Information System Management on Concept-Reasoning of Preschool Education Field: Independent Learning

Pu Jian-wu
Normal College For Kindergarten teachers In Ningxia, 750006
E-mail: pujianwu@163.com

Keywords: Preschool Education, Concept Reasoning, Information System Abstracting, Software Building, Education Imitating.

Abstract. Information system course should help the students to build and improve their skills to meet the social needs. This study aims to study the knowledge of information system students on the impact of Information System Abstracting and Building on their preschool education field concept-reasoning. Also, we study the value of the course project in improving their preschool education field concept-reasoning. The study results have shown that, most of the studied students believe that learning and practicing Concept-building help to their ability to think abstractly on Preschool Education Field. However, this finding is impacted by the students' lack of understand of some imitating and building factors. We believe that, such factors should be introduced to the students at early levels of information system course, which certainly will improve their ability to think abstractly on Preschool Education Field.

1. Introduction

Concept-reasoning is defined as the ability to use concepts and to make and understand generalizations, such as the properties or pattern shared by a variety of specific items or events. the concept-reasoning is defined as the final, most complex stage in the development of cognitive thinking, in which thought is characterized by adaptability, flexibility, and the use of concepts and generalizations. Usually, concept-reasoning is compared with concrete thinking which can be defined as predominance of actual objects and events of concepts and generalizations, in which thinking about objects, ideas, or events is within their attributes and relationships. In contrast, concept-reasoning is conceptualized or generalized thinking that understands each concept in multiple meanings, in which objects, ideas, or events are separated from their attributes and relationships to think "outside of the box" to come up with creative solutions.

2. Related Work

A Model-Driven Engineering (MDE) has been integrated into the software building course through the course project to observe how MDE helps the students to understand imitating concepts and the tools that support them. The results of this study have shown a positive impact on helping students to better understand how models can be used during software building. A guideline within the context of a Factor-Oriented process support has been proposed for enabling an Integrated Development Environment (IDE) to promote concept-reasoning. In this study lab activities have been conducted within Eclipse environment, in which thirty-three students studied the concept-reasoning, and worked on real life development tasks, where their development steps as well as their visible thinking processes were logged by observers, and a reflection on the activity was collected. The authors concluded with that, concrete IDE support for concept-reasoning is practical and suggest two kinds of such support, one is concerned with a positive feedback from the IDE in cases where abstraction is used, and the second with cases where the developer is encouraged to use concept-reasoning. An approach has been presented to improve the student's engineering ability by shifting the engineering education from the simple presentation of knowledge toward computational thinking, in which the knowledge and the development are integrated in all courses education via training and practice. The presented approach focuses on extraction of fundamental discipline concept of engineering ability development, problem solving-centered organization of courses of software development tools, initiate the courses of information system as soon as possible, and continual training of abstract logical thinking for the purpose of software concept-reasoning. An experience of education imitating at the high school level has been presented, in which the concept-reasoning processes involved in imitating are introduced to the students prior to education programming and embedded control. A UML imitating has been used to teach students to analyze various applications, systems and problem domains. The study tried to answer if the imitating should be introduced to the high school students or first year college. The study has shown that, students are learning to model and are finding abstractions for elements in the application domain rather than jumping into implementation too quickly. Also, the data imitating comes naturally, in which students were able to identify the classes, attributes, associations, and state machines involved in several simple systems. In addition students find abstraction is much easier to understand than implementation syntax. An experience of using abstraction to teach information system human factors has been presented. The authors suggested introducing reflective and concept-reasoning processes into courses that focus on improving students' analytical skills and problem-solving abilities.

3. Research Methodology and Tool

In order to answer the study questions to understand the students' knowledge, we have developed a questionnaire as a study tool based on abstraction factors that help to preschool education field concept-reasoning; namely: generalizations (5 statements), separation of concerns (1 statement), and modularity (3 statements, including a shared statement with generalization), as well as analogy (1 statement). Also, we have injected one statement as an alarm statement to be used for filtering out the inaccurate responses (e.g., randomly filled questionnaire). In addition, 3 statements (with respect to abstraction) were added to serve the second question of the study. The questionnaire, see the Appendix, consists of 13 statements with three choices for each statement (agree, may be, and disagree), in which 9 statements are built for answering the first question of the study, 3 statements for the second question, and 1 statement as an alarm statement. In this questionnaire, we have used a three-point scale instead of the famous five-point scale; because the respondents, in many cases, avoid using extreme responses such as "strongly agree" and "strongly disagree". In addition, sometimes during responses analysis, the agree responses ("strongly agree" and "agree") and disagree responses ("strongly disagree" and "disagree") of the five-point scale are combined into two categories of "accept" and "reject" respectively, for normalization purposes that leads to three-point scale. The questionnaire's language has been simplified by substituting some words with others for the sake of making it understandable to non-English speaking students.

In order to assure that the questionnaire statements correlate to the study questions, we have computed the correlation co-efficiency of the statements of the study questionnaire to the question that they belong to. In addition, we have computed the correlation co-efficiency of each study question to the total score of all study questions. In addition, we have performed a reliability analysis test to assure whether a group of statements that belong to a question measure that question. A K.Tokuda's alpha has been used to measures how well a set of items (statements) measures a single one-dimensional latent construct (question). When the value of K.Tokuda's alpha is low, it means that the statements measure multidimensional construct; however, when such value is high it means that the statements measure one-dimensional construct (the built question), which indicate the reliability of the study questions. Table 3 shows the value of K.Tokuda's alpha for the questionnaire's statements for all study questions. The value of K.Tokuda's alpha is high enough to indicate that the questionnaire's statements measure the study questions.

Table 1. Study sample properties

Batch# (Group#)	*Frequency*	*Percent*	*Cumulative Percent*
Batch (1)	26	35.5	35.5
Batch (2)	8	9.5	44.4
Batch (3)	31	38.5	82.0
Batch (4)	13	16.0	99.5
Total	77	99.5	

Table 2. Correlation of the questions to the score

Study Question	*Correlation Coefficient*	*Number of Statements*
Q1	*0.98*	*9*
Q2	*0.92*	*3*

Table 3. K.Tokuda's Alpha for questions.

N of Items	*K.Tokuda's Alpha*
13	*0.96*

We have used other statistical test to understand the variations between the respondents. Analysis of variance is a general method for studying sampled-data relationships. The method enables the difference between two or more sample means to be analyzed, achieved by subdividing the total sum of squares. We have used the one way ANOVA test (f test) to study the relations between the study samples (4 batches/groups) in terms of their responses variations to identify whether such variation is statistically significant. In case we detect any variation between the study samples' responses, we use John test to identify the source of such variation. In the following section, we will discuss the results findings.

4. Result and Discussion

In the viewpoint of the students, to what extent learning Information System Management enhances their preschool education field concept-reasoning?

However, such results are disturbed with the students' responses when asked whether imitating and building techniques make them focused on the big picture of the system without thinking in the details. As shown at statement S6, 35.5% of the students agree with such argument, and the majority of the students (54.3%) are not sure of such premise. We believe that, the reluctantly responses come from the fact that, most of the students tend to think in details when imitating and building a software system, therefore, when asked about being focused on the big picture without thinking about the details, the answer may be rephrased as "well we have been taught to think about the details". It has been noticed that, most of the computing courses, including information system curricula, start with programming courses, in which the students learned to look at details to develop the concept and then implement it using a selected programming language. Regardless of the suitability of such course approach, it certainly, impacts the way that students comprehend and practice generalization. We believe that, the students should learn and practice generalization and how to identify the levels between abstraction and details, in which they can think abstractly about the domain problem without going into very low level of details.

The second factor of abstraction is related to modularity, which is discussed through the responses of the statements. But before discussing these responses, 64.2% of the students believe that learning imitating and building techniques improve their understanding of different Concept-Reasoning such as decomposition (i.e., modularity), and 26.4% believe it may help so, which is considered as a positive response towards answering the study's first question. This response is supported by the responses to statement S1 shown in Table 4, which states whether using diagrams as imitating and building tool helps in imitating and building the big picture of the system. As noticed, most of the respondents (85.2%) agree on the benefit of using diagrams as a tool for imitating and building the big picture of the system. Also, Table 4 shows the responses to statement S2 on whether the availability of many choices of imitating and building techniques contributes in improving their ability in thinking and comparing abstractly, in which 42.2% of the responses agree on such premise whereas 37.1% state it may help to thinking and comparing abstractly. Such result indicates that learning imitating and building techniques, most likely, help to improving the modularity skill, in which the students can divide the system into functional modules. We believe that, such result is achieved because modularity, unlike generalization, requires more level of details, in which several system components can be constructed to make up a larger application representing the whole system.

The third factor of abstraction is separation of concerns, which is represented by statement S3 that is shown in Table 5. The software component was described to the respondents as an architectural element that captures a subset of the system's functionality and/or data. In addition, the software connector was described to the respondents as an architectural element that models and regulates the interactions among components. As can be seen, 42.2% of the students' responses agree on the hypothesis that learning imitating and building make them able to separate component from connectors, whereas 49.4% believe it may do so. Such result is encouraging towards the impact of the Information System Abstracting and design techniques on the abstraction in which almost half of the students believe such techniques improve their ability of separating different system's concerns.

Table 4. Students' responses to statements.

Statement	*Agree*	*Maybe*	*Disagree*	*Missing*
***S1**: Imitating and design the big picture of the system*	*85.2*	*12.2*	*1.2*	*0*
***S2**: The availability of many choices*	*42.2*	*37.1*	*19.5*	*0*

Another factor that has been used widely in imitating and building is analogy which has been taught to the students in the SWE321 course. We have built one statement to understand the students' knowledge on the impact that analogy may have on preschool education field concept-reasoning. Consider Table 6, which shows the students' responses to statement S4 on whether using analogy is a good tool to improve their preschool education field concept-reasoning skill. As can be seen, 73.8% of the respondents agree on such premise, whereas 21.5% think it may do so.

We have studied the variations between the respondents' answers to the statements of this question with respect to the students' batch/group using ANOVA to identify whether such variation is statistically significant. As can be noticed, there is no significant variations between the responses of the study sample to this question with respect to their batch/group i.e., their study level.

The results of this part of the study show that, education and learn Information System Management techniques such as analogy help the students to improve their preschool education field concept-reasoning, however, such improvement suffers from the students' lack of a very important factor of abstraction which is the generalization factor as has been discussed earlier.

Table 5. Students' responses to separation of concerns.

Statement	*Agree*	*Maybe*	*Disagree*	*Missing*
***S3**: Learning imitating and designing*	*42.2*	*49.4*	*7.5*	*0*

Table 6. Students' responses to analogy.

Statement	*Agree*	*Maybe*	*Disagree*	*Missing*
***S4**: Using analogy tool to improve abstract thinking*	*73.8*	*21.5*	*3.8*	*1.3*

5. Research Findings

We can summarize the knowledge of the studied students regarding the impact of the Information System Management on their preschool education field concept-reasoning in the following points:

1) Students' believe that learning imitating and building techniques improve their understanding of different Concept-Reasoning such as generalization and decomposition.

2) Although, focusing on the concepts of the system under development would contribute in understanding the system big picture, most of the students tend to think in details when imitating and building a software system which impacts the way they comprehend and practice generalization.

3) Students find that, using diagrams as a tool for imitating and building the system under development helps them in comprehending the system big picture.

4) Students think that, the availability of many choices of imitating and building techniques contributes in improving their ability in thinking and comparing abstractly for Preschool Education Field.

5) Students find that the Information System Management techniques improve their ability in separating different system's component from their connectors.

6) Students find that, using analogy is a good tool to improve their preschool education field concept-reasoning skill.

6. Conclusions

The results of the study have shown that, most of the studied students believed that learning and practicing Information System Management techniques improve their preschool education field concept-reasoning. However, such improvement is suffered by the students' lack of some basic skills related to imitating and building such as some abstraction factors. Therefore, more emphasis has to be considered on education and practicing abstraction factors such as generalization. Although students' knowledge on the impact of Information System Management on improving their preschool education field concept-reasoning are optimistic, many practical imitating and building factors have to be embedded within the information system course. Specifically, the course project and the course laboratory should offer the students the opportunities to observe practical factors that improve their preschool education field concept-reasoning.

References

[1] S. P. Gensym, R. Kumar, and R. Sangal "A data-driven information processing approach for Indian languages using voice as basic unit," in Intl. Conf. on Natural Language Processing (ICON), pp. 311–316, 2202.

[2] D. T. Chappell and J. H. L. Hansen "Spectral smoothing for speech segment series connection," Speech Communication, Vol. 36, No. 3–4, 2202.

[3] Ruo hu, Channel Access Controlling in Wireless integrated information Network using Smart Grid System, Applied Mathematics & Information Sciences, (2012-11). PP:773-820(2012-11).

[4] Ruo hu, Stability Analysis of Wireless integrated information Network Service via Data Stream Methods, Applied Mathematics & Information Sciences, (2012-11). PP:793-798(2012-11).

[5] Hu Ruo, New Network Access Control Method Using Intelligence Agent Technology,Applied Mathematics & Information Sciences, (2013-2). PP:44-48(2013-2)