구조지향 방법론이 객체지향 방법론에 미치는 영향 : 지식간섭의 관점에서

김 인 재[†]·정 덕 훈[†]

요 약

새로운 정보기술의 생명주기는 점점 짧아지고 그 기술이 점차 복잡해짐에 따라 그 기술을 이해하기가 어려워 진다. 본 연구의 목적은 객체지향 기술의 채택에 관한 이해를 돕기 위하여 객체지향 기술이 전통적인 소프트웨어 프로세스 기술과는 다른 패러다임의 기술이기 때문에 프로세스 중심의 구조지향 기술로부터 영향을 받을 것으로 판단된다. 본 연구는 구조지향 방법이 객체지향 방법론에 미치는 지식간섭 현상을 일종의 2×2 팩토리얼 유사 실험계획법을 사용하여 실증적으로 분석하였다. 구조지향 방법론과 객체지향 방법론을 사용한 기간을 2개의 독립변수로 선택하였고 객체지향 방법론에 대한 사용 편의성을 1개의 독립변수로 설정하였다. 자료는 현재 활동중인 구조지향 및 객체지향 방법론에 모두 경험이 있는 자료처리와 관리협회(DPMA : Data Processing and Management Association) 회원으로부터 수집하였다. 최종 결과는 구조지향 방법론의 경험이 객체지향 방법론의 사용편의성에 부의 영향을 주는 것으로 나타났다. 향후 연구과제로 문화가 다른 국가에서 유사한 연구 수행한 후에 그 결과를 비교하는 것도 의미가 있을 것이다.

The Effects Of Structured Methods On Object Orientation : A Knowledge Interference Prespective

Injai Kim[†]·Dukehoon Jeong[†]

ABSTRACT

The life cycle of new information technologies is getting shorter, and the technologies are becoming more complex and difficult to understand. The need to better understand adoption of object orientation motivates this paper. Adoption of object orientation should certainly be influenced by prior software process technologies, such as the process-oriented structured methods, because object orientation is considered to be a paradigm shift from conventional software process technologies. This study aims to empirically analyze knowledge interference of the structured methods with object orientation. A two factorial quasi-experimental design is set forth. The period of experience using the structured methods and the period of experience using object orientation are selected as two independent variables, and the perceived ease of use is chosen as one dependent variable. Data are gathered from active members of Data Processing Management Association (DPMA), who have experiences in using both the structured methods and object orientation. The final results empirically show that previous experience using the structured methods negatively influences the perception of using object orientation that is one of critical factors to technology adoption. It is suggested here that a future study dealing with the same research topic in other countries will provide new insights about comparative studies.

키워드 : 구조지향 방법론(Structured Methods), 객체지향 방법론(Object Orientation), 기술수용(Technology Acceptance), 지식간섭(Knowledge Interference)

1. Introduction

A new paradigm is a product of social changes. System developers have changed their viewpoints about processes and data in system development. Processes are emphasized in the structured methods. The structured methods, developed in the late 1960s and in the early 1970s, introduced standard methodologies by which a system could be divided into process-oriented modules. The process-oriented modules, mostly programmed in third-generation languages, could be kept flexible to accommodate frequent changes in system development processes. The information engineering approach emphasized data. Information engineering [8], which emerged in the late 1980s, assumed that (1) data lie at the center of modern data processing ; and (2) data are stable but processes are not. The object-orientation paradigm,

※ 본 연구는 동국대학교 논문게재연구비 지원으로 이루어졌음.

† 정 회 원 : 동국대학교 정보관리학과 교수

논문접수 : 2002년 3월 7일, 심사완료 : 2002년 7월 25일

which emerged in the mid-1980s, focused not on processes but rather on objects that encompass processes and data. The concepts of encapsulation and inheritance are based upon hierarchical layouts of objects. Objects are data entities with which operations and processes are combined (encapsulation), and an object can inherit characteristics from single or multiple ancestors (inheritance). Inheritance is important because it enforces consistency on the definitions of related objects. Advocates for the object-orientation paradigm insist that these two concepts make it possible to reuse programming code and develop a reliable system [1, 2, 6, 13]. Even though the basic change of these principles has brought out many different opinions, both object orientation and information engineering are based on underlying concepts of the structured methods, and thus, are variations of this approach. The structured methods, information engineering, and object orientation are new ideas, and the others are tools to support these concepts. For example, the third generation languages (3GLs), such as COBOL and FORTRAN, are appropriate for use with the structured methods because these languages are designed for implementing decomposed functions.

Several empirical studies of object orientation show possible knowledge interference of procedural languages with object-oriented languages [5, 11] and differences between the cognitive process of novice and experienced programmers [3, 12]. Detienne and Davis [4, 5] investigate the design strategies of object orientation for both projects using object orientation for procedural and declarative problems. The external validity of these results remains questionable because the problem set was given to small number of subjects in a simulated situation. This paper empirically analyzes the existence of the knowledge interference of the structured methods with object orientation at industry settings.

2. Research Design

One key variable affecting technology adoption is chosen from the Technology Acceptance Model (TAM) [4]. In the basic TAM, behavior intention to use a new technology depends mostly upon perceived usefulness and perceived ease of use of the new technology. To empirically test the interaction between the structured methods and object orientation, a two-by-two quasi-experimental design is set forth. The length of experience in using the structured methods and the length of experience in using object orientation are selected as the two independent variables.

The focus of this research design is on empirically testing the knowledge interference of the structured methods with object orientation. The previous experience in using the structured methods may affect the perceived ease of use of using object orientation. The length of experience in using the structured methods and the length of experience in using object orientation are categorized into two groups, such as high and low, according to the median value of each length of experience. Four cells are built on theses two categorical variables : the level of using the structured methods and level of using object orientation. The values in each four cell represent the perceived ease of use in using the object-oriented analysis and design (OOAD), and object-oriented programming (OOP).

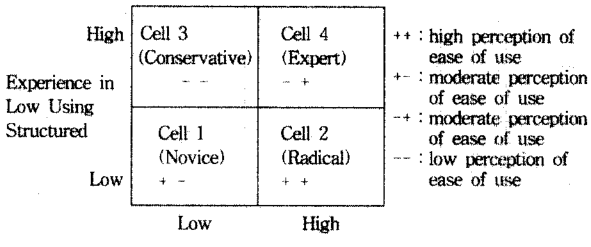
Cell 1 (Novice) : The subjects in this cell do not have much experience in systems development, neither the structured methods nor object orientation. Their previous experience in using the structured methods may not have much effects upon their use of the object orientation, and they are not accustomed to using object orientation. Their perceived ease of use of using object orientation will be moderate compared with the perceptions of using object orientation in the other cells.

Cell 2 (Radical) : The subjects in this cell have much experience in using object orientation, but not much experience in using the structured methods. Their limited experience in using the structured methods may not affect their perceptions of using object orientation. Their perceptions of ease of use will be the highest among the four cells.

Cell 3 (Conservative) : The subjects in this cell have much experience in using the structured methods, but not much experience in using object orientation. Their experience in using the structured methods may strongly affect their perceptions of using object orientation. Their perceptions on ease of use will be the lowest among the four cells.

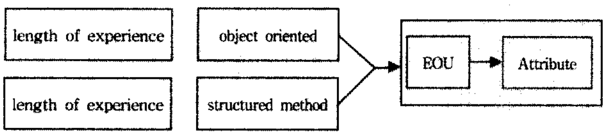
Cell 4 (Expert) : The subjects in this cell have much experience in using both the structured methods and object orientation. Even though their experience in using the structured methods may affect their use of object orientation, they also have strong perceptions about using object orientation.

Thus, there seems to be a relatively strong interaction between the two methodologies. Their perceptions of ease of use will be moderate compared with those of the other cells. The relationship between individual experiences and perception of ease of use is shown in (Figure 1).


High
Low
Experience in
Low Using
Structured
Cell 3
(Conservative)
- -
Cell 4
(Expert)
- +
Cell 1
(Novice)
+ -
Cell 2
(Radical)
+ +
++ : high perception of ease of use
+- : moderate perception of ease of use
-+ : moderate perception of ease of use
-- : low perception of ease of use
Low
High
Experience in Using Object Orientation

(Figure 1) The effects of individual experiences on perception

Research Question : Is there any interference of the structured methods with object orientation? The length of experience in using the structured methods and the length of experience in using object orientation are selected as two independent variables for analyzing the interaction between these two methodologies. The basic idea behind the interference is that the more a person gets used to using the structured methods, the less he or she feels comfortable in using object orientation. For example, the perceptions of subjects in Cell 2 (Radical) should be significantly different from those in Cell 3 (Conservative).



(Figure 2) TAM Relations

3. Data Collection And Analysis

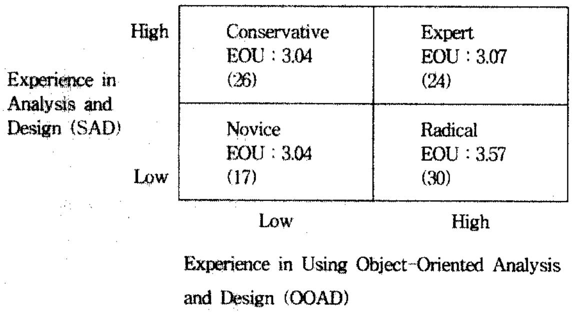
Data were gathered from members of Data Processing Management Association (DPMA). Nine DPMA chapters across four mid-land states in U.S. participate in this survey. One hundred and nine subjects having experiences in using both the structured methods and object orientation responded to the structured questionnaires (response rate = 15 %). The Cronbach's alpha for the measure of perceived ease of use is respectively 0.868 for the object-oriented analysis and design (OOAD) and 0.901 for object-oriented programming (OOP). The period of experience in using the structured methods ranged from less than one month to three hundred months for structured analysis and design (SAD), and from less than one month to three hundred twelve months for structured programming (SP). The median values of these experience periods, were both one hundred twelve months. The period of experience in using object orientation ranged from less than one month to ninety-seven months for both OOAD and OOP. The median values of these experience periods were both six months. Based on these median values, two-by-two factorial designs for OOAD and OOP can be set forth. The summary of the research variables is shown in <Table 1>.

<Table 1> The Operationalization of Research Variables

Research Variables	Previous Measure	Item Type (Number of Items)	Cronbach's α
Ease of Use	[4, 9, 10]	A 5-point Likert Scale (6 Items)	OOAD : 0.868
			OOP : 0.901
Amount of Experience in Using Structured Methods	[7]	Metric (Unit : Number of Months)	-
Amount of Experience in Using Object Orientation	[7]	Metric (Unit : Number of Months)	-

4. Results

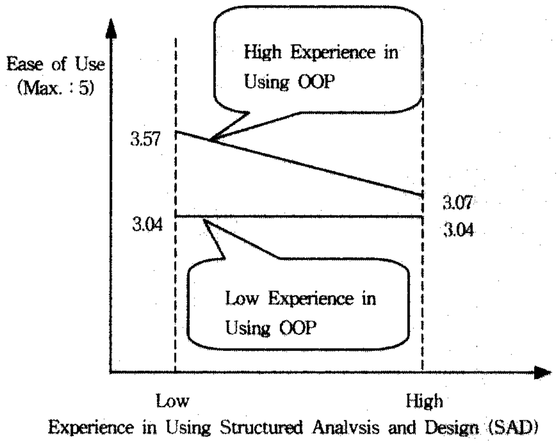
The results to research question are obtained and showed knowledge interaction of the both OOA and OOD. The mean of ease of use (EOU) and number of subjects for OOAD are shown in (Figure 3).



( ) : the number of subjectes

(Figure 3) The Ease of Use (EOU) of OOAD

An ANOVA for EOU of OOAD showed both main and interaction effects : the effect of the experience period of using SAD ($F = 4.219$, $p < 0.05$), the effect of the experience period of using OOAD ($F = 3.595$, $p < 0.10$), and the inter action effect of both (F = 3.386, p < 0.10). When a strict criterion (= 0.05) is applied, the effect of SAD is significant . That means that the previous experience in using SAD influences EOU more than the other experience does. The significantly no difference between Conservative and Expert (F = 0.02), and the significant difference between Novice and Radical (F = 6.96, p < 0.01) were caused by the high experience in using SAD. The more the groups, such as Conservative and Expert, were exposed to using SAD, the less the groups had learning effects in using OOAD. Two contrast groups, Conservative and Radical, differed significantly as expected (t = -3.52, p < 0.01). Thus the statistical analyses supported a visual effect of a graph in (Figure 4).



(Figure 4) The Interaction Effect between SAD and OOAD

The mean of EOU and number of subjects for OOP is shown in (Figure 5).

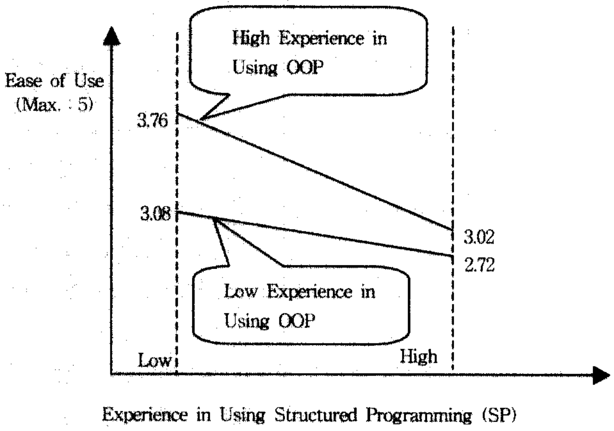
Experience in Using Structured Programming (SP)		
High	Conservative EOU : 2.72 (27)	Expert EOU : 3.02 (22)
Low	Novice EOU : 3.08 (18)	Radical EOU : 3.76 (29)
	Low	High

Experience in Using Object-Oriented Analysis and Design (OOAD)

( ) : the number of subjectes

(Figure 5) The Ease of Use (EOU) of OOP

An ANOVA for EOU of OOP showed not the interaction effect but the significant main effects : the effect of the experience period of using SP (F = 17.567, p < 0.001), the effect of the experience period of using OOP (F = 12.406, p < 0.001), and the interaction effect of both (F = 1.946). This demonstrated the effect of knowledge interference. The difference between Conservative and Novice (F = 3.35, p < 0.10) and the difference between Expert and Radical (F = 16.16, p < 0.001) imply that the more a person has experience in using SP, the less he or she is relatively comfortable in using OOP. The partial learning effect in using OOP is also shown : the difference between Conservative and Expert (F = 2.50) and the difference between Novice and Radical (F = 11.86, p < 0.001). Two contrast groups, Conservative and Radical, differ significantly as expected (t = -5.91, p < 0.001). Thus the statistical analyses supported a visual effect of a graph in (Figure 6).



(Figure 6) The Interaction Effect between SP and OOP

The summary of results are as follows on the <Table 1> and <Table 2>.

<Table 1> Result of Research Question

Research Question		TEST		Result
EOU of OOAD		ANOVA	Main & Interaction	support
Experience	SAD	ANOVA	F = 4.219 (95%)	support(**)
Experience	OOAD	ANOVA	F = 3.595 (90%)	support
Interaction effect		ANOVA	F = 3.386 (90%)	support
EOU of OOP		ANOVA	Interaction	not support
		ANOVA	Main	support(**)
Experience	SP	ANOVA	F = 17.567(99%)	support(**)
Experience	OOP	ANOVA	F = 12.406(99%)	support(**)
Interaction effect		ANOVA	F = 1.946	not support

** significant

<Table 2> Comparison of two quasi-experimental design

Method	Two-quasi-experiments			

SAD	Conservative and Expert
	Novice and Radical
		F = 0.02
		F = 6.96
			no difference
			difference
				not support
				support

SP	Conservative and Novice
	Expert and Radical
		F = 3.35
		F = 16.16
			difference
			difference
				support
				support

OOP	Conservative and Expert
	Novice and Radical
		F = 2.50
		F = 11.86
			difference
			difference
				support
				support

5. Conclusions

From a two-factorial quasi-experimental design, the knowledge interference of the structured methods with object orientation is empirically tested in industry settings. EOU from Technology Acceptance Model is one of two variables that can influence technology adoption. The evidence that previous experience using the structured methods influences EOU with object orientation suggests two possible training methods for systems professionals to gain maximum learning effects : (1) one method for persons having much experience in using the structured methods and (2) another one for persons having little experience in using the structured methods. If experienced programmers tend to organize their logic along the lines of functional relations [3], the way of thinking basically differs from that of object orientation. For example, many contrast examples may help experienced structured programmers to understand object orientation. How to reduce the knowledge interference of the structured methods with object orientation will be critical to future adoption of object orientation. The existence of the knowledge interaction on research findings may suggest more effective training method for object orientation. The next stage of this research is to conduct a similar study across different cultures. Such a study will provide a valuable basis for managing information technologies of global organizations. Several research limitations are imposed on this research. While the subjects are chosen to ensure variety, the participating subjects are confined to the one organization in the mid-west area. Nationwide data may increase external validity of the study.

References

[1] G. Booch, "Object-Oriented Development," IEEE Transactions on Software Engineering, SE-12, No.2, pp.211-221, 1986.

[2] J. Burch, "Object-Oriented Approach : Magic Bullet or Viable Development Methodology?," Journal of Systems Management, Vol.44, pp.9-42, 1993.

[3] S. P. Davies, D. J. Gilmore, & T. R. Green, "Are Objects That Important? Effects of Expertise and Familiarity on Classification of Object-Oriented Code," Human Computer Interaction, Vol.10, pp.227-248, 1995.

[4] F. D. Davis, R. P. Bagozzi, & P. R. Warshaw, "User Acceptance of Computer Technology : A Comparison of Two Theoretical Models," Management Science, Vol.35, No.8, pp. 982-1003, 1989.

[5] F. Detienne, "Design Strategies and Knowledge in Object-Oriented Programming : Effects of Experience," Human-Computer Interaction, Vol.10, pp.129-169, 1995.

[6] L. Garceau, E. Jancura, & J. Kneiss, "Object-Oriented Analysis and Design : A New Approach to Systems Development," Journal of Systems Management, Vol.44, pp.25-32, 1993.

[7] T. Hill, N. D. Smith, & M. F. Mann, "Role of Efficacy Expectations in Predicting the Decision to Use Advanced Technologies : The Case of Computers," Journal of Applied Psychology, Vol.72, No.2, pp.307-313, 1987.

[8] J. Martin, "Information Engineering, Book II : Planning and Analysis," Englewood Cliffs, New Jersey : Prentice Hall, 1990.

[9] K. Mathieson, "Predicting User Intentions : Comparing the Technology Acceptance Model with the Theory of Planned Behavior," Information Systems Research, Vol.2, No.3, pp. 173-191, September, 1991.

[10] G. C. Moore, & I. Benbasat, "Development of an Instrument to Measure the Perceptions of Adopting Technology Innovation," Information Systems Research, Vol.7, No.2, pp. 192-222, September, 1991.

[11] N. Pennington, A. Y. Lee, & B. Rehder, "Cognitive Activities and Levels of Abstraction in Procedural and Object-Oriented Design," Human-Computer Interaction, Vol.10, pp.171-226, 1995.

[12] M. Weiser, & J. Shertz, "A Study of program problem representation in novice programmers," International Journal of Man-Machine Studies, Vol.19, pp.391-398, 1983.

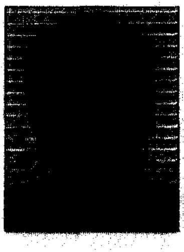
[13] E. Yourdon, "Object-Oriented Observations," Am. Programmer, Vol.2, No.7, pp.3-7, 1989.

[14] Shouhong Wang, "Toward a general model for web-based information systems," International Journal of Information Management, Vol.21, pp.385-396, 2001.

[15] Cho, Ihlsoon and Kim, Young-Gul, "Critical Factors for Assimilation of Object-Oriented Programming Languages," Journal of Management Information Systems, Vol.18, No.3, pp.125-156, 2002.

[16] Agarwal, Ritu, Sinha, Atish P. and Tanniru, Mohan R, "Cognitive Fit in Requirements Modeling : A Study of Object and Process Methodologies," Journal of Management Information Systems, Vol.13, No.2, pp.137-162, 1996.

[17] Sheetz, Steven D., Irwin, Gretchen, Tegarden, David P, "Exploring the Difficulties of Learning Object-Oriented Techniques," Journal of Management Information Systems, Vol.14, No.2, pp.103-132, 1997.



김 인 재

e-mail : ijkim@dgu.ac.kr

1983년 서울대학교 산업공학과(학사)

1985년 KAIST 경영과학과(석사)

1985년~1991년 LG전자 중앙연구소 전산실

1996년 The University of Nebraska-Lincoln 경영정보학(박사)

1997년~1998년 한남대학교 경영대학 경영학과 조교수

1998년~현재 동국대학교 경영대학 정보관리학과 조교수

관심분야 : 소프트웨어공학, 시스템개발방법론, 데이터베이스 응용

정 덕 훈

e-mail : duke@dgu.ac.kr

1986년 University of Georgia 경영과학(학사)

1989년 The George Washington University 경영정보학(석사)

1996년 The George Washington University 정보관리학(박사)

1996년~1997년 The George Washington University 연구교수

1997년~현재 동국대학교 경영대학 정보관리학과 조교수

관심분야 : 정보공학, 통신정책 등