Evaluating The Effectiveness Of The Constructionist Approach To Object-Oriented Systems Analysis And Design Pedagogy

Benjamin Khoo, PhD, New York Institute of Technology, USA

ABSTRACT

A major limitation in traditional classes in systems analysis and design is that students often are unable to "experience" the process in a real world setting. This limitation can be overcome by using the constructionist approach in a local environment that includes the students working in teams to develop a system design based on a real organization. This allows them to be involved in the process to "do" or "construct" knowledge as they work on the system design. This paper describes a case scenario that was developed based on the constructionist approach, to teach students object-oriented systems analysis and design (OOAD) using the Unified Process. A proof of concept evaluation was conducted and the results indicate that this approach is effective in OOAD pedagogy.

Keywords: Object-Oriented, Analysis and Design, Pedagogy, Constructionism

INTRODUCTION

The advent of the computer has stimulated the development of computer-based systems. As computers became more complex, so are the skills required to develop systems that execute on them. In general, information systems (IS) can be defined as computer-based systems that are required for an organization's business application. The 1970s saw more complex systems being developed because of the advancement in technology. As the systems became more complex, more skills were required to develop them. IS development involves a series of processes. Often IS are developed based on the system development life cycle (SDLC) using elicited user requirements. Today IS is a specialized skill area with a body of knowledge that is taught in different courses in the curriculum of most universities.

According to Pressman (Pressman, 1992) the software system development process as consisting of three broad generic phases - the definition, development and maintenance phases. The definition phase defines the "what" of the software system, the development phase defines the "how" and the maintenance phase defines the support and future necessary changes. Almost every text on software development includes a SDLC model, there are some variations but, in general, the basic phases or activities are always present. The basic phases that are present are the analysis, design, testing, implementation and maintenance phases. While each of the analysis, design, testing, implementation and maintenance phases need to be performed in all cases, there are a number of ways their interactions can be organized. The major models that are in use today are the Spiral model (Boehm, 1988) and the Unified Process model (Jacobson, 1999; Kruchten, 2000).

The explosive growth of the object-oriented technology in the 1990s has resulted in a paradigm shift in systems development methodologies. The Unified Process (Jacobson, 1999; Kruchten, 2000) is a direct result of the shift initiated by object-oriented technology. There are four main iterative phases in the Unified Process (UP) - Inception, Elaboration, Construction and Transition. Each of the phases has associated workflows and can have multiple iterations. See Figure 1.

The Inception phase includes tasks such as defining the scope, planning, feasibility studies and system investigation to model system processes and define requirements. The Elaboration phase includes modeling the system processes as usecases, realizing the usecases and extracting a preliminary set of requirements from them. The goal of the Elaboration phase is to produce a system design in the form of a class diagram that meets the requirements. In the Construction phase, programming code is written with the class diagram as blueprint and is tested based on requirements. When the software system has completed all the tests successfully, it is released and deployed on the clients' computers. The Transition phase involves releasing the software system to the client.

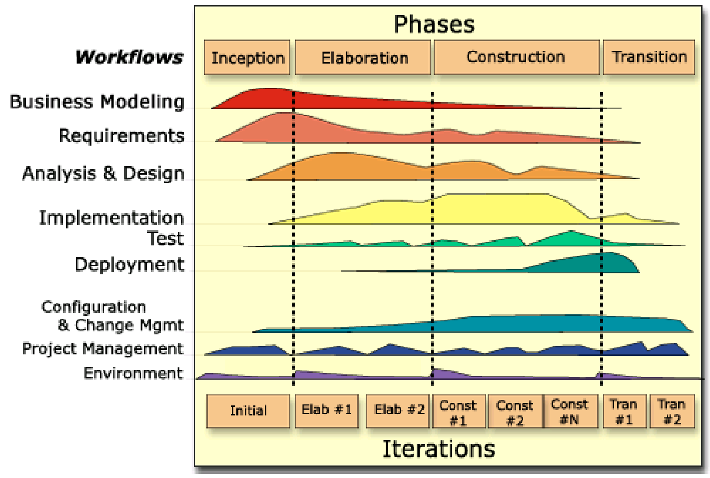

Figure 1. The Rational Unified Process (from Rational University)

This paper describes an effort to harness the strengths of any localized academic environment and utilize it as a reservoir of resources to assist students, through the Unified Process, to develop a systems design based on a case scenario for "California Community College" using the constructionist approach (Papert, 1980a). The constructionist approach uses constructive tasks to impart knowledge. The objective is to motivate learning through activity. In this way, learning is made more effective. The great Chinese sage Confucius once said, "I hear and I forget. I see and I understand. I do and I remember." This approach allows the students to have a hands-on experience through interactions with the available local resources to iteratively elicit systems requirements and verify processes, model the system by using the industry's top rated modeling tool - Rational Rose, learn the Unified Process and ultimately, develop a system design based on the requirements.

LITERATURE REVIEW

Constructionism is a major principle in contemporary education theory and a strategy for learning. There are two facets to constructionism - that learning takes place as a result of actively constructing new knowledge and that learning is effective when "constructing" or "doing" activities that are personally meaningful. It is widely accepted in educational circles that an important part of the learning process consists of "hands-on" construction. Constructionism has been supported by the success of children educational activities based on building blocks (Resnick, 1991). It is a well-established methodology for learning (Papert, 1991; Resnick, 1991). The constructionist approach uses constructive tasks to impart knowledge. Its goal is to develop creativity and motivate learning through activity. Constructionism asserts that knowledge is not simply transmitted from the teacher to students, but is actively constructed in the mind of the learner through various hands-on activities. In describing constructionism, Papert wrote "If one eschews pipeline models of transmitting knowledge in talking among ourselves as well as in theorizing about classrooms, then one must expect that I will not be able to tell you about my idea of constructionism. Doing so is bound to trivialize it. Instead, I must confine myself to engage you in experiences (including verbal ones) liable to encourage your own personal construction of something in some sense like it. Only in this way will there be something rich enough in your mind to be worth talking about." (Papert, 1991).

Constructionism suggests that learners make their ideas by constructing their own knowledge structures. It has been shown that learning is more effective when it is activity-based rather than passively received (Brown, 1989). The active "constructing" or "doing" tasks leads to discovery.

The concept of discovery learning is not new. Discovery learning can be described as experimentation with some extrinsic intervention -- clues, coaching and a framework to help learners get to a reasonable conclusion. It has appeared many times in educational philosophy, Dewey stated, "There is an intimate and necessary relation between the processes of actual experience and education" (Dewey, 1938). It is also supported by learning theorists/psychologists such as Piaget, Bruner and Papert, "Insofar as possible, a method of instruction should have the objective of leading the child to discover for himself." (Bruner, 1967) However, it has never received overwhelming acceptance even though it has enjoyed a few positive swings of the educational-trend pendulum in American education (Jacobs, 1992).

The learner draws on his own experience and prior knowledge to discover the knowledge to be learned. This is embodied in a personal, internal, constructionist environment. Bruner stated that "Emphasis on discovery in learning has precisely the effect on the learner of leading him to be a constructionist, to organize what he is encountering in a manner not only designed to discover regularity and relatedness, but also to avoid the kind of information drift that fails to keep account of the uses to which information might have to be put." (Bruner, 1967)

MOTIVATION

The main motivation for this project is to provide an experiential learning experience for the students. This will complement the theoretical knowledge that is taught in the classroom. Learning psychologists like Resnick, Brown, Piaget, Bruner and Papert have long expounded the need for "hands-on" learning experiences to improve pedagogy effectiveness.

In learning object-oriented systems analysis and design (OOAD), students often are unable to grasp the full implications of the techniques found in textbooks. The knowledge found in the written text and in diagrams is passive in that they do not fully illustrate the process. Questions from textbooks usually are constrained by scope and space. They often address issues far from the academic domain and thus, problem statements in textbooks do not convey a sense of reality to the students who are addressing them. Students are not placed in a situation where they can interact with the actors involved in each system and thus, are not able to appreciate the problem or issues at hand. The traditional mode of teaching OOAD did not allow the student to actively experience any interaction with a real person in trying to address an IS problem that requires the development of a system design. An approach to overcome this limitation was initiated to harness the strengths of the localized academic environment and to utilize it as a model of the world where the problem that needs to be addressed exists. The problem statement requires that various usecases, for example for registration, borrowing and returning books to the library, scheduling activities for the sports and recreation department, be developed based on a case scenario for a fictitious academic institution called "California Community College (CCC)" using the constructionist approach (Papert, 1980a). This approach requires that the students iteratively interact and conduct system investigations in the real local academic institution so as to gather enough information about the processes in each of the usecases. The information gathered will assist the students bring to "life" similar usecases that needed to be addressed in the CCC. Thus, this constructionist approach enables the student to actively participate in the various tasks in OOAD, which will help them to learn as a result of their "doing."

APPROACH

The class OOAD assignment requires that students develop a system design for a specific area (or department) in CCC. Beginning with the Inception through the Elaboration phases, students are encouraged to approach and interview individuals working in the various departments on campus to obtain realistic procedures that are applicable to the areas that they are working on. The information elicited are analyzed and captured as Systems Requirement Specifications (SRS) where each requirement is listed by functionality and itemized numerically, for example, a functional requirement = FR1.0. Students are required to identify the scope (or boundary) of the target system and the main actors. An actor is defined as an external person or sub-system that initiates or receives information from the target system. Students are then to determine the goals that each initiating actor wants accomplished the services, functions or "uses" that the target system should provide. The processes (or procedures) to achieve each of these goals describe a usecase. Thus, each usecase is one way of using the target system. Each usecase is then diagrammatically represented in a usecase model. The overall usecase model for the target system is then developed, using Rational Rose J Edition (currently, Rose has been updated and is now called Rational Software Architect). This approach provides students the opportunity to have the "hands-on" experience required to be familiar with each usecase. This approach utilizes the resources available locally in the various departments of the local academic institution to provide inputs for developing each usecase for CCC. This way, students "learn by doing" and are in a better position to describe the usecase narrative. Each usecase is then described on a Usecase Specification (US) that describes the procedural flow of the usecase. The US is a static representation of a usecase.

In this way, the students' need to experience 'real' systems and learn through 'doing and experiencing', is met. This approach is based on the philosophy of constructionism (Papert, 1980a; Papert, 1980b). As constructionism provided the fundamental basis for the approach to develop the OOAD assignment, a vehicle has to be chosen to deliver the approach. The availability of the academic alliance program called Software Engineering for Educational Development (SEED) from the Rational Corporation, which is now part of IBM Inc., with its leading industrial OOAD tool called Rational Rose (at the present time, Rose has been updated and is now called Rational Software Architect) provided an excellent candidate.

MODELING TOOL

This OOAD assignment is intended to get the student involved in an OOAD exercise and come away with knowledge of what a good system design using the UP requires. A few criteria are necessary for this project to be successful. One major criterion is that the students should be able to interact with a local academic environment; so as to appreciate the real-life procedures and system needed to develop each usecase. This is provided through the local campus community. This also provides the hands-on experience for students to grasp and learn the UP concepts.

Another important criterion is that the OOAD artifacts for the system design have to be modeled diagrammatically. This is to allow the students to translate their understanding of the UP into Unified Modeling Language (UML) constructs for the target system. Often the UML constructs are available on OOAD tools. The modeling of the target system UML constructs in an OOAD tool provided an important OOAD artifact - the OOAD model, which includes the usecase model, sequence diagrams, collaboration diagrams and class diagrams. Modeling OOAD in a diagrammatic tool also provided hands-on experience for students to develop the system design using UML construct. In order for this project to work, students should be able to easily get access to an OOAD tool. The success of the whole project actually depended on the students being able to get access to this OOAD tool.

The IBM Scholars (and Academic Alliance) Program made Rose available to academic institution free of charge. Rose has been consistently ranked as the most favored OOAD tool in industry and arguably has about 90% of the current OOAD tool market. This made Rose the ideal tool for this project. Through the SEED program, Rose was made available to students. Students are also exposed to the industry's leading OOAD tool. This project provided the students an experiential knowledge of OOAD.

CASE SCENARIO

The project required that the OOAD assignment be one that mimics a real academic institution (or a college campus). The scenario used to develop such a system is called a case scenario. A case scenario describes a situation that exists in the world. Information for the case scenario can be gathered by means of interviewing, reading and observing a real situation in the local campus. Information from the local campus can be used to provide input to the UP and used to plan, analyze, model and design the system design for the target system in the OOAD assignment.

The case scenario is one that is derived from a practical and real academic institution. The scenario is the "California Community College" (CCC). CCC is an educational institution that provides academic undergraduate courses to students working towards an undergraduate degree. There is currently no computerized system but there is now a budget set aside to computerize the operations in the various departments in the college. The possible modules/areas for CCC that require systems design are:

1. Registrar Office
2. Residence/Housing Office
3. Requisitions Office
4. Health Services
5. Plant & Maintenance Office
6. Continuing Education Office
7. University Computing Services Office
8. Grants & Endowment Office
9. Human Resources Office
10. Student Services Office
11. Sports & Recreational Office
12. Campus Security Office
13. Library
14. Campus Cafeteria

This case scenario served as the background for the OOAD assignment. The OOAD assignment is implemented based on the available campus resources.

IMPLEMENTATION

Students in the class were divided into teams of 3 or 4 members (with approval) each. Each team was required to develop a specific module of the CCC OOAD assignment. The systems design was to be developed using the Rational Rose J Edition tool.

The term-long OOAD assignment will allow the student to practice the skills learned in class. The end product will be a professionally produced system design for an organization. Each team will choose the project module. The choice should depend on the ease of access to and cooperation from the department or unit, and on the perceived information systems needs of the organization. Teams were expected to develop their respective modules based on the Unified Process concepts (which include the business modeling, requirement definition, analysis and design activities). Each module chosen by a team required at least six usecases. Each team member is required to work on two usecases. The assignment will be completed in stages. At the end of each stage, each team will submit one portion of the project (a "deliverable"). All deliverables are to be submitted both electronically as well as a paper copy for instructor comments. Deliverables at each phase of the project will be required at pre-determined dates. See Figure 2. Teams are expected to follow the deliverable schedule to ensure that the project is completed in the allocated time. Towards the end of the term, teams are expected to present their system design module to the class.

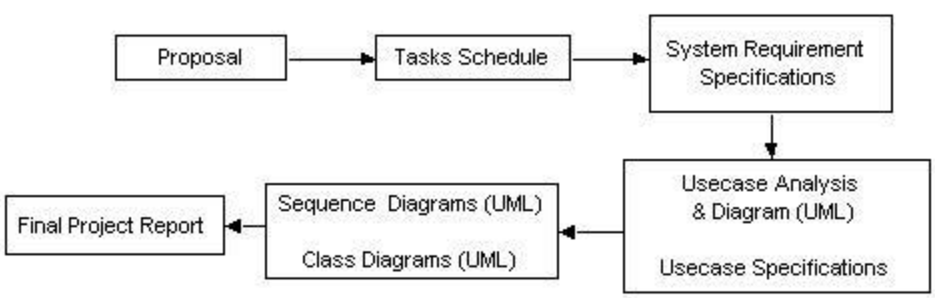

Figure 2. Schedule of deliverables for each stage of the project.

The final product is required to be professionally prepared and organized, containing all the earlier deliverables as well as supplementary materials, if available. At the end of the term, students were required to take an OOAD test to determine their knowledge or understanding of the OOAD modeling concepts.

RESEARCH METHODOLOGY

The complexity of both the human learning process and OOAD process makes their interaction less predictable than we would like. Even the best intentions can result in unusable systems or, more often, in systems with problems.

To evaluate the effectiveness of the constructionist approach, student participants in the OOAD classes were tested on their knowledge of OOAD modeling at the end of each term. The body of knowledge required for OOAD modeling includes understanding the Unified Process (Jacobson, 1999; Kruchten, 2000) and usecases; the purposes for using usecases; the ability to analyze a case/scenario based on information elicited/received to derive the requirements, to define the usecases involved, to identify the components involved in each usecase and in the system, to identify the best diagrammatic artifacts (for example, usecase modeling, sequence, collaboration, etc diagrams) to realize/model the usecases and to derive the class diagrams from the usecase realization (Booch, et al, 1999; Rumbaugh, et al, 2005; Fowler, 2004; Quatrani, 2000).

A group of students (called the "Traditional" group with 46 students) in the OOAD class were taught OOAD using the traditional pen and paper approach. One term later, another group of students (called the "Constructionist" group with 51 students), also in the OOAD class, was taught OOAD using the constructionist approach. The class instructor, teaching materials, content and test questions were the same for both groups except for the assignments. The Traditional group was given six short case scenario assignments while the Constructionist group was given the term-long project with the constructionist approach as assignment. The test for each group was administered after all the assignments were completed for the term.

The Traditional and Constructionist student groups were tested using 9 key OOAD scenario-based questions based on a standardized industrial "OOAD with UML" test. The percentage of correct answers was tabulated for each question for both groups of students. The outcome-related question and hypotheses are:

Question:

Can the constructionist approach to OOAD pedagogy improve student's knowledge of OOAD modeling?

NULL Hypothesis, H_O:
$\mu_{Traditional} - \mu_{Constructionist}$ = or > 0

The constructionist approach to OOAD pedagogy will not improve student's knowledge of OOAD modeling or will have no effect on the student's knowledge of OOAD modeling. (The Traditional student group will perform equally or better than the Constructionist student group.)

Alternative Hypothesis, H_A:
$\mu_{Traditional} - \mu_{Constructionist} < 0$

The constructionist approach to OOAD pedagogy will improve student's knowledge of OOAD modeling. (The Traditional student group will perform worse than the Constructionist student group or post-test student group will perform better than the pre-test student group.)

The data collected from the Traditional and Constructionist groups are as follows:

Table 2. Percentage of correct answers for each question by group

Test Questions	Traditional (%)	Constructionist (%)
1	55.8	66.1
2	50.6	55.4
3	54.3	50.0
4	68.1	75.0
5	48.4	59.0
6	41.8	55.4
7	81.9	94.7
8	51.8	64.3
9	33.2	57.2
Average	54.0	64.1

This hypothesis will be tested using the t-statistics. The t-statistics will be calculated using SPSS version 12.0 for Windows.

Paired Samples Statistics

		Mean	N	Std. Deviation	Std. Error Mean
Pair 1	Tradn.	53.989	9	14.1853	4.7284
	Constr.	64.122	9	13.6153	4.5384

Paired Samples Correlations

	N	Correlation	Sig.
Pair 1 Tradn. & Constr.	9	.851	.004

Paired Samples Test

	Paired					t	df	Sig. (2-tailed)
	Mean	Std. Deviation	Std. Error Mean	95% Interval of Difference				
				Lower	Upper			
Pair 1 Tradn.-Constr.	-10.1333	7.6197	2.539	-15.9904	-4.2763	-3.990	8	.004

The t-statistic for $t_{8,\ 0.95}$ where α = 5% at the 95% confidence interval is -1.860. The null hypothesis will be true if t is > or = -1.860 but since the test statistic has a value of -3.990, the null hypothesis is rejected. The alternate hypothesis is thus proven to be statistically true: The constructionist approach to OOAD pedagogy will improve student's knowledge of OOAD modeling.

CONCLUSION

The experiment above was conducted as an effectiveness measure of the constructionist approach to OOAD pedagogy. The results obtained from the experiment indicated that this project was successful. The constructionist approach to OOAD pedagogy based on the case scenario has fulfilled its primary motivation.

RECOMMENDATIONS FOR FUTURE RESEARCH

This experiment evaluated the effectiveness of the constructionist approach to OOAD modeling. The efficiency of constructionism in assisting students develop knowledge structures in the process of OOAD modeling can be investigated in the future.

ACKNOWLEDGMENTS

I would like to thank all the students in my OOAD classes, both past and present, which had contributed feedback. It is such feedback that initiated this project and ultimately, made this project a success. I would also like to thank IBM, in general, and the SEED program in particular, for providing the Rational Rose J Edition tool to the university and its students free of charge.

AUTHOR INFORMATION

Benjamin Khoo completed his Ph.D. (Information Systems) at the University of Maryland, Baltimore County. He is a member of two honor societies and was awarded the Phi Kappa Phi Dissertation Research Grant. He has published regularly in the major information systems journals. He is interested in both basic and applied research to further the effectiveness, usability, and ultimately the utility of information systems; and also the pedagogical issues related to these areas. Prior to becoming an academician, he was a member of the Technical Staff (Software Engineer) of a large telecommunication corporation.

REFERENCES

1. Boehm, B. W. (1988) A Spiral Model for Software Development and Enhancement. *IEEE Computer*, 21(May), 1988, pp. 61-72.
2. Booch, G, Rumbaugh, J., Jacobson, I. (1999) *The Unified Modeling Language User Guide*. Reading, MA.: Addison-Wesley.
3. Brown, J.S., Collins, A., Duguid, P. (1989) Situated Cognition. In Robert W. Lawler & Masoud Yazdani (Eds.), *Artificial Intelligence and Education*, Volume 2, 1989, pp. 254-268.
4. Bruner, J.S. (1967) *On Knowing: Essays for the Left Hand*. Cambridge, Mass: Harvard University Press.
5. Dewey, J. (1938) *Experience and Education*. New York: MacMillan.
6. Fowler, M. (2004) *UML Distilled*, Third Edition. Reading, MA.: Addison-Wesley.
7. Jacobs, G. (1992) Hypermedia and Discovery-Based Learning: A Historical Perspective. *British Journal of Educational Technology*, Volume 23, Number 2, 1992, pp.113-121.
8. Jacobson, I., Booch, G., and Rumbaugh, J. (1999) *The Unified Software Development Process*. Reading, MA.: Addison-Wesley.
9. Kruchten, P. (2000) *The Rational Unified Process - An Introduction*. 2nd edition. Reading, MA.: Addison-Wesley.
10. Papert, S. (1980a) Computer-based microworlds as incubators for powerful ideas. In R. P. Taylor (Ed.), *The Computer in the School: Tutor, Tool, Tutee*. New York: Teacher College Press.
11. Papert, S. (1980b) *Mindstorms: Children, Computers, and Powerful Ideas*. New York: Basic Books.
12. Papert, S. (1991) Situating Constructionism. In Idit Harel & Seymour Papert (Eds.), *Constructionism.* New Jersey: Ablex Publishing Company.
13. Pressman, R.S. (1992) *Software Engineering: A practitioner's Approach*, McGraw-Hill, pp. 34 - 36.
14. Quatrani, T. (2000) *Visual Modeling with Rational Rose 2000 and UML*. Reading, MA.: Addison-Wesley.
15. Resnick, M. (1991) Xylophones, Hamsters and Fireworks: The Role of Diversity in Constructionist Activities. In Idit Harel & Seymour Papert (Eds.), *Constructionism*. New Jersey: Ablex Publishing Company.
16. Rumbaugh, J., Jacobson, I., Booch, G. (2005) *The Unified Modeling Language Reference Manual*, Second Edition. Reading, MA.: Addison-Wesley.