

# 클래스 다이어그램의 레이아웃과 복잡도가 모델 이해도에 미치는 영향

## Effect of Layout and Complexity of Class Diagram on Model Comprehension

김진만\* 권태희\*\* 임좌상\*\*\*  
Jinman, Kim Taehee, Kwun Joasang, Lim

### 요약

UML 다이어그램은 시스템을 모델링 하기 위한 언어로 사실상 표준으로 널리 사용되고 있다. 그 가운데 특히 클래스 다이어그램은 시스템을 클래스와 클래스들간 관계로 시각화한 것으로 순공학적, 역공학적 시스템 모델링 및 구현에 직접적인 영향을 미친다. 많은 연구에서 클래스 다이어그램의 레이아웃과 복잡성이 시스템 모델링의 이해에 미치는 영향을 설명하고 있다. 하지만 연구 결과가 혼재되어 있어 레이아웃 적용의 효과를 판단하는데 어려움이 따른다. 따라서 본 연구에서는 기존 연구에서 제시된 클래스 다이어그램 레이아웃 기준을 토대로 2 (레이아웃) x 2 (복잡성) 실험을 설계, 레이아웃 적용으로 인한 이해도를 측정하였다. 47명의 피험자가 실험에 참여하여 서로 다르게 제시된 레이아웃과 복잡성에 대해 이해하도록 하였다. 그 결과 레이아웃을 적용한 실험 그룹에서 유효한 효과를 나타내었고 뿐만 아니라 복잡성에서도 유효한 결과를 확인할 수 있었다. 하지만 이 둘의 상호작용 측정에 있어서는 유효한 결과를 얻지 못했다.

### ABSTRACT

As a de facto standard for system modeling, UML diagrams have been widely used in the industry. Of these, the class diagram is useful to visualize the classes and their relationships and thus directly influences the forward and reverse processes of system modeling and development. Many earlier studies have attempted to examine as to what impact the layout and complexity of the class diagram would have on the comprehension of system modeling. However, their findings have been equivocal and it is not easy to understand the effects of diagram layout. Accordingly this research relied on the guidelines of diagram layout suggested in the earlier studies and designed a 2 (layout) x 2 (complexity) factorial design to examine their impact on diagram comprehension. 47 subjects were participated in the experiment where class diagrams were provided differently in their layout and complexity. We found that a good layout was significantly more effective than the bad. Furthermore the results were significantly persistent in the complex group than the simple. However, there was no interaction effect between layout and complexity of the diagram.

☞ Keywords: Model Comprehension(모델 이해), Class Diagram Layout(클래스 다이어그램 레이아웃), Class Diagram Complexity(클래스 다이어그램 복잡도)

## 1. 서론

UML(Unified Modeling Language)은 1997년

OMG (Object Management Group)에 의해 고안된 것으로 시스템을 모델링 하기 위해 학계와 업계에서 사실상 표준(de facto)으로 사용되고 있다. UML 다이어그램 가운데 클래스 다이어그램은 전체 시스템의 클래스간 구조와 연관관계를 파악하는데 유용하다[1]. 순공학적으로 클래스 다이어그램은 요구사항을 어떻게 구현할 것인지를 표현하여 시스템 구현에 직접적인 영향을 미친다. 또한 역공학적으로도 클래스 다이어그램은 시스템을 이해하는데 매우 유용하다. 시스템이 모델링 된 클

\* 정 회 원 : 상명대학교 일반대학원 컴퓨터학과과 박사과정  
hansumo81@gmail.com

\*\* 정 회 원 : NHN 제작  
taehee.kwun@nhn.com

\*\*\* 정 회 원 : 상명대학교 디지털미디어학부 교수  
jslim@smu.ac.kr(교신저자)

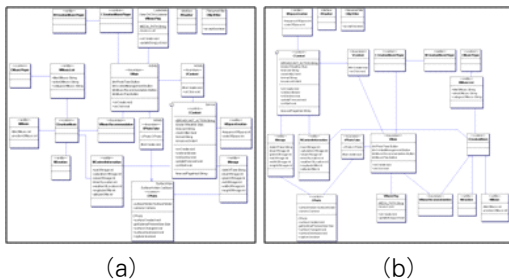
[2010/10/04 투고 - 2010/10/20 심사 - 2011/01/05 심사완료]

래스 다이어그램의 이해 정도에 따라 구현된 소스 코드가 달라질 수 있다. 즉, 클래스 간의 위치, 연결, 패키지와 같은 다이어그램의 레이아웃이 어떻게 배치되고 그려졌는지에 따라 구현할 시스템의 이해가 달라지는 것이다. UML이 출현한 이후 다이어그램의 이해도와 관련된 다양한 연구가 진행되고 있다[2-8]. 이를 크게 나누면 다이어그램 레이아웃이 모델 이해에 미치는 영향[9-14]과 UML 스테레오 타입이 모델 이해에 미치는 영향[8, 15]으로 구분할 수 있다.

많은 연구에서 UML 다이어그램의 레이아웃 기준을 제시하고 있다. 하지만 기준의 선정 근거가 부족하고 혼재해 있기 때문에 시스템 모델링에 있어 실무자는 어떻게 레이아웃을 적용할지 혼란을 겪고 있다[14]. 이를 해결하기 위해 일반화된 클래스 다이어그램 레이아웃 기준이 필요하고, 그 레이아웃은 단순 나열식이 아닌 실무에 적용 가능한 것이어야 한다. 따라서 본 연구는 클래스 다이어그램의 레이아웃과 복잡성이 이해도에 미치는 영향에 대한 것으로 첫 째, 레이아웃의 유용성을 검증하기 위해 레이아웃이 적용된 ‘좋은’ 클래스 다이어그램과 미 적용된 ‘나쁨’ 클래스 다이어그램의 이해도를 실험을 통해 비교 분석했다. 둘째, 다이어그램의 복잡성에 따라 이해도에 미치는 영향이 달라진 결과를 제시한 선행 연구를 토대로 ‘단순’ 클래스 다이어그램과 ‘복잡’ 클래스 다이어그램 각각이 이해도에 미치는 영향에 대해 실험 분석했다. 그 결과 클래스 다이어그램 레이아웃과 복잡성이 이해도에 미치는 영향을 확인 했다.

## 2. 문헌 연구

### 2.1 문제 제기



(그림 1) CASE 도구(Together 2005)에서 동일한 클래스 다이어그램 레이아웃의 변화 (전, 후) ((a) 원래 클래스 다이어그램, (b) 동일한 다이어그램을 다시 불러 온 경우)

UML 도구 (예: Together, ROSE)를 사용하여 클래스 다이어그램을 작성할 시에 시스템 설계자는 자신 혹은 프로젝트에서 정해진 규칙 및 규정을 고려하여 클래스 배치 및 클래스 간 연결과 같은 레이아웃을 적용한다. 시스템 개발자는 이러한 다이어그램을 통해 시스템을 이해하고 구현작업을 수행한다. 이때, 설계자에 의해 모델링 된 다이어그램이 어떤 변경 없이 개발자에게 전달되어야 시스템 개발의 에러를 줄 일 수 있다. 즉, 작성된 다이어그램의 레이아웃이 보전되어야 구현단계에서의 실수를 줄일 수 있는 것이다. 하지만 대부분의 CASE 도구들은 작성된 클래스 다이어그램의 레이아웃 정보를 무시하고 새롭게 클래스들을 배치하고 연결한다. 즉, 설계자에 의해 생성된 UML 다이어그램이 개발자에게는 다르게 보일 수 있는 것이다. 이는 서로 다른 CASE 도구뿐만 아니라 그림 1처럼 같은 도구에서 작성된 다이어그램에서도 동일한 문제가 발생한다. 따라서 다이어그램의 레이아웃이 서로 다름으로 인해 매번 위치를 원래대로 변경하는 수고가 필요하게 되고 특히, 복잡한 다이어그램인 경우에는 레이아웃을 재조정 하는데 많은 인지적, 시간적 노력이 수반된다. 이러한 문제를 해결하기 위해서는 명확한 레이아웃 기준을 CASE 도구에 적용되어야 하며 이는 충분히 공통적이고 심미적인 원리를 포함해야 하고 테스트를 통해 검증되어야 한다.

### 2.2 다이어그램 레이아웃의 영향에 관한 연구

다이어그램 레이아웃의 영향과 관련된 연구는 클래스 다이어그램 레이아웃을 제시하고 이를 적용한 다이어그램의 이해도 효과를 검증하는 실험 연구가 주로 이루어졌다. 하지만 관련연구의 수가 적고 레이아웃 영향의 결과가 연구에 따라 혼재하고 있다.

Purchase[16]는 심미적 레이아웃 기준을 적용한 클래스다이어그램을 사용하여 간단한 질문을 통해 이해도를 측정하였다. 그 결과 다이어그램 내 클래스 간 연관관계 선들의 '교차'가 이해도에 큰 영향을 미쳤고 선의 '직교성'과 '대칭성'은 서로 상충된다는 결과를 보여주었다.

Purchase et al.[10]의 후속 연구에서는 두 번의 실험을 통해 클래스다이어그램의 심미적 레이아웃 기준이 이해도에 미치는 효과를 측정하였다. 첫 번째 실험에서는 다이어그램의 연산 측면에서 레이아웃의 영향을 측정하였으나 사용된 레이아웃 기준 모두 이해도에 영향을 미치지 않는 결과가 나타났다. 두 번째 실험에서는 인지적 측면에서 레이아웃을 적용하여 그 효과를 측정했다. 그 결과 클래스 간 연관선의 '구부러짐'만이 이해도에 효과가 있는 것으로 나타났다. 클래스 배치 분포, 연관선의 길이, 대칭성, 직교성, 정보흐름과 관련된 레이아웃은 유의한 결과를 나타내지 않았다. 즉, 인지적으로 적용된 레이아웃이 연산(기계)적 레이아웃보다 다이어그램의 이해도에 좀 더 영향을 미친다는 것을 알 수 있다.

Hofman[17]은 다이어그램 레이아웃 기준과 함께 실제 시스템설계자를 대상으로 제시된 레이아웃 알고리즘의 사용성 테스트를 수행했다. 그 결과 대부분의 설계자들은 레이아웃 알고리즘이 적용된 다이어그램이 모델 수정 시간을 단축시켜준다고 응답했고 또한 평균적으로 다이어그램의 수정 사항이 매우 적었다. 하지만 제시한 레이아웃 알고리즘에 대한 정성적 평가가 이루어졌으며 클래스간의 상속, 집합 관계는 고려되지 않았기에 평가 결과를 일반화하기 어렵다.

Wong and Sun[14]은 클래스다이어그램과 순차 다이어그램 관련 레이아웃을 지각적 조직화(Perceptual Organization)와 지각적 분리화(Perceptual Segregation)로 분류하여 제시하였다. 하지만 이 연구는 레이아웃 기준의 분류에 있어 이론적 근거가 부족하여 객관성이 떨어진다. 또한 레이아웃 기준 적용에 대한 구체적인 지침이 없기

때문에 실무에 이를 적용하기에는 한계가 있다.

Lemon et al.[18]은 Gestalt 이론 중 사물 인식에 영향을 미치는 현상을 세 가지 (유사성 (Similarity), 근접성 (Proximity), 연속성 (Continuity))로 제한하여 레이아웃의 기준을 설정하였다. 기존의 레이아웃 연구는 주로 그래픽 이론에 근거한 반면, 이 연구는 심리학 관점에서 UML 다이어그램의 레이아웃 기준을 제시했다는 점에서 차별성을 가진다.

Eichelberger[19]는 기존의 레이아웃 연구가 단순히 기준 나열에 불과한데 반해 레이아웃 기준의 단계적 적용을 알고리즘으로 나타냈고 프로그래밍 언어로 구현한 점에서 다른 연구와 차별성을 가진다. 하지만 제시된 레이아웃의 실증적 검증이 부족하여 때문에 연구 결과를 일반화하기 어렵다.

본 연구는 선행 연구를 통해 수집된 레이아웃 기준을 조합하여 클래스다이어그램에 적용할 수 있는 레이아웃 기준을 사용했다. 레이아웃은 총 4개 단계의 절차를 거쳐 정의하였다. 1) 문헌 연구에 제시된 레이아웃 기준을 수집하였고, 2) 수집된 기준의 의미가 유사한 것끼리 집단화 하였다. 3) 2건 이상의 선행연구에서 공통적으로 채택된 기준을 식별 하였으며, 4) 레이아웃 이외의 부가 정보 (예: 라벨, 주석, 추상적 기준을 제거하였다. 추출된 레이아웃 기준은 부록 1에 나타내었다.

이와 같이 다이어그램의 레이아웃과 관련된 효과를 측정하는 연구들이 혼재해 있다. 특히 레이아웃이 다이어그램 이해도에 영향을 미치는지 그렇지 않은지를 쉽게 판단하기에는 어려움이 있다. 따라서 본 연구에서는 레이아웃이 다이어그램의 이해에 효과가 있는지를 검증하기 위해 다음과 같은 가설을 설정했다.

**H01:** 레이아웃에 따라 다이어그램 이해도에 차이가 없을 것이다.

레이아웃<sub>총음</sub> = 레이아웃<sub>나쁨</sub>

### 2.3 다이어그램 복잡성의 영향에 관한 연구

다이어그램 레이아웃이 이해도에 미치는 영향에 대한 선행연구 중 다이어그램의 복잡성이 이해도에 영향을 미친다는 결과를 보여준 연구가 있다[16, 18]. 이는 다이어그램의 이해도가 레이아웃뿐만 아니라 복잡성에 의해서도 영향을 받는다는 것을 시사하고 있다.

Purchase[16]의 연구에서는 다이어그램 내 클래스나 연관관계 선이 많을수록 레이아웃이 이해도에 영향을 미치는 것으로 나타났다. 하지만 간단한 다이어그램에 대해서는 그 반대의 결과가 나타났다. 이 연구에서는 클래스다이어그램 자체에 대한 문법적 정보를 묻는 간단한 문제를 제시하여 이해도를 측정하였다. 이는 가장 낮은 수준의 이해도 측정으로 본 연구에서는 문법적 정보와 함께 의미적 정보의 이해를 측정하기 위해 시나리오 기반의 질문을 함께 적용했다.

Gestalt 이론을 적용한 Lemon et al.[18]의 연구에서는 사물 인식에 영향을 미치는 현상을 배경으로 선정된 레이아웃을 적용하였고, 그 결과 복잡한 다이어그램에 대해서만이 레이아웃 적용이 효과적이었다. 이는 Purchase[16]의 연구 결과와 유사하게 복잡한 다이어그램에서의 레이아웃 기준 적용이 효과적이라는 결과를 보여주고 있다.

이처럼 다이어그램의 이해도에 영향을 미치는 요인이 레이아웃 외에 복잡성이라는 추가 요인이 있음을 두 연구의 결과를 통해 알 수 있다. 따라서 본 연구에서는 다이어그램의 복잡성과 관련해 이해도의 효과를 검증하기 위해 다음과 같은 가설을 설정했다.

**H0c:** 다이어그램 복잡성에 따라 이해도에 차이가 없을 것이다.

$$\text{복잡도} \sim \text{단순} = \text{복잡도} \sim \text{복잡}$$

또한, 다이어그램의 레이아웃과 복잡성의 상호작용 효과로 인한 이해도의 차이가 있는지를 알아보기 위해 다음과 같은 가설을 추가로 설정했다.

**H0i:** 다이어그램 복잡성에 따라 레이아웃 적용 효과에 차이가 없을 것이다.

$$\text{이해도} \sim \text{레이아웃} = \text{이해도} \sim \text{복잡성}$$

## 3. 연구 방법

### 3.1 실험설계 및 사례

본 연구는 현장에서 적절히 통제하기 쉽지 않기 때문에 통제실험방법을 채택하였다. 본 실험에서는 레이아웃과 다이어그램의 복잡성과의 상관관계에 따른 가설 [16, 18]을 검증하기 위해 표 1 처럼 2 (레이아웃) X 2 (복잡성) 연구로 설계되었다. 레이아웃은 ‘좋은’/‘나쁜’으로, 복잡성은 사례에 포함된 정보의 양에 따라 ‘복잡’/‘단순’으로 통제되었다.

(표 1) 2 x 2 실험 설계

레이아웃 \ 복잡도	복잡 (CPLX: Complex)	단순 (SMPL: Simple)
좋은 (CL: Cognitive Layout)	조건 1	조건 2
나쁜 (NCL: Non-Cognitive Layout)	조건 3	조건 4

(표 2) 실험 그룹의 사례 배분

실험그룹	하위그룹	피험자수	실험작업
A	A-1	24	조건 1 (IFIP_CL-CPLX)
	A-2	24	조건 4 (학사_NCL-SMPL)
B	B-1	23	조건 3 (IFIP_NCL-CPLX)
	B-2	23	조건 2 (학사_CL-SMPL)

실험에는 소프트웨어 공학과목을 수강하는 대학생 47명이 참여했다. 모든 피험자는 객체지향, 요구분석, 클래스다이어그램에 대한 내용을 숙지하고 있으며, 본 실험을 수행하는데 필요한 지식

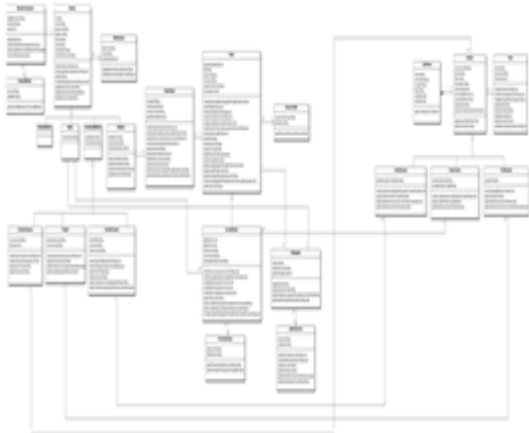
이 충분하다고 판단했다[20]. 따라서 표 2처럼 실험에 앞서 모든 피험자들은 예비 뽑기를 통해 4개의 그룹에 무작위로 할당하였다.

본 연구에서의 복잡성은 사례를 통하여 통제되었다. 동일한 사례에 대하여 복잡성을 조정하면 동일 사례를 2회 수행하게 되어 학습효과로 인해 실험목적을 달성할 수 없으므로 각기 다른 2개의 사례(복잡, 단순)를 설정하였다. 첫 번째 사례는 Kabeli and Shoval[21]와 Shoval et al.[22]의 연구에 사용됐던 IFIP (International Federation for Information Processing)로서 그림 2, 4에 보이는 것과 같이 복잡한 사례로 설정했고, 두 번째 사례는

그림 3, 5에 보이는 것과 같이 학사업무를 설계한 단순한 사례로 설정했다(Purchase et al.[11, 12]과 Gutwenger et al.[23]의 연구에 사용됐던 ‘학사업무’).

### 3.2 실험 변수

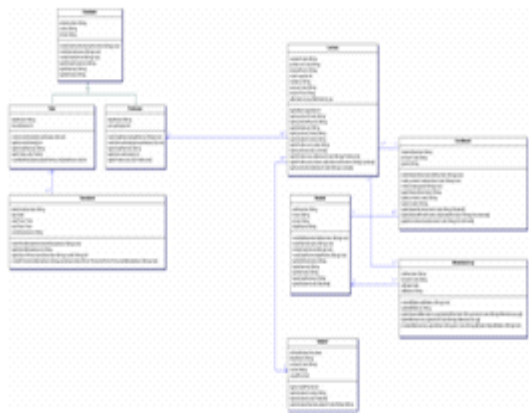
본 실험의 독립변수는 클래스 다이어그램의 ‘레이아웃’과 ‘복잡성’이다. 첫 번째, 레이아웃은 ‘좋은’과 ‘나쁨’ 즉 레이아웃 기준의 적용과 미 적용 2개의 수준으로 구성했다. 두 번째, 복잡성은 ‘복잡’, ‘단순’으로 구분되었다. 복잡성 측정은 Sheldon and Hong[24]의 클래스 다이어그램 복잡성 (CCD: complexity of class diagram) 매트릭, 클래스



(그림 2) IFIP\_CL-CPLX 클래스 다이어그램



(그림 4) IFIP\_NCL-CPLX 클래스 다이어그램



(그림 3) 학사\_CL-SMPL 클래스 다이어그램



(그림 5) 학사\_NCL-SMPL 클래스 다이어그램

(표 3) 각 사례의 클래스 복잡도 평가

사례 \ 비교	CCD	클래스 커플링	클래스 요소
IFIP	31.5	22	클래스 수: 24개 속성 수: 88개 메소드 수: 103개
학사업무	11.7	10	클래스 수: 9개 속성 수: 42개 메소드 수: 58개

커플링, 클래스 다이어그램 요소 수 (클래스/속성/메소드 수) 3가지 관점에서 측정했다. Sheldon and Hong[24]은 CCD가 30을 넘으면 복잡한 다이어그램이고 30 미만은 비교적 단순한 다이어그램이라고 정의했다. 표 3은 본 실험에 사용된 IFIP와 학사업무 사례에 대한 클래스 다이어그램 복잡성을 평가한 것이다. 세 가지 관점에서 다이어그램을 비교한 결과 IFIP 사례가 학사업무 사례보다 충분히 복잡한 것을 알 수 있다.

중속변수는 다이어그램 이해도와 관련된 선행 연구에서 일반적으로 사용된 다이어그램에 대한 이해점수를 사용했다[9-12, 14, 18, 19, 23, 25-30]. 이해도는 프로그램 및 모델 이해도를 측정할 때 주로 사용되고 있는 질문방식을 통해 측정했다 [30-33]. 이해도를 묻는 문항은 크게 정적 정보와 동적 정보로 구성된다. 정적 정보를 묻는 문항은 클래스 다이어그램에 대한 지식을 묻는 문항으로 Bloom[34]연구에서의 인지 텍사노미 레벨 중 가장 낮은 레벨인 ‘Knowledge’ 영역에 해당한다. 이는 단지 학습했던 내용을 기억하여 질문에 답하게 하는 낮은 수준의 이해도 측정이다. 동적 정보를 묻는 문항은 주어진 클래스 다이어그램이 어떤 업무 및 기능을 수행하는지를 이해해야만 풀 수 있는 난이도를 가진 문항으로 Bloom[34]의 인지 텍사노미 레벨의 2번째 레벨인 ‘Comprehension’에 해당한다. 이는 피험자로 하여금 학습했던 클래스 다이어그램의 지식을 사용하여 클래스 간의 관계 정보, 클래스의 행위 정보를 분석하게 만든다. 이렇게 클래스 다이어그램에 대해 정적 정보와 함께 동적 정보를 분석하게 함으로써 레이아웃의 적용 효과를 보다 구체적으로 살펴볼 수 있었다.

문항은 각 다이어그램 당 30문항이며 정적 문항 15개, 동적 문항 15개로 이루어져 있다

### 3.3 실험절차와 환경

실험은 소프트웨어공학 과목의 중간고사 형식으로 진행되었고 다이어그램의 이해를 포함한 질문 답변에 총 1시간을 부여하여 가외변수로서 시간을 통제하였다. 실험에 앞서 피험자에게 본 실험의



(그림 6) 실험 화면

목적인 클래스 다이어그램 이해도 평가에 대해 설명했으며, 실험 시간, 실험 방법 등에 대해 설명했다. Purchase[16], Purchase et al.[10]의 연구에서 처럼 실제 문제와 유사한 형태의 예비 문제를 풀게 하여 컴퓨터 사용, 실험 시스템 등 실험 환경에 대해 익숙해지도록 유도하였다. 모니터 화면 (17인치, 1280\*1024 해상도)에는 IFIP 또는 학사업무 사례에 대한 이해도를 묻는 질문이 표시되며 피험자는 키보드와 마우스를 사용하여 응답을 작성하였다. 객관식 문항은 라디오 버튼의 응답으로, 주관식 문항은 텍스트 입력 창을 통해 응답을 받았다. 예비 문제가 끝난 후에는 실험에 사용된 사례(IFIP, 학사업무)에 대해 구두 설명을 했고 간략한 사례 설명서와 IFIP 클래스 다이어그램 1장, 학사업무 클래스 다이어그램 1장씩 총 2장의 클래스 다이어그램이 출력물로 주어졌다. 사례에 대한

설명이 끝난 후 피험자는 실험 시스템에 자신의 학번을 입력함으로써 시스템에 접속하여 문제를 풀어나갔다. 그림 6처럼 화면에는 특정 사례에 대한 이해도를 묻는 문항 30개가 표시되며 모든 문항에 대한 응답을 작성하고 ‘확인’버튼을 누르면 다른 사례에 대한 퀴즈 화면이 표시된다. 표 4는 실험에 사용된 문항의 일부를 나타내었다.

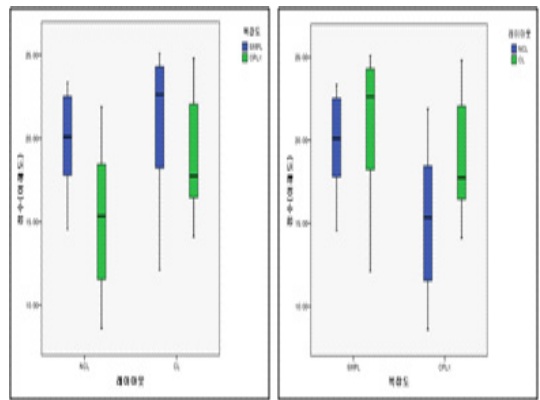
(표 4) 이해도 측정 질문

IFIP 이해도 측정 질문
Q1. Reviewer는 최소 2개의 Paper를 검토해야 한다.
Q3. 논문은 반드시 1명의 저자에 의해 작성되어야 한다.
Q22. 컨퍼런스에 등록된 튜토리얼을 조회한 후 특정 튜토리얼에 대한 발표자 정보(소속, 연구분야, 튜토리얼 요약)을 조회하려 한다. 어떤 클래스가 사용되는가?
Q27. "세미나"라는 새로운 세션을 만들려고 한다. 어떤 클래스를 상속받으면 되는가?
Q30. 현행 선임리뷰어 카테고리는 "선임인지 아닌지"만 입력하게 되어있다. 하지만 리뷰어구분을 보다 표준화하기 위해 1=일반 2=선임 처럼 정수 코드값을 입력받도록 변경하려 한다. 어떤 클래스의 어떤 속성이 수정되어야 하는가?
학사업무 이해도 측정 질문
Q2. 동일한 과목에 대해 2개 이상의 강좌가 열릴수 있다.
Q4. 모든 교수는 최소 1개 이상의 강의를 해야 한다.
Q24. 교수는 특정 강의에 대해 수강생 성적을 입력하려 한다. 어떤 클래스가 사용되는가?
Q27. 개설된 강의가 현재 수강인원 초과상태인지 확인하려 한다. 어떤 클래스에 이를 구현하는 메소드가 추가되어야 하는가?
Q29. 새로운 수강신청 시 기존 신청되어 있는 강의와 중복되는지를 확인하려 한다. 어떤 클래스에 이를 구현하는 메소드가 추가되어야 하는가?

실험의 이해도 측정을 위해 웹 시스템을 구축하였으며 질문화면은 Internet Explorer 6.0 브라우

저에 최적화 하여 설계됐다. 개발 환경은 Apache Tomcat 5.5.25 WAS와 MySQL 5.0.51b DBMS로 서버를 구축했으며 개발 언어로는 JDK1.6, JSP2.0, Html, Javascript를 사용했다. 피험자의 이상행동으로 인한 시간측정 손실을 방지하기 위해 웹 페이지 내에서 ‘뒤로 가기’를 막아냈으며, 모든 응답 데이터는 DB에 저장되도록 하였다.

#### 4. 실험 결과와 분석



(그림 7) 상자도표 ((a) 레이아웃 (b) 복잡성)

(표 5) 레이아웃과 복잡성에 따른 이해도에 대한 기술통계

		CPLX	SMPL	전체
CL	평균 (표준편차)	18.72 (3.47)	20.89 (4.17)	19.80 (3.82)
	피험자 수	13	12	25
NCL	평균 (표준편차)	15.15 (4.38)	19.89 (2.93)	17.52 (3.66)
	피험자 수	11	11	22
전체	평균 (표준편차)	16.93 (3.92)	20.39 (3.55)	18.66 (3.74)
	피험자 수	24	23	47

데이터 분석에 앞서 이상 값이 있는지 표준편차를 계산하였다. 그림 7처럼 상자도표를 통해 실



험 데이터의 이상 값의 유무를 확인했다. 그 결과 이상 값없이 총 47건의 실험 데이터를 분석에 사용했다.

복잡성을 실험한 집단 간, 레이아웃을 실험한 집단 간의 피험자 수와 이해도의 평균, 표준편차는 표 5와 같다.

레이아웃, 복잡성에 따른 종속변수(이해도)의 차이를 분석하기 위해 이원분산분석(Two-Way ANOVA)을 실시하였다. 복잡성, 레이아웃의 주 효과와 레이아웃 x 복잡성 상호작용효과에 대한 통계적 유의성을 검정한 분산분석 결과는 표 6과 같다.

(표 6) 복잡성과 레이아웃에 따른 이해도에 대한 분산분석 결과

		df	F	Sig.
주효과	복잡성 (A)	1	9.81	.003**
	레이아웃 (B)	1	4.28	.045*
상호작용 효과	A x B	1	1.35	.252

\*\* p<.01 \* p<.05

#### 4.1 H0I: 레이아웃 검증

H0I는 ‘레이아웃에 따라 다이어그램 이해도에 차이가 없을 것이다’라는 가설이다. 레이아웃이 다이어그램 이해도에 영향을 주는지에 대한 검정 결과 유의수준 .05에서 레이아웃의 적용 유무에 따라 집단간 유의한 차이가 있는 것으로 분석되었다 [F(1, 46)=4.28, p=.045]. 표 5를 보면, 클래스 다이어그램의 레이아웃 수준에 따른 이해점수의 평균은 CL 조건이 NCL 조건보다 높은 것으로 나타났다 (19.80 > 17.52). 이는 레이아웃의 기준을 적용한 다이어그램이 미 적용 한 다이어그램보다 이해도가 높은 결과를 나타내고 있다. 그러므로 영가설은 기각하며, ‘유의 수준 .05에서 레이아웃 적용은 다이어그램의 이해도에 효과적이다’라고 결론 내린다.

#### 4.2 H0c: 복잡성 검증

H0c는 ‘다이어그램 복잡성에 따라 이해도에 차이가 없을 것이다’라는 가설이다. 이해도에 대한 복잡성의 효과를 분석한 결과, 유의수준 .01에서 복잡성에 따라 다이어그램의 이해도에 차이가 있는 것으로 나타나 [F(1, 46)=9.81, p=.003], 영가설은 기각한다. 표 5를 살펴보면, 클래스다이어그램의 복잡성 수준에 따른 이해점수의 평균은 CPLX 조건이 SMPL 조건보다 낮은 것으로 나타났다 (16.93 < 20.39). 이는 일반적으로 복잡한 다이어그램이 단순한 다이어그램에 비해 이해도가 낮은 것을 나타낸다. 그러므로 ‘유의수준 .01에서 복잡성은 다이어그램 이해도에 영향을 미친다’라고 결론 내린다.

#### 4.3 H0i: 복잡성과 레이아웃 상호작용 검증

H0i는 ‘다이어그램 복잡성에 따라 레이아웃 적용 효과에 차이가 없을 것이다’라는 가설이다. 다이어그램의 복잡성과 레이아웃 간의 상호작용에 대한 분산분석결과 유의 수준 .05에서 상호작용에 따른 유의한 차이가 없는 것으로 분석되어 [F(1, 46)=1.35, p=.252], 영가설은 기각하지 않는다. 따라서 ‘유의수준 .05에서 레이아웃과 복잡성의 상호 작용은 없다’고 결론짓는다. 또한 복잡한 다이어그램에서 레이아웃 적용 시 이해점수 평균이 미 적용 시 이해점수 평균보다 높게 나타났고 (18.72 > 15.15), 단순한 다이어그램에서는 이해점수의 차이가 크지 않지만 레이아웃 적용 시에 이해점수가 좀더 크게 나타났다 (20.89 > 19.89).

### 5. 결 론

본 연구에서는 많은 다이어그램 레이아웃 기준으로 인한 실무자의 혼란[14]을 해결하고자 레이아웃과 관련된 문헌연구를 통해 레이아웃 기준을 수집하여 공통적으로 사용된 레이아웃을 식별하여 레이아웃 적용 효과를 실험을 통해 검증하였



다. 그 결과, 레이아웃은 다이어그램의 이해에 영향을 미치는 것으로 나타났다. 이는 Purchase et al.[10]의 두 번째 실험 결과와 같이 인지적 레이아웃이 적용된 다이어그램이 이해하기 쉬운 것을 나타낸다. 그러나 본 연구에서 복잡성이 다이어그램 이해도에 미치는 영향을 검증한 결과, 레이아웃의 효과가 복잡한 다이어그램에서만 나타난다는 Lemon et al.[18]과 Purchase[16]의 연구 결과와 달리 복잡성과 레이아웃의 상호작용이 없는 것으로 나타났다. 즉, 다이어그램의 복잡성에 상관없이 레이아웃 적용의 효과가 나타났다. 달리 말하면, 다이어그램의 복잡, 단순에 상관없이 레이아웃의 적용만으로 다이어그램의 이해도 향상에 긍정적인 효과를 가져올 수 있다는 결론을 내릴 수 있다. 본 연구에서는 복잡성, 레이아웃을 독립변수로 다이어그램의 이해도를 측정하기 위해 학생을 피험자로 사용하였다. 따라서 본 연구는 피험자 선정의 문제와 실험 특성상 복잡성을 통제하기 위한 사례 선별에 한계가 있어 이 결과를 일반화하는데 주의가 필요하다. 특히 본 연구가 학생을 대상으로 이루어져서 그 결과를 실무에 적용하기에는 주의가 필요할 수 있다. 그러나 본 연구의 피험자는 UML을 사용하여 소프트웨어를 설계한 경험이 있는 고학년이라는 점에서 어느 정도 그 결과를 실무에 예측할 수 있겠다. 또한, 본 연구의 실험에서 사용한 클래스 다이어그램 사례를 이해하기 위해 많은 경험이 필요치 않기 때문에 학생을 피험자로 하는데 문제가 없다고 판단된다[20].

본 연구는 레이아웃이 클래스 다이어그램의 이해에 미치는 영향에 대한 연구방법으로 통제 실험을 선택했다. 따라서 본 연구에서 얻은 결과를 일반화 하기 위해서는 소프트웨어 개발 현장에서의 평가가 이루어져야 할 것이다. 또한 응답시간을 함께 측정함으로써 시간 측면에서의 복잡도에 따른 레이아웃 적용 효과 차이를 검증할 필요가 있다. 끝으로 본 연구에서는 클래스, 연관관계 선을 중심으로 레이아웃의 효과를 평가하였지만 이

외에 색, 관계역할, 글꼴 등 클래스 다이어그램 이해에 영향을 주는 요소를 고려한 후속 연구가 이루어져야 할 것이다.

## 참 고 문 헌

- [1] V. Rajlich, "Program comprehension as a learning process", IEEE Computer Society, pp.343-350, 2002.
- [2] L.C. Briand, "Software documentation: how much is enough?", Published by the IEEE Computer Society, 2003.
- [3] S. Tilley and S. Huang, "A qualitative assessment of the efficacy of UML diagrams as a form of graphical documentation in aiding program understanding", ACM, 2003.
- [4] L.C. Briand, Y. Labiche, M. Di Penta and H.D. Yan-Bondoc, "An experimental investigation of formality in UML-based development", IEEE Trans. Software Eng., Published by the IEEE Computer Society, pp.833-849, 2005.
- [5] I. Reinhartz-Berger and D. Dori, "OPM vs. UML--Experimenting with Comprehension and Construction of Web Application Models", Empirical Software Engineering, Springer, vol.10, No.1, pp.57-80, 2005.
- [6] M.C. Otero and J.J. Dolado, "An empirical comparison of the dynamic modeling in OML and UML", J. Syst. Software, Elsevier, vol.77, No.2, pp.91-102, 2005.
- [7] B. Anda, K. Hansen, I. Gullesen and H.K. Thorsen, "Experiences from introducing UML-based development in a large safety-critical project", Empirical Software Engineering, Springer, vol.11, No.4, pp.555-581, 2006.
- [8] L. Kuzniarz, M. Staron and C. Wohlin, "An empirical study on using stereotypes to improve

- understanding of UML models", Published by the IEEE Computer Society, 2004.
- [9] O. Andriyevska, N. Dragan, B. Simoes and J.I. Maletic, "Evaluating UML class diagram layout based on architectural importance", IEEE, 2005.
- [10] H.C. Purchase, M. McGill, L. Colpoys and D. Carrington, "Graph drawing aesthetics and the comprehension of UML class diagrams: an empirical study", Australian Computer Society, Inc., 2001.
- [11] H.C. Purchase, J.A. Alder and D.A. Carrington, "Graph layout aesthetics in UML diagrams: user preferences", *J.Graph Algorithms Appl.*, vol.6, No.3, pp.255-279, 2002.
- [12] H.C. Purchase, D. Carrington and J.A. Alder, "Empirical evaluation of aesthetics-based graph layout", *Empirical Software Engineering*, Springer, vol.7, No.3, pp.233-255, 2002.
- [13] M.K. Coleman and D.S. Parker, "Aesthetics-based graph layout for human consumption", *Software: Practice and Experience*, John Wiley & Sons, vol.26, No.12, pp.1415-1438, 1996.
- [14] K. Wong and D. Sun, "On evaluating the layout of UML diagrams for program comprehension", *Software Quality Journal*, Springer, vol.14, No.3, pp.233-259, 2006.
- [15] F. Ricca, M. Di Penta, M. Torchiano, P. Tonella and M. Ceccato, "How design notations affect the comprehension of Web applications", *Journal of Software Maintenance and Evolution: Research and Practice*, John Wiley & Sons, vol.19, No.5, pp.339-359, 2007.
- [16] H. Purchase, "Which aesthetic has the greatest effect on human understanding?", Springer, pp.248-261, 1997.
- [17] P. Hofman and M. Piasecki, "Automatic improvement of UML diagrams layout", pp.89-98, 2006.
- [18] K. Lemon, E.B. Allen, J.C. Carver and G.L. Bradshaw, "An empirical study of the effects of gestalt principles on diagram understandability", IEEE Computer Society, 2007.
- [19] H. Eichelberger, "Aesthetics of class diagrams", pp.23-31, 2002.
- [20] V.R. Basili, F. Shull and F. Lanubile, "Building knowledge through families of experiments", *IEEE Trans. Software Eng.*, vol.25, No.4, pp.456-473, 1999.
- [21] J. Kabeli and P. Shoval, "Comprehension and quality of analysis specifications--a comparison of FOOM and OPM methodologies", *Information and Software Technology*, Elsevier, vol.47, No.4, pp.271-290, 2005.
- [22] P. Shoval, A. Yampolsky and M. Last, "Class Diagrams and Use Cases-Experimental Examination of the Preferred Order of Modeling", Citeseer, pp.453-472, 2006.
- [23] C. Gutwenger, M. Jünger, K. Klein, J. Kupke, S. Leipert and P. Mutzel, "A new approach for visualizing UML class diagrams", *ACM*, pp.179-188, 2003.
- [24] F.T. Sheldon and H. Chung, "Measuring the complexity of class diagrams in reverse engineering", *Journal of Software Maintenance and Evolution: Research and Practice*, John Wiley & Sons, vol.18, No.5, pp.333-350, 2006.
- [25] H. Eichelberger and J.W. von Gudenberg, "UML class diagrams-State of the art in layout techniques", Citeseer, pp.30 - 34, 2003.
- [26] W. Lai, "Layout adjustment and boundary detection for a diagram", Published by the IEEE Computer Society, 2001.
- [27] J. Seemann, "Extending the sugiyama algorithm for drawing UML class diagrams: Towards automatic layout of object-oriented software

- diagrams", Springer, pp.415-424, 1997.
- [28] M. Eiglsperger, M. Kaufmann and M. Siebenhaller, "A topology-shape-metrics approach for the automatic layout of UML class diagrams", ACM, pp.189, 2003.
- [29] C. Hagen, "Similarity of spatial configurations in interactive layout", pp.67 - 72, 2005.
- [30] S. Yusuf, H. Kagdi and J.I. Maletic, "Assessing the comprehension of UML class diagrams via eye tracking", IEEE Computer Society, 2007.
- [31] P. Shoval and I. Frumermann, "OO and EER conceptual schemas: A comparison of user comprehension", J. Database Manage., Idea Group Inc, 701 E. Chocolate Ave, Suite 200, Hershey, PA, 17033-1240, USA, vol.5, No.4, pp.28-48, 1994.
- [32] M. Peleg and D. Dori, "The model multiplicity problem: experimenting with real-time specification methods", IEEE Trans. Software Eng., vol.26, No.8, pp.742-759, 2000.
- [33] Y.G. Guéhéneuc, "TAUPE: towards understanding program comprehension", ACM, 2006.
- [34] B.S. Bloom, M.D. Engelhart, E. Frust, N. Hill and D. Krathwohl, "Taxonomy of educational objectives: The classification of educational goals. Cognitive domain", Longman, 1956.

〈부록 1〉 레이아웃 기준과 선행연구와의 관계

레이아웃 기준	선행 연구
상속, 집합 관계의 선은 하나로 결합한다.	Wong and Sun[14], Eichelberger[19], Purchase et al.[11]
같은 그룹의 클래스끼리는 같은 크기, 군집하여 그린다.	Wong and Sun[14], Eichelberger[19], Coleman and Stott-Parker[13], Lemon et al.[18]
선의 교차를 최소화한다.	Wong and Sun[14], Eichelberger[19], Purchase et al.[11], Purchase[16], Hofman[17], Coleman and Stott-Parker[13], Lemon et al.[18]
선의 구부러짐을 최소화한다.	Wong and Sun[14], Eichelberger[19], Purchase et al.[11], Purchase[16], Purchase et al.[10]
연관된 클래스끼리는 가깝게 배치한다.	Wong and Sun[14], Eichelberger[19]
부모클래스는 자식클래스에 가까운 중앙에 배치한다.	Wong and Sun[14], Eichelberger[19], Coleman and Stott-Parker[13], Lemon et al.[18]
다이어그램 요소 간 중첩을 피한다.	Wong and Sun[14], Eichelberger[19]
부모클래스는 자식클래스 위쪽에 배치한다.	Wong and Sun[14], Coleman and Stott-Parker[13]
선의 구부러짐은 직각만 허용한다.	Wong and Sun[14], Purchase et al.[11], Purchase[16], Purchase et al.[10]
리딩흐름을 고려하여 정보가 위에서 아래로, 좌측에서 우측으로 흐르게 배치한다 (또는 일정한 방향으로 정보가 흐르게 배치한다).	Wong and Sun[14], Purchase et al.[10]
계층관계가 있는 구조는 구분 짓는다.	Eichelberger[19], Coleman and Stott-Parker[13]
레이아웃의 너비가 너무 넓지 않게 한다.	Purchase et al.[11], Coleman and Stott-Parker[13]
노드에 연결된 선 간의 간격 (angle)을 최대화 한다.	Eichelberger[19], Purchase[16], Coleman and Stott-Parker[13]

## ◎ 저 자 소개 ◎



### 김진만

2003년 한서대학교 수학과 졸업(학사)  
2006년 상명대학교 디지털미디어대학원 정보통신학과 졸업(석사)  
2008~현재 상명대학교 일반대학원 컴퓨터과학과 박사과정  
관심분야 : 소프트웨어공학, MDA, 모바일, 증강현실, 감성콘텐츠  
E-mail : hansumo81@gmail.com



### 권태희

2007년 상명대학교 소프트웨어학부 졸업(학사)  
2009년 상명대학교 일반대학원 컴퓨터과학과 졸업(석사)  
2010~현재 NHN 재직  
관심분야 : 소프트웨어공학, 지도서비스  
E-mail : tahee.kwun@nhn.com



### 임좌상

1991년 New South Wales University MIS전공 졸업(석사)  
1994년 New South Wales University MIS전공 졸업(박사)  
1997~현재 상명대학교 디지털미디어학부 교수  
관심분야 : MIS, 소프트웨어공학, 모바일웹, 감성콘텐츠  
E-mail : jslim@smu.ac.kr