A Comparison of Three Documentation Styles for Educational Data Analysis

Sin-Ban Ho[1,1], Ian Chai[2], and Chuie-Hong Tan[3]

[1] Faculty of Computing and Informatics, Multimedia University, 63100 Cyberjaya, Malaysia
[2] Faculty of Engineering, Multimedia University, 63100 Cyberjaya, Malaysia
[3] Faculty of Management, Multimedia University, 63100 Cyberjaya, Malaysia
{sbho, ianchai, chtan}@mmu.edu.my

Abstract. Frameworks are increasingly employed as a useful way to enable object-oriented reuse. However, understanding frameworks is not easy due to their size and complexity. Previous work concentrated on different ways to document frameworks, but it was unclear which ones actually were better. This paper presents results in investigating the different philosophies for framework documentation. The philosophies include minimalist, patterns-style and extended javadoc (Jdoc) documentation. Using a survey of 90 intermediate undergraduates engaged in Command and Adaptor design patterns coding work, this exploratory study discovered that minimalist documentation has positive impact in encouraging knowledge acquisition, significantly in terms of the framework functional workings. This concludes that documentation solutions with the minimalist principle can lead intermediate undergraduates to faster growth in learning two of the design patterns.

Keywords: Educational Data Analysis, Learning Analytics, Knowledge Management, Empirical Research Result, Knowledge Surveying Work.

1 Introduction

One of the key challenges to object-oriented frameworks is introducing design patterns to intermediate undergraduates. Intermediate undergraduates are those who have already had some experience with the framework in question but are not yet experts, i.e. they are between the novice and advanced levels. This research work on online documentation adapts the philosophy of pair programming in agile development [1]. The subjects would perform the coding details of a particular portion of the code while the instructor ensures that the coding exercise is being followed with the help of time check-point in the documentation. The scope of this research project is to tackle intermediate undergraduate documentation or tutorials. This is a very important part because once past the beginner stage, one often has the familiarity to figure out the details.

[1] International Conference on Data Engineering (DaEng), p. 1-8, 2013.

T. Herawan et al. (eds.), *Proceedings of the First International Conference on Advanced Data and Information Engineering (DaEng-2013)*, Lecture Notes in Electrical Engineering 285, DOI: 10.1007/978-981-4585-18-7_79,

2 Motivation of the Study

Studies in pedagogical documentation show that the behavior in organizing a programming guide is a domain that has been used to describe the way beginners learn how to use a framework. For some time, studies have reported behavior differences in pedagogical framework documentation. The three philosophies being evaluated in this study include minimalist [2], patterns-style [3, 4] and extended javadoc documentation [5, 6]. Each is compatible with the idea of mixing texts, examples and diagrams.

Our main research question is to empirically test whether minimalist, patterns documentation or Jdoc presentation would give better performance in teaching intermediate undergraduates how to use design patterns. This question is indeed the main concern that is being challenged. In this paper, we use the Command and Adaptor design patterns [7] as the basis of study on the impact of the documentation philosophies. This study is a follow-up of the earlier study of beginners [8, 9]. In this paper, we intend to find the impact of intermediate undergraduates after they have undergone the beginning stage of programming within the Swing framework context.

3 Experiment Description

This research work used an exercise-based research typically used in empirical software engineering. One of the main components of the research methodology is exercise-based investigation, which was preceded with the presentation of a certain documentation set. The formulated hypotheses were used to design the documentation sets and the respective exercise, which were pre-tested for usability, soundness, and readability before it was rolled out for collecting data from the field. The data collected were then statistically analyzed using suitable data analysis techniques.

3.1 Documentation Procedure

The participants would follow the documentation and create java source code that imports the main Swing package i.e. javax.swing.* and two AWT (Abstract Windowing Toolkit) packages i.e. java.awt.event.* and java.awt.*. The expected result from these tasks is to have an outcome of running Command and Adaptor (CmdAdp) programs. An example of the CmdAdp documentation is shown in [10], which is organized into pieces to formulate the minimalist documentation.

The background information section is added to the top of each piece in order to form the patterns style [11]. For Jdoc, the background information is replaced by the output of the javadoc tool, which comprises of the extracted information from the source code about interfaces, methods and data-fields [12].

To provide a picture of the relative total length of the documentation, the documentation size is measured in kilobytes, as proposed by Beizer [13]. Through this approach, we can quantitatively characterize the documents. Table 1 gives quantitative information about the character of the documents used in this experiment.

Table 1. Characterize the relative documentation quantitatively.

Quantitative characterization	Minimalist	Patterns	Jdoc
1. Relative total length (in kilobytes, KB)	244 KB	293 KB	340 KB
2. Information that is relatively available	Short overview list of work tasks	Background information	Classes, method and interface information
3. Number of document files	10 files	13 files	22 files
4. Total sections	9 sections	14 sections	11 sections
5. Total paragraphs	17 paragraphs	27 paragraphs	24 paragraphs

3.2 Hypotheses

Standard significance testing is used to clearly specify the effects of the three documentation philosophies. The null hypotheses are stated as follows.

$E1H_0$ - There will be no difference between patterns and minimalist documentation for the intermediate undergraduates in doing the same exercise.

$E2H_0$ – There will be no difference between patterns and Jdoc documentation for the intermediate undergraduates in doing the same exercise.

$E3H_0$ – There will be no difference between minimalist and Jdoc documentation for the intermediate undergraduates in doing the same exercise.

The interpretations are derived from the rejection or non-rejection of these hypotheses for each expectation.

3.3 Experimental Design

Our experimental design uses one independent variable (factor) and five dependent variables. The independent variable consists of the documentation group. The dependent variables are the completion time, number of difficulties faced, semi completion time, workings and comprehension (understanding of the exercise).

Independent variable:

Documentation style: We use three documentation philosophies, as described in section 2, each with a similar purpose: to complete the given work task.

Dependent variables:

Semi Completion time: Time taken for the subjects to do their first compilation.

Completion time: The time taken to finish the entire exercise.

Comprehension: The subjects have to identify the method, procedure, line of the code, and constants that perform the given task. There are a number of questions to test their understanding of the code.

Workings: This is to test how well the subjects are able to follow the instructions for assigning default settings to the CmdAdp components.

Number of difficulties faced: Instead of giving all the detailed steps, some parts of the documentation allow the learners interact with the system. The subjects are to record and accumulate the number of problems they encounter.

3.4 Participants

There are 90 participants in this study. 33 (36.7%) are female and 57 (63.3%) are male, with the mean years in the university of 2.97, and SD of 0.436, a minimum 2 years and maximum 4 years in the university. Participants are all information technology undergraduates who undergo the object-oriented programming course at the university. The normal age of the students at this level is 22 years old.

To be able to test the hypotheses of our experiment, three different groups of the CmdAdp documentation are required. We arrange the participants into three different groups, according to their tutorial sections. Table 2 shows more detailed information about the groups.

Table 2. The detailed information and ANOVA tests results of years in the university (year) and previous achievement of C Language course (CLang), C++ (CPP), Data Structures and Algorithms (DataStruct) grades, and Cumulative Grade Point Average (CGPA).

Documentation style:	Minimalist	Patterns	Jdoc		F	p-value
N (participants)	26	26	38			
Mean (year)	3.08	2.92	2.92		1.175	0.314
Std. deviation (year)	0.077	0.110	0.058			
Mean (CLang)	3.16	2.95	3.16		1.015	0.367
Mean (CPP)	3.08	3.08	3.17		1.101	0.337
Mean (DataStruct)	3.04	1.58	1.74		7.843	0.001*
Mean (CGPA)	3.12	3.08	3.20		0.499	0.609

Note: * Statistically significant at 0.05 level

During the lectures, the students are taught basic object-oriented programming (OOP) principles. The lectures are supplemented by practical tutorial sessions where the students have the opportunity to make use of what they have learned through the completion of various java coding exercises using the assigned on-line documentation. Prior to this experiment, the preliminary stage of the on-line documentation presents the Swing framework [8]. The second stage discusses five of the design patterns [9]. This experiment focuses on the third stage of the intermediate undergraduates learning, which is on the CmdAdp design patterns. The participants in this experiment are regarded as intermediate undergraduates since they have attempted the prior two stages. They are not advanced users since they have not completed the OOP course yet.

3.5 Validity

To see whether the groups differ significantly, we perform ANOVA tests on the three groups of participants. In Table 2, with all the p-values > 0.05, except for the Data Structures and Algorithms course that they took in the prior semester, there is no major significant difference detected. The random assignments of the three tutorial groups are balanced in terms of their years in the university, the courses like C and C++ language, and Cumulative Grade Point Average (CGPA).

Furthermore, the total completion time of the participants shows an almost perfectly symmetric distribution. Thus, there is no evidence that slower participants hurried because of others having finished before them, in spite of the particular participant group working in the same laboratory at the same time. A final consideration is the precision and accuracy of time stamps recorded by the participants. Although the participants are informed that they have at most two hours to complete the work task, by cross checking, we discover that their responses in the time stamp to be highly reliable.

4 Data Analysis, Results and Discussion

Statistical analyses are conducted using Statistical Package for Social Science (SPSS). The results are based on the sample of 90 responses. The data is analyzed to see if one of the documentation sets let the participants compile (**Semi-Completion**) and finish the fastest (**Completion**) with the number of difficulties recorded by the subject at these intervals (**Number of difficulties**), as well as understand the most (**Comprehension**). We also check for test scores on how well their knowledge in the inner workings of the framework (**Workings**). Since we do not want to rely on the assumption of normal distribution, we test for the normality of the dependent variables. From the normality test, we discover that all dependent variables except **Number of difficulties** are normally distributed for each participant group. Thus, for this dependent variable, medians will be used as the expected values, rather than the means.

In order to determine whether any of the categories differed on any of the scales for the dependent variables, mean scores (and standard deviations) are computed for each category on each scale. Using the documentation type as the independent variable and the four dependent measures, the data are subjected to an analysis of variance. In Table 3, the minimalist column is bold-faced to indicate this documentation style has the best performance. Table 4 presents the results of the separate multivariate tests. Multivariate F-tests are conducted to determine which of the dependent variables differ across the various categories. These values are obtained via tests of between-subjects effects using Multivariate Analysis of Variance (MANOVA) with a Scheffe test adjustment [14]. We choose this test to examine the sample sizes, since the three documentation groups in this experiment are unequal. From these results, we observe that one out of four independent variables is significant.

Table 3. The means of all categories

Category (Dependent variable)	Mean		
	Minimalist	***Patterns***	***Jdoc***
1. Semi-Completion (hh:mm:ss)	**0:31:06**	0:33:59	0:36:56
2. Completion Time (hh:mm:ss)	**0:58:11**	1:04:29	1:07:03
3. Comprehension (Scale:0-18)	**14.69**	13.31	14.08

Category (Dependent variable)	Mean		
	Minimalist	*Patterns*	*Jdoc*
4. Workings (Scale: 0-4)	**3.42**	2.81	2.87

Table 4. Multivariate effects of the documentation type on dependent variables.

Category (Dependent variable)	F	Significance
1. Semi-Completion time	1.657	0.197
2. Completion time	2.305	0.106
3. Comprehension	1.077	0.345
4. Workings	4.639	0.012*

Note: * Statistically significant at 0.05 level

In terms of **Semi-Completion** and **Completion** in Table 3, the subjects who use minimalist documentation complete their first compilation and complete the experiment faster than the ones using the other two documentation styles. When looking for the standard significance level of 0.05 (i.e. 95% probability) in Table 4, there is evidence that the patterns group are not significantly slower. Therefore, we conclude that there is no significant difference between patterns and the other two documentation styles as to how long it takes the subjects to complete the experiment. Subjects using minimalist are faster than both of the others perhaps because there is less text to read, while subjects using patterns style are faster than subjects using Jdoc perhaps because it is not cluttered with too much class information such as inheritance and subclasses.

As for **Comprehension**, there is no significant difference between how well the subjects understand the materials. This might be because the students are still able to understand the CmdAdp code in the end, irrespective of the document styles. Their learning may reach a maturation effect after going through the four work tasks of documentation. Furthermore, this can be due to the experiment being conducted at the end of the semester. The participants learn enough from the prior eleven weeks of tutorials and lectures on object-oriented programming to bias their performance in the final stage of the experimental run.

Regarding **Workings**, the subjects in the minimalist documentation group exhibit significantly better workings scores than the other documentation styles at the 5 per cent level. Interestingly, this indicates that the $E1H_0$, $E2H_0$ and $E3H_0$ hypotheses in section 3 are rejected. These rejections show that the patterns documentation and the other two styles are not the same in teaching the subjects about completing the work tasks with the designated settings. Spending more time in directly instructing the coding of the CmdAdp can be more beneficial in having the default result rather than flooding the intermediate undergraduates with too much background information. Too much background information may motivate intermediate undergraduates to try something different. They are more confident to differ since they are equipped with the additional background.

Table 5. Kruskal-Wallis test on the number of difficulties.

Chi-square	Degree of freedom (DF)	Asymptotic significance
2.502	2	0.286

Since the **Number of difficulties** is not normally distributed over the comparison of the three groups, we use the Kruskal Wallis test [15]. With the two-sided asymptotic significant value in Table 5 more than 0.05, the number of difficulties faced by the subjects has no significant difference among the three groups. The participants might not record fully the number of difficulties they have solved the task. In summary, among the strong proxies that confirm minimalist advantages include the fastest semi completion time, the fastest completion time, the highest comprehension and workings scores. Hence, we conclude that minimalist documentation is relatively superior to others in encouraging the positive knowledge transfer strategies of intermediate undergraduates.

5 Conclusion

In this work, a set of philosophies for organizing pedagogical textual and graphical information on the CmdAdp documentation has been proposed. From the results, we realize that the effects of the patterns style documentation are not supreme all the time. Perhaps, for intermediate users, patterns are not always the best. Furthermore, Pressman [16] suggested that patterns are not suitable for every situation. Interestingly, minimalist documentation shows an overwhelming advantage in terms of the intermediate users' completion speed and comprehension in fulfilling requirements.

In order to remove any variation between groups, each group is exposed to the three documentation styles in three different stages. For instance, if a group is given minimalist documentation in the first stage, the group uses patterns-style in the second stage before proceeding with Jdoc in the final third stage of CmdAdp exercise. This provides the opportunity for each group to attempt the three techniques.

The quantitative results show that minimalist documentation did not have a significant impact on the time and comprehension that it took to perform the programming tasks. Nevertheless, in terms of the functional workings of the framework, minimalist documentation had a practically and significantly positive impact, in spite of the fact that the participants were not experts in applying design patterns into programming tasks. The aim of using the most effective documentation is to provide intermediate users with a good process that will lead to faster growth in learning the CmdAdp design patterns. All these results demonstrate the behaviors of intermediate users in using the documentation solutions.

References

1. Martin, R.C., Newkirk, J.W., Koss, R.S.: Agile Software Development: Principles, Patterns, and Practices. Pearson Education Int'l, Upper Saddle River, NJ, pp. 13, pp. 43-84 (2012)
2. Carroll, J.M.: Minimalism beyond the Nurnberg Funnel. MIT Press, Cambridge, MA (1998)
3. Chai, I.: Pedagogical framework documentation: how to document object-oriented frameworks: an empirical study. PhD dissertation, University of Illinois at Urbana-Champaign, IL, http://www.cs.uiuc.edu/research/techreports.php?report=UIUCDCS-R-99-2077 (2000)
4. Johnson, R.: Documenting frameworks using patterns. In: Proc. ACM Object-Oriented Programming, Systems, Languages and Applications (OOPSLA'92), pp. 63-76. ACM Press, Vancouver, British Columbia, Canada (October 1992)
5. Berglund, E.: Designing electronic reference documentation for software component libraries. J. Systems and Software, vol. 68, no. 1, 65-75 (2003)
6. Cockburn, A.: Supporting tailorable program visualisation through literate programming and fisheye views. J. Information and Software Technology, vol. 43, no. 13, 745-758 (2001)
7. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns: Elements of Reusable Object-Oriented Software. Pearson Addison-Wesley, Reading, MA (1994) [Commonly called the "Gang of Four" or "GoF" book]
8. Ho, S.B., Chai, I., Tan, C.H.: Comparison of different documentation styles for frameworks of object-oriented code. Behaviour and Information Technology, vol. 28, no. 3, 201-210 (2009)
9. Ho, S.B., Chai, I., Tan, C.H.: An empirical investigation of methods for teaching design patterns within object-oriented frameworks. International Journal of Information Technology and Decision Making, vol. 6, no. 4, 701-722 (2007)
10. Example of the documentation fragment which was presented in all the three documentation groups, http://pesona.mmu.edu.my/~sbho/Swing/Jdoc/t10jDoc/ 302greetAct.html
11. Example of the documentation fragment that is available in the patterns style documentation, but not available in the minimalist and Jdoc documentation, http://pesona.mmu.edu.my/~sbho/Swing/Composite/t12Pat/302greetAct.html
12. Example of the documentation fragment that is available in the Jdoc documentation, but not available in the minimalist and patterns style documentation, http://pesona.mmu.edu.my/~sbho/Swing/Pattern/t10Pat/GreetingAction.html
13. Beizer, B.: Software is different. In: Patel, D., Wang, Y. (eds.) Comparative Studies of Engineering Approaches for Software Engineering, vol. 10, pp. 293-310. Baltzer Science Publishers, Norwell, MA (2000)
14. Neter, J., Kutner, M.H., Nachtsheim, C.J., Wasserman, W.: Applied Linear Statistical Models. McGraw Hill, Boston, MA (1996)
15. Field, A.: Discovering Statistics Using SPSS, 3rd ed. SAGE Publications Ltd, London, pp. 560-567 (2011)
16. Pressman, R.S.: Software Engineering: A Practitioner's Approach, 7th ed. McGraw Hill, New York, pp. 347-354, pp. 835-837 (2010)