based system is developed to demonstrate the proposed learning model. With the proposed model, novices are expected to gain higher level design skills through practicing design patterns.

Learning model is designed specifically for the application of Gang of Four design patterns by Gamma *et al.* [5]. Cognitive methods applied in the model aimed at minimizing the complexity in the design pattern selection process through instructional design of the learning materials [9,13,14,15,18].

2. METHOD AND SYSTEM DESIGN

2.1 Stepwise Pattern Selection Process

Design pattern selection process is simplified to match with novices' level. Cognitive methods are applied in both parts of pattern selection process which are problem domain and solution domain. The instructional design o0f the selection process is applied in a problem-based learning environment to allow novices construct the base knowledge on how to apply design patterns [25].

To address complexity on solution domain, pattern selection process is split into three sequences of sub-process to allow phased understanding as shown in Figure 1. Each sub-process is guided with sub-goal which needs to be completed to proceed to the next sub-process. The sub-goal as shown in Figure 2 represents the objective of each sub-process which is structured using a top-down approach. Upon completion of each successive sub-process, the anticipated solution is more directed towards specific design patterns. With Sequencing Method [22, 23] and Sub-goal Method [13] applied, the complexity of the design pattern knowledge is reduced, enabling novices to complete the process and construct knowledge base, phase by phase.

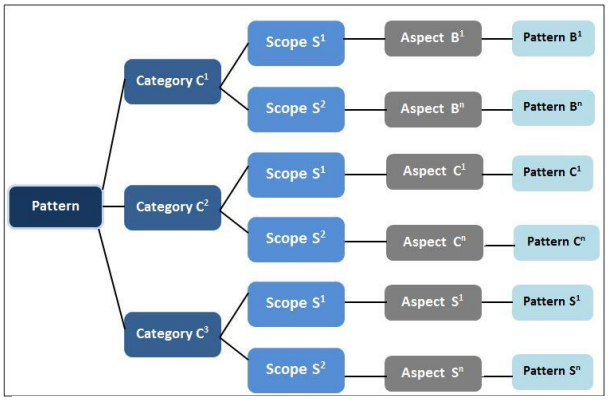

Figure 1. Simplified Design Pattern Selection

The sequence of design pattern selection process is structured as below:

i) Phase 1 - Identify Design Strategy

First selection phase sets to identify how the problem can be solved; by manipulating object's communication, object's creation, or object's composition.

The solution lists represent GoF pattern category; Behavioral, Creational and Structural. Behavioral patterns primarily deal with communication between class objects. This category describes the way objects and classes interact to distribute responsibilities. It manifests ways to change the object's behavior while retaining the same interface for the class. Creational patterns abstractly define the class instantiation. This type of pattern deals with such design situation through the creation of class objects. In solving design issue, Creational patterns propose the right mechanism to instantiate an object fitting to the design situation, of what, when and how the objects should be created. Structural patterns mainly use composition to combine classes and objects into a larger structure. This type embraces design changes through combining or adapting different objects together. At this phase, novices only need to focus on identifying a general design strategy to solve the design problem.

ii) Phase 2 - Identify Design Scope

The second phase further specifies the pattern's list. This phase sorts the pattern's list by its design scope, the dynamic behavior of the design that identifies how the objects execute the changes. Object-scope allows multiple objects to run simultaneously, thus the task can be changed dynamically at run-time. Class-scope uses inheritance to delegate the tasks, creating relationship between the class and its sub-classes. At this phase, the selection of patterns is narrowed from the pattern category in the first phase.

iii) Phase 3 - Identify Design Intention

The third or final phase explicitly addresses the changing object behavior in a pattern. From previous phase, this phase lists out pattern's variation within the selected category and scope. At this phase, novices will be able to match the pattern that is most identical to the addressed issue.

2.2 Keywords Highlight on Design Problem

Keywords highlight feature is added throughout pattern selection process to minimize complexity on problem domain. The main objective is to guide novices in identifying the underlying design issue that needs to be addressed. Selected keywords as shown in Figure 2 are highlighted to understand design flaws in the design problem.

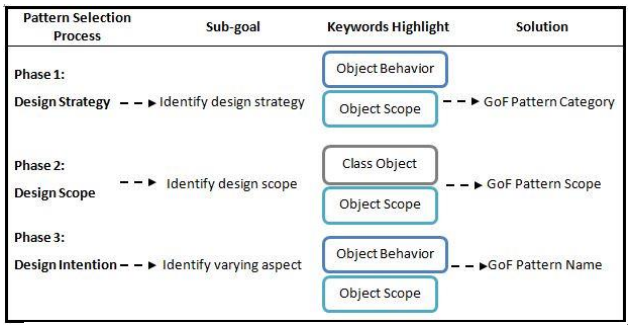

Figure 2. Cognitive Based Design Pattern Selection Process

Keywords extraction process involves multiple text processing algorithms which are parsing, tagging and tokenizing to obtain the list of keywords from the text. The objective is to enable keywords highlighting on the problem text. The text is first analyzed by the parsing algorithm to structure the sentence into sub-phrases. On each selection phase, selected type of keywords are extracted and highlighted based on the objective manifested in the sub-goal as shown in Figure 2. Keyword extraction

feature implements Signaling method to reduce mental load invested on unnecessary processing during learning [14].

The first selection phase aims to guide novices in identifying a strategy to redesign the system. An example workflow of the first selection phase is shown in Figure 3 using Interactive Quiz Environment task. First phase demands a thorough understanding on the entire design structure in order to see the changing part of it. Novices must be able to differentiate the types of participating objects and the relationships between the objects. Verb phrase could reflect the task or behavior of the class objects, while Adverb phrase describes how the task is performed. Highlighting verb and adverb phrases in the problem text would guide novices to identify the object behavior and its scope thus map the changing behavior with the most suitable strategy for the solution.

In the second selection phase, novices need to identify the scope of the system design. This phase solely focuses on determining how the class objects execute the changing behavior. It is important to identify design scope in order to choose a pattern that would properly address the changes. In a problem text, noun phrases represent the class objects, while adverb phrases might signal how the participating class objects execute the behavior. These highlighted phrases would help novices to see if the objects need to run dynamically or statically, thus lead to identifying the design scope.

The final selection phase specifically addresses the changing elements of the system that should be redesigned. It can be identified by analyzing the affected class objects and how the objects perform the task. From the previously selected design's strategy and scope, we highlight verb and adverb phrases that represent object behavior and object scope to aid novices identify the varying aspect of the class objects and how they should be restructured. At this final phase, novices would be able to select a single most suitable pattern to solve the attended problem.

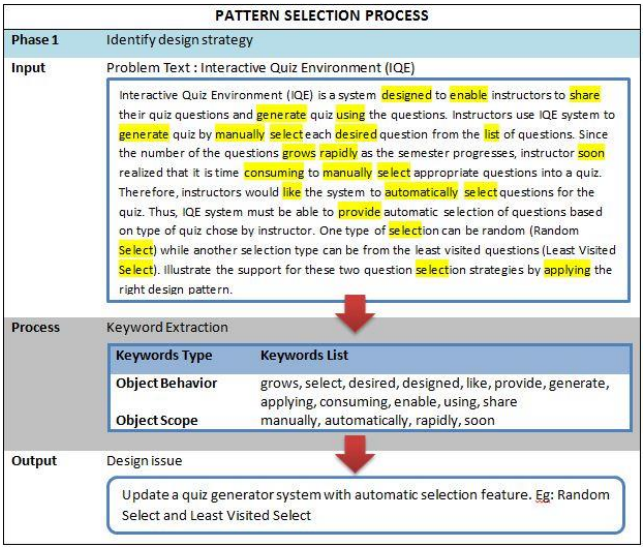

Figure 3. Pattern Selection Process (Phase 1)

2.3 Cognitive Based Design Pattern Selection Tool

A web based system is developed to demonstrate the proposed learning model. The system is developed in Java incorporating MVC (Model View Controller) architecture and MySQL for database. OpenNLP API is used for text processing in keyword extraction algorithm. The conceptual view of the cognitive based learning model for design pattern selection process addressing both problem domain and solution domain is illustrated in Figure 4.

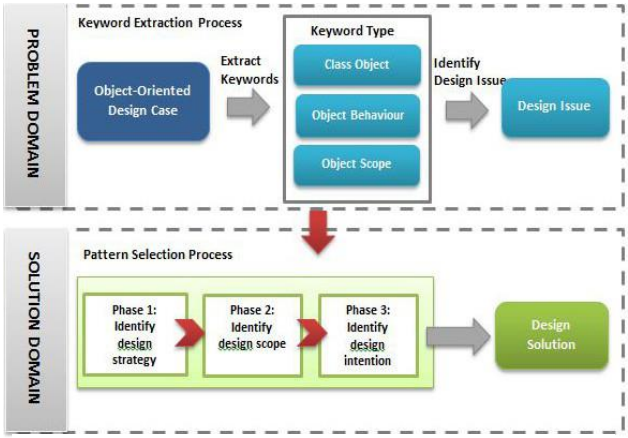

Figure 4. Cognitive Based Learning Model for Design Pattern Selection

2.4 Evaluation

An experiment session is designed to assess the effect of cognitive based design pattern selection model implemented in the web based prototype tool. The result from the evaluation is inferred to the population of novice developers. The effectiveness of the prototype is evaluated based on two aspects:

a) How far the proposed design pattern selection model could assist novice in selecting suitable design pattern to the attended design problem

b) How far the proposed design pattern selection model could assist novice in acquiring design pattern selection skills.

2.4.1 Experiment Design

The experiment is set up to measure the effect of cognitive methods proposed through instructional design of learning materials in assisting novices acquiring pattern selection skills. Thus, the pre and post-test study is chosen to measure the difference in performance score before and after an interaction session with the prototype tool. The experiment is structured in four sessions; i) Introductory lecture, ii) Pre-test, iii) Training (Interaction with web based tool) and iv) Post-test.

Introductory session highlights the importance of design patterns as an effective tool in developing better quality software. Then, the principles of object-oriented design implemented in the patterns are described. Case studies are presented to expose novices with real application of design patterns. To evaluate novices' understanding on pattern selection's strategy, the design tasks provided in the pre-test and post-test are delivered without cognitive manipulation on the learning materials. The tasks provided during training session are cognitively manipulated to encourage skills acquisition on pattern selection. At post stage of training, novices are expected to acquire the skill in selecting pattern.

For this experiment, two object-oriented design tasks are chosen from past study [16]. Based on the principles outlined by [16] in creating the design task, the task for this experiment is selected based on the following reasons:

i). the task reflects to the real application of design pattern in a real software design issue

ii). the task is created within a domain knowledge in which novices are familiar with

iii). the task presents the issue within a medium size application that is feasible to novices

The experiment was conducted covering twelve out of 23 design patterns in the GoF catalogue. The patterns are selected based on suitability score for beginner level and also usability factor in real software development. The lists of selected patterns are Factory, Abstract Factory, Prototype, Singleton, Template, Observer, State, Strategy, Adapter, Bridge, Decorator and Facade.

2.4.2 Subject

The subjects selected as sample size represents novice developers with no past experiences in real software development. The other criteria assessed on selecting the sample size is proficiency level in object-oriented design in which object-oriented programming and design courses are set as prerequisite courses taken in previous semester. The sample size of 20 students is selected from undergraduate students of Computer Science in Universiti Malaysia Terengganu, Malaysia.

2.4.3 Data Collection

Two methods are used to collect data which are questionnaire and the performance score on pattern selection. The questionnaire is completed prior to the experiment session to gauge subjects' demographic information. Selection score is recorded on each completed task in the pre and post-test respectively.

To measure the significance of the variable measured in this experiment, the data collected will be analyzed using hypothesis testing. Null hypothesis (H0) is set as a base to test validity of H1, for selection score as below:

H0: Cognitive based design pattern selection learning model does not give significant difference in mean selection score for the pre-test and post-test.

H1: Cognitive based design pattern selection learning model gives significant difference in mean selection score for the pre-test and post-test.

To gain feedback on subjects' learning experience with the support tool, mental effort rating is assessed to reflect the complexity experienced by each subject during the training session. Mental effort is assessed using self-reporting report. After completion of each task, subjects are asked to rate their perceived difficulty. The rating scale is adapted from self-reporting method using 7-point Likert scale to assess the cognitive load [22]. A question to assess mental effort is "How difficult you find to complete the task?" with scale from 1-"Very Easy" to 7-"Very Difficult". The actual level of cognitive load occupied on mental load can be directly assessed with mental effort [17]. Mental effort rating is combined with performance score to measure cognitive efficiency as an indicator to the effectiveness of proposed model in the prototype tool. Additionally, for each task attempted in the support tool, performance score is assessed on each selection phase respectively. This is done to gain additional insight on the difficulty of each selection phase.

3. RESULT

Shapiro-Wilk is used to conduct normality test on the difference of selection score between the pre-test and post-test session. The significant value on pre-test score is less than the significant level of P (0.05) which indicates the data is not normally distributed. The score in post-test is constant, thus normality test is omitted. Hence, a non-parametric statistical method, the Wilcoxon Signed Ranked, is used to test the sample data.

The result in Table 1 summarizes the mean difference of selection score between the pre-test and post-test. Based on the table, three subjects achieved higher score in the pre-test, while the other 17 subjects show no difference between the pre-test and post-test.

Table 1. Mean Selection Score

			Ranks	
		N	**Mean Rank**	**Sum of Ranks**
Posttest score-Pretest score	Negative Ranks	3[a]	2.00	6.00
	Positive Ranks	0[b]	.00	.00
	Ties	17[c]		
	Total	20		

a. Posttest score < Pretest score
b. Posttest score > Pretest score
c. Posttest score = Pretest score

From statistical test, the calculated P value is 0.083, which is greater than the significant value 0.05 as recorded in Table 2. Thus, null hypothesis (H0) is accepted. With the significant value of 0.05, it can be concluded that the proposed pattern selection model does not give significant difference in mean selection score. With limited evidence collected to support the hypothesis indicating effectiveness of the proposed learning model in enhancing design pattern selection skill, the distribution of selected patterns made by subjects however show positive inclination towards the right pattern. This is based on the justification made on the distribution of selected pattern category. Thus, although the selection is imprecise, it can be inferred that subjects are able to recognize the suitable design strategy for the attended design problem.

Table 2. Statistical Test on Pattern Selection Score

	Test Statistics[a]
	Posttest score – Pretest score
Z	-1.732[b]
Asymp. Sig. (2-tailed)	.083

a. Wilcoxon Signed Ranks Test
b. Based on positive ranks

4. SUMMARY AND CONCLUSION

In this paper, we describe the design of a learning model for design pattern selection. Our main objective is to facilitate novices in understanding and gaining design skills through design patterns. We highlight two main complexity faced by novice in applying design patterns that are contributed from the pattern documentation as solution domain and the design

problem as problem domain. As previous studies only address complexity on pattern documentation (solution domain), this study focuses on reducing complexity on the problem domain as well. Then, we design a learning model for design pattern application focusing on minimizing complexity on both domains. We apply cognitive theory in the proposed learning model for two purposes:

i) To reduce complexity in problem-pattern mapping by addressing the complexity from the problem context represented as problem domain and also the pattern as solution domain

ii) To facilitate analysis on problem description in problem domain in order to address the underlying design issue that channels towards identification of the right pattern solution.

Sequencing, Sub-goals and Signaling are applied on the proposed learning model. Sequencing Method and Sub-goals method are applied in the design of the stepwise pattern selection process, while Signaling method is applied on the keyword highlight feature. A web based prototype is developed to demonstrate the proposed learning model.

Experiment session is administered to evaluate the effectiveness of the proposed design pattern selection learning model in assisting novices gaining skills in pattern selection. The pre and post study is applied to measure differences in mean selection score for the design tasks. The experiment is designed considering possible threats to the validity of the collected data. In discussing about the degree in which the result validly reflects the population, a small threat might be imposed from the selection of GoF patterns used in the evaluation session. From 12 patterns used in this experiment, the result somehow could not be generalized to the entire 23 GoF patterns.

Analysis is conducted on performance score collected in the pre and post-test session. The result from the statistical test showed limitation of the proposed model in enhancing understanding on design pattern selection for novices. However, the result on the first selection phase shows positive inclination towards identifying the right pattern. Hence, we could summarize that while the selected patterns are not precise, it can be inferred that through practices with the proposed learning model, novices are able to recognize the suitable design strategy to solve the attended design problem.

5. ACKNOWLEDGEMENTS

We wish to thank the Ministry of Education (MOE) Malaysia for providing financial support for this study under the Research Acculturation Collaborative Effort (RACE) Grant Scheme.

6. REFERENCES

[1] Ahmed, S., & Wallace, K. M. 2003. Understanding the differences between how novice and experienced designers approach design tasks. Research in engineering design. 2003 Feb 1;14(1):1-1.DOI= http://doi.org/10.1007/s00163-002-0023-z

[2] Berdun, L., Amandi, A., & Campo, M. 2014. An intelligent tutor for teaching software design patterns. *Computer Applications in Engineering Education*, *22*(4), 583–592. DOI=http://doi.org/10.1002/cae.20582

[3] Claire, E., & Wick, M. R. 2005. Teaching Design Patterns in CS1 : A Closed Laboratory Sequence based on the Game of Life, 487–491.

[4] Denzler, C., & Gruntz, D. 2008. Design patterns: Between programming and software design. *In Proceedings of the International Conference on ACM/IEEE 30th Software Engineering, 2008. ICSE '08.* pp. 801–804.

[5] Gamma, E. 1995. *Design patterns: Elements of Reusable Object-Oriented Software*. Pearson Education India.

[6] Ghazarian, A. 2012. Work in progress: Transitioning from novice to expert software engineers through design patterns: Is it really working? In *Proceedings - Frontiers in Education Conference, FIE*. DOI=http://doi.org/10.1109/FIE.2012.6462302

[7] Ghazarian, A. 2015. A theory of software complexity. *Proceedings - 4th SEMAT Workshop on General Theory of Software Engineering, GTSE 2015*, 29–32. DOI=http://doi.org/10.1109/GTSE.2015.11

[8] Jalil, M. and Noah, S. A. 2007. The Difficulties of Using Design Patterns among Novices: An Exploratory Study. In Proceedings of the Fifth International Conference on Computational Science and Its Application, ICCSA'07 (Kuala Lumpur, Malaysia, August 26-29, 2007), 97-103. DOI=http://dx.doi.org/10.1109/ICCSA.2007.75

[9] Kalyuga, S. 2011. Cognitive Load Theory: How Many Types of Load Does It Really Need? *Educational Psychology Review*, *23*(1), 1–19. DOI=http://doi.org/10.1007/s10648-010-9150-7

[10] Kolfschoten, G., Lukosch, S., Verbraeck, A., Valentin, E., & Vreede, G.-J. de. 2010. Cognitive learning efficiency through the use of design patterns in teaching. *Computers & Education*, *54*(3), 652–660. http://doi.org/10.1016/j.compedu.2009.09.028

[11] Konecki, M., Orehovački, T., & Kermek, D. 2009. Design Patterns – education and classification challenge. *Proceedings of the 20th Central European Conference on Information and Intelligent Systems Aurer, Boris ; Bača, Miroslav ; Rabuzin, Kornelije - Varaždin : Faculty of Organization and Informatics (2009), 375-381*, 6.

[12] Köppe, C., & Pruijt, L. 2014. Improving Students' Learning in Software Engineering Education Through Multi-level Assignments. *Proceedings of the Computer Science Education Research Conference*, 57–62. DOI=http://doi.org/10.1145/2691352.2691357

[13] Margulieux, L. E., Catrambone, R., & Guzdial, M. 2016. Employing subgoals in computer programming education. *Computer Science Education*, *26*(1), 44–67. DOI=http://doi.org/10.1080/08993408.2016.1144429

[14] Mayer, R. E., & Moreno, R. 2003. Nine Ways to Reduce Cognitive Load in Multimedia Learning. *Educational Psychologist*, *38*(1), 43–52. DOI=http://doi.org/10.1207/S15326985EP3801_6

[15] Merriënboer, J. J. G. Van. 2003. Taking the Load Off a Learner's Mind: Instructional Design for Complex Learning, Educational psychologist, 38(1), 5-13. DOI=http://doi.org/10.1207/S15326985EP3801

[16] Ouyang, Y. 2002. Explaining Design Patterns Trough One Application, In Proceedings of the 32nd ASEE/IEE Frontiers in Education Conference (Boston, MA, November 06-09, 2002). IEEE Press, Washington, DC, 6-11. DOI=http://dx.doi.org/10.1109/FIE.2002.1158639

[17] Paas, F., Renkl, A., & Sweller, J. 2003. Cognitive Load Theory and Instructional Design: Recent Developments. Educational Psychologist, 38(1), 1–4. DOI=http://doi.org/10.1207/S15326985EP3801_1

[18] Pollock, E., Chandler, P., & Sweller, J. 2002. Assimilating complex information, *Learning and instruction, 12(1), 61-86.*

[19] Riaz, M., & Williams, L. 2012. On the Design of Empirical Studies to Evaluate Software Patterns : A Survey, North Carolina State University. Dept. of Computer Science (2012).

[20] Subburaj, R., Jekese, G., & Hwata, C. 2015. Impact of Object Oriented Design Patterns on Software Development, *6*(2), 961–967.

[21] Sweller, J. 1988. Cognitive load during problem solving: Effects on learning. *Cognitive Science*, *12*(2), 257–285. DOI=http://doi.org/10.1016/0364-0213(88)90023-7

[22] Sweller, J. 2008. Human Cognitive Architecture, 14.

[23] Sweller, J., Ayres, P., and S. K. 2011. *Cognitive Load Theory*. Springer New York. DOI=http://doi.org/10.1007/978-1-4419-8126-4

[24] Tao, Y., Liu, G., Mottok, J., Hackenberg, R., & Hagel, G. 2015. Just-in-Time-Teaching experience in a Software Design Pattern course. In *IEEE Global Engineering Education Conference, EDUCON* (Vol. 2015–April, pp. 915–919. DOI=http://doi.org/10.1109/EDUCON.2015.7096082

[25] Yan, J., & Lavigne, N. C. 2014. Promoting college students problem understanding using schema-emphasizing worked examples. *Journal of Experimental Education*, *82*(1), 74–102. DOI=http://doi.org/10.1080/00220973.2012.745466

[26] Zhang, C., & Budgen, D. 2012. What do we know about the effectiveness of software design patterns? *IEEE Transactions on Software Engineering*. DOI=http://doi.org/10.1109/TSE.2011.79