

클래스 도출을 지원하는 세가지 완충 다이어그램

Buffer zone: Three Diagrams for Assisting Class Extraction

이서정(Seojeong Lee)*

초 록

객체지향 및 컴포넌트지향 소프트웨어 개발 방법론의 클래스의 도출은 개발하려는 소프트웨어의 전체 품질을 결정하는데 중요한 요소이다. 각 방법론에서는 이 부분에 대해 클래스의 정의와 도출하는 방법을 설명하고 있지만 초보 분석자들이 활용하기는 어려운 점이 많다. 그 중, 개념적 갭(conceptual gap)으로 인한 문제는 클래스 도출에 가장 큰 영향을 미친다. 연구에서는 이런 부분을 보완하기 위한 장치로써 세 가지 다이어그램을 제안한다. 요구분석과정의 결과를 도식적으로 정리하여 클래스 도출을 보완하기 위한 도구로 정보다이어그램(Information Diagram), 행위다이어그램(Behavior Diagram) 그리고 관리다이어그램(Management Diagram)이다. 세 다이어그램은 시스템의 관련자(stakeholder)가 시스템에 어떤 서비스를 요구하는가에 대한 관점으로 필요한 정보를 찾고, 그에 포함될 수 있는 세부 행위를 분석하고, 누가 이 서비스에 대한 책임이 있는가의 명세이다. 본 연구에서 제안하는 다이어그램들은 기존의 소프트웨어 개발 방법에서 클래스를 도출하는 과정에 활용할 수 있다.

ABSTRACT

Class extraction of object or component based software development methodology is the major factor for software quality. Each method has the class definition and extraction method however there are some troubles when the beginners try. Especially, the conceptual gap results to make the class extraction hard. This research suggests three diagrams to support it. They are Information diagram, Behavior diagram and Management diagram. They specify which services a stakeholder wants, which information to support the service, which actions to solve the service and who has the responsibility for those. Any analysis process which takes class extraction can utilize these diagrams

키워드 : 소프트웨어 개발방법, 클래스 도출, 정보다이어그램, 행위다이어그램, 관리다이어그램
Software development methodology, class extraction, Information diagram,
Behavior diagram, Management diagram

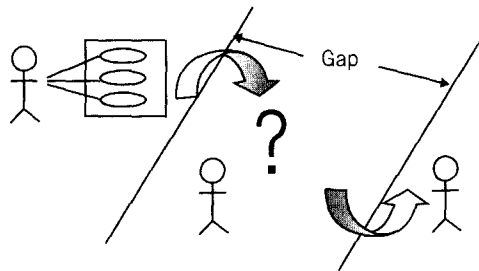
* 숙명여자대학교 연구원

1. 서론

객체 지향이나 컴포넌트 기반 기술 개념이 도입되면서 이를 지원하는 많은 방법론들이 개발되었고, 이들 중 국내외적으로 많이 사용되는 방법론으로는 Rational사의 RUP (Rational Unified Process), Computer Associates사의 CBD96, Compuware사의 UNIFACE, PrincetonSofttech사의 Select, 한국전자통신연구원에서 개발한 마르미(MaRMI : Magic and Robust Methodology Integrated) 등이 있다. 이들 방법론들은 요구사항을 명세한 시나리오를 분석하여 객체나 컴포넌트를 도출한다. 시나리오 중 동사 또는 동사형을 찾아 유즈케이스 명사 또는 명사형을 찾아 후보 클래스를 찾은 후, 선별하는 과정을 거친다.[7] 하지만, 시나리오가 완전하지 않거나, 분석가가 도메인에 익숙하지 않을 경우 클래스 도출은 완전하기 어렵다. 예를 들어, RUP의 경우 유즈케이스 다이어그램(usecase diagram)은 시스템은 사용자에게 어떤 서비스를 요구하는가를 포괄적으로 설명한 그림이고, 이에 대한 정적(static) 정보는 클래스 다이어그램에 명세하고 동적(dynamic) 정보는 시퀀스 다이어그램(sequence diagram), 액티비티 다이어그램

(activity diagram) 혹은 협력 다이어그램(collaboration diagram)에 표현한다. 이들 사이에는 표현의 상세수준의 차이가 커 기존 시스템의 클래스를 활용하거나 경험이 많지 않은 경우, 필요한 클래스를 완전히 분석하거나 전체 시스템을 여러 부분으로 나누어 분석하는 경우 클래스의 크기나 상세수준을 일정하게 유지하기는 어렵다. 이런 차이를 개념적 갭(conceptual gap)이라 하며 <그림 1>과 같이 표현할 수 있다.[8][4] 개념적 갭은 명세수준의 상세 정도의 차이에서 발생하는 것으로 전 단계에 비해 많은 정보가 다음 단계에 나타나는 경우 그에 대한 근원과 활용에 대한 명세가 모호해 질 수 있다. 각 단계마다 이런 갭이 발생할 수 있는데 특히 요구분석 단계와 분석단계 중 클래스의 정의 단계 사이에서 발생하는 갭은 이후 개발의 전 과정에 큰 영향을 미치게 되며 개발이후 소프트웨어의 품질에 큰 영향을 미친다.

또한, 최근 객체지향 및 컴포넌트지향 방법론의 주요 관심은 전체과정의 일관성 있는 연결과 완전한 추적성(tracability)에 치우쳐있다. 이 부분은 매우 중요하지만, 분석 초기단계에서 클래스가 완전히 도출되지 않는다면 비용과 시간의 낭비를 초래하게 되며 방법론



<그림 1> 유즈케이스와 클래스 다이어그램 사이의 갭

자체의 크기가 큰 경우 그 낭비는 더 심해진다. 방법론의 크기가 크다는 것은 산출물(artifact)의 종류가 많고 작성과정이 길다는 뜻을 의미한다.[6]

본 연구에서는 이러한 개념적 갭을 완충하기 위한 장치로써 세 가지 다이어그램을 제안하고 이를 적용한 분석결과를 실험을 통해 보인다. 정보다이어그램, 행위다이어그램 그리고 관리다이어그램의 세 가지 다이어그램은 시스템의 관련자가 시스템에 요구하거나 제공 받을 수 있는 서비스에 관점을 두고 각각 정보 행위 그리고 관리사항을 표현한다. 정보는 서비스를 제공하기 위해 필요한 데이터를 의미하고, 행위는 서비스에 수반되어야 하는 세부 행위로 정의한다. 그리고 관리의 서비스의 책임은 누가 가지는 것인가 혹은 서비스의 형태는 무엇인가에 대한 명세이다. 이 방법은 독립적이기 보다는 기존의 클래스를 도출하는 분석모델의 개념적 갭을 줄이기 위한 완충 장치로써, 실험은 두 가지 분석모델에 초보단계와 중간단계의 분석가에게 적용했으며, 본 방법을 RUP에 추가하여 사용한 경우와 그렇지 않은 경우의 비교를 해본다.

2장에서는 세가지 다이어그램의 배경과 최근 RUP의 복잡하고 긴 개발과정을 줄일 수 있는 대안으로 제안된 실험적 방법 EOS를 소개한다. 3장에서는 세 가지 다이어그램에 대한 상세한 설명을 하고 4장에서는 실제 모델에 적용한 결과를 보인다. 마지막으로 결론과 향후 과제로 구성된다.

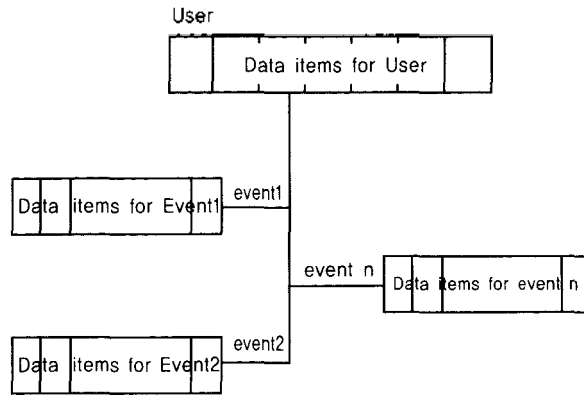
2. 관련 연구

2.1 정보구조모델링(Information Structure Modeling)의 활용

본 연구에서 제안하는 세가지 다이어그램은 기존의 정보구조모델링에서 사용한 다이어그램을 기반으로 했다. 본 연구에서는 작성 기준이 모호한 부분과 본 연구에 적용하는데 맞지 않는 부분을 수정하였으며, 본 절에서는 이에 대한 설명을 한다.

정보구조 모델링은 [1]에서 제안한 정보구조도(Information Structure Diagram)에서 출발하였다. 정보구조도의 기본 개념은 사용자가 시스템에 발생할 수 있는 이벤트(event)를 기준으로 이에 필요한 정보를 표현했으며, [2]와 [3]의 연구를 통해 <그림 2>와 같이 그린다. 그러나, 많은 정보가 있을 경우, 모두 명세하기 힘들고 어느 수준에서 추상화(abstraction)을 해야 하는 지에 대한 기준이 모호하다.

본 연구에서는 이벤트 기준이 아니라 서비스를 기준으로 하고, 정보의 경우 이벤트에 필요한 정보 대신 서비스를 요청하는데 입력 또는 선택해야 하는 정보로 기준을 수정한다. 그리고 정보구조도 내부의 사각형을 정보구조 모델링에서 하나의 데이터 테이블로 취급했던 것을 사용자가 서비스를 요청하기 위해 접근하는 사용자 인터페이스의 개념으로 수정했다. 그러므로 정보의 구조라기 보다는 구성을 의미하므로 정보다이어그램으로 변경했다. 이는 사용자 인터페이스 설계를 개발단계의 초기에 도입하는 최근의 방법론에 많이 부



〈그림 2〉 정보구조도의 개념

합된다.[5]

행위구조도(Behavior Structure Diagram)는 정보구조 다이어그램에서 추출된 이벤트를 수행하기 위한 세부행위를 나타내며 객체지향 프로그래밍언어로 구현 시 메소드를 의미한다. 기존의 행위에 대한 기준이 모호하며 작성수준이 언급되지 않아 실제 모델에 적용하기 위해서는 적용상황에 따라 나름대로 기준을 정해야 한다. 본 연구에서는 클래스의 도출을 지원하는 것이 목적이므로, 서비스에 포함되는 세부서비스 또는 서비스를 구성하는 메소드로 정의한다. 정보다이어그램과 마찬가지로 이유로 행위다이어그램으로 부른다.

관리구조도(Management Structure Diagram)는 추출한 이벤트에 대해 시스템 개발자와 운영자의 관점에서 필요한 정보나 행위를 추출하여 작성한다. 본 연구에서는 서비스의 책임을 누가 맡게 되는지, 서비스를 제공하기 위해서는 어떤 데이터가 필요한지 등을 표현하여, 클래스의 도출을 지원한다.

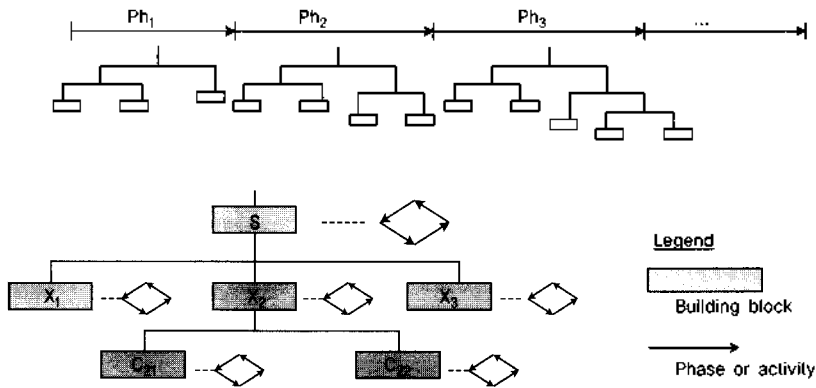
행위구조도와 관리구조도의 모양은 사용자의 이해를 돕기 위해 정보구조도와 같은 골격

을 사용하고, 사각형 내부의 내용만 다르다. 그 외, 정보모델링을 활용한 방법에는 배경도(Context Diagram), 이벤트다이어그램(Event Diagram) 등이 포함된다.

2.2 EOS(Evolutionary, Object-oriented Software development)

1900년대 Rational사의 UML이 소프트웨어 개발 방법론 시장의 점유율이 높아지면서 RUP의 활용이 극대화 되었다. 그러나, 이 방법은 너무나 거대해서 팀의 생산성을 향상시키고, 프로젝트 관리 면에서도 기한에 맞춰 납품을 하는데 장애요인이 되었으며, 이를 극복하기 위한 방법 중의 하나로 EOS를 들 수 있다.[9] 이 방법의 기본 개념은 다음과 같다.

“단계별(phase)로 어떤 행동(certain activities)을 원칙(discipline)에 따라 반복(iteration)하면서 세부 아키텍처 단위(architectural unit: software building blocks or artifacts)로 개발하는 과정”



〈그림 3〉 EOS 프로세스 개념

〈그림 3〉은 EOS의 소프트웨어 개발 과정을 보여주며, RUP을 이용하여 큰 시스템을 개발하는 경우 분석에 미숙하거나 시나리오가 완전하지 않은 경우 무리가 많은 반면, EOS를 이용하면 작은 단위로 나누어 분석, 설계, 구현 및 시험과정을 모두 거치는 방식으로 진행하므로 시나리오의 변경이나 추가에 유연하게 반영할 수 있다.

이 방법은 작은 단위를 대상으로 개발 단계를 반복적으로 적용하여, 전체 모델을 대상으로 분석하는 것에 비해, 시나리오가 완전하지 않은 경우 유리한 방법이다. 하지만, 작은 단위로 나누는 기준이 모호하여 경험이 부족한 분석가가 시도하기에는 여전히 어려운 점이 있다.

3. 클래스 도출

본 연구에서 제안하는 세 가지 다이어그램은 클래스 도출 과정이 필요한 방법에는 모두 적용이 가능하며 이를 설명하기 위해, 기존

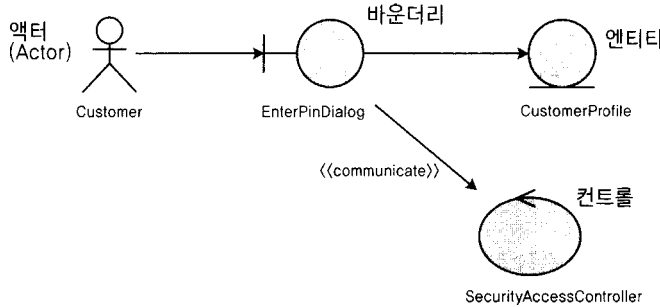
방법론 중 가장 대표적인 RUP를 활용한다.

RUP의 클래스는 엔티티(entity), 컨트롤(control) 그리고 바운더리(boundary)이다. 엔티티는 정적(static) 정보로써 구현 후, 데이터 테이블이 될 수 있다. 컨트롤은 프로그램 모듈로 구현될 수 있으며, 바운더리는 사용자 인터페이스나 API와 같은 관련 시스템과의 인터페이스로 구현될 수 있다.

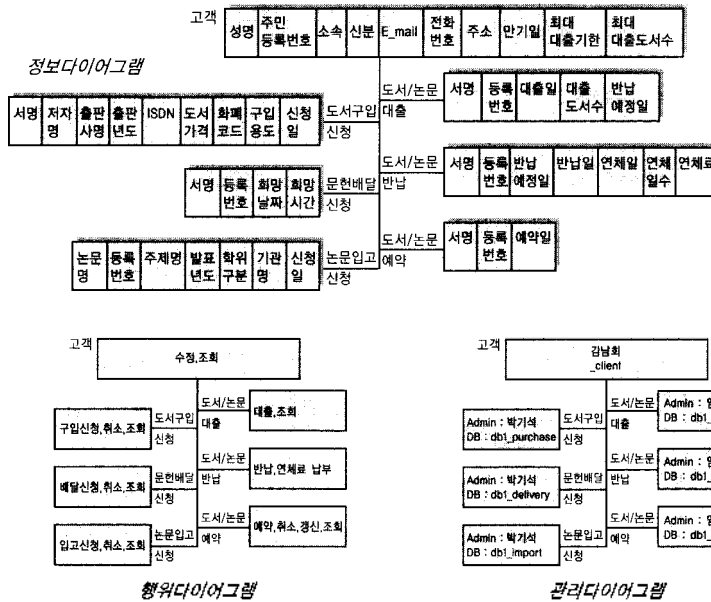
〈그림 4〉는 RUP의 클래스를 기반으로 액터(actor)가 시스템에 서비스를 요청하고 이를 처리해주는 메커니즘을 모델링 한 것이다. 여기서 액터는 하나의 클래스로 데이터 테이블로 구현될 수 있다.

본 연구에서 제안하는 세가지 다이어그램의 예로 〈그림 5〉는 도서관리시스템의 일부를 분석한 것이다.

정보다이어그램은 고객이 시스템에 접근하기 위해 필요한 정보-성명, 주민등록번호, 소속 등을 보여준다. 또한, 도서구입 신청, 문헌배달 신청, 논문입고 신청, 도서는문 대출, 도서는문 반납 그리고 도서는문예약의 서비스를 요청할 수 있고, 각 서비스에 따른 필요



〈그림 4〉 RUP의 클래스 종류



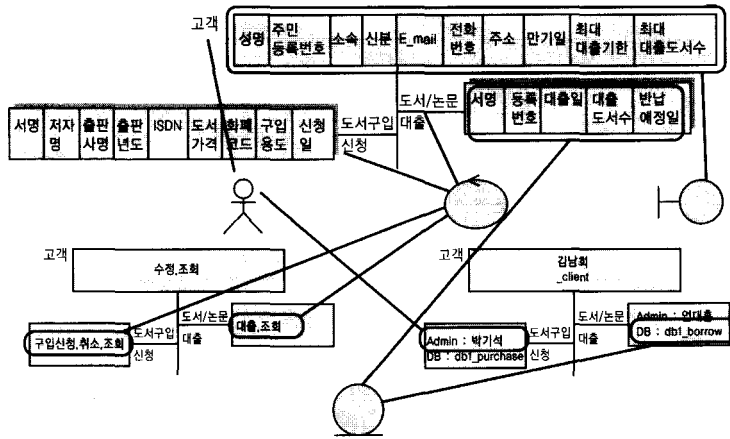
〈그림 5〉 세가지 다이어그램의 적용

정보가 무엇인지를 보여준다. '고객'은 RUP의 액터가 될 수 있고, 성명, 주민등록번호 등은 바운더리가 된다. 한편, 각 서비스는 컨트롤이 되며 이에 필요한 정보는 클래스의 데이터항목으로 분류된다.

행위다이아그램은 정보다이아그램과 같은 골격을 사용하며, 정보다이아그램에서 도출

된 서비스에 대한 상세서비스 또는 메소드를 정의하며, 이는 서비스가 컨트롤 클래스로 도출되면서 그에 포함되는 메소드 또는 기능 항목으로 분류된다.

관리다이아그램은 상황에 따라서 가변적으로 상세 정도가 달라질 수 있는데, 본 예에서는 두가지를 취급한다. Admin은 서비스 관리



〈그림 6〉 RUP클래스와의 연관관계

자를 의미하며, DB는 서비스를 제공하기 위한 정보가 저장된 데이터베이스를 의미한다. DB는 엔티티로 구분되며, Admin은 이에 연관된 액터로 구별된다. 이를 바탕으로 〈그림 6〉은 〈그림 5〉의 일부가 RUP의 클래스로 분류되는 연관관을 보여준다.

결국, 세가지 다이어그램을 활용해 기존 RUP의 개념적 갭을 완충할 수 있다는 것을 보여준다. 경험이 많은 분석가의 경우에는 갭이 문제가 되지 않고 오히려 시간이 많이 걸리는 작업이 될 수 있지만, 중간 수준 이하의 경우에는 갭을 완충하는 본 연구의 제안을 사용하지 않더라도 시간을 단축시킬 수는 없을 뿐더러, 시간이 단축되었다 하더라도 믿을 만한 클래스 도출이 되지 못하는 경우가 대부분이다.

이러한 상황을 감안하여, 초보 그리고 중간 수준의 집단에게 두 가지 모델을 분석하는 실험을 해보았다.

4. 실험 및 분석

12명의 인원을 선별해서 초보 두 그룹과 중간수준 두 그룹으로 나누고 의료클리닉과 은행의 현금지급기의 분석을 실험했다. 초보는 두 번 이하의 분석 경험의 소유자이고, 중간수준은 세 번 이상 열 번 이하의 경험을 가진 사람들로 구성했다. 대상 도메인은 의료클리닉과 은행의 현금지급기 두 가지이며, 의료클리닉은 상대적으로 복잡한 규모이고 현금지급기는 작고 단순한 규모의 대상이다. 두 시스템은 이미 구현이 되어 있는 상태의 모델이며, 〈표 1〉은 실제 구현된 클래스의 수에 대한 실험 대상자들이 도출한 클래스의 수의 비율이다.

〈표 2〉는 클래스 도출에 걸린 시간을 초보 실험자가 기본 RUP를 적용했을 때를 100으로 한 비율이다.

본 연구에서 제안한 방법의 효과를 알기 위해, 클래스 도출의 정확도와 도출에 걸린 시간을 이용해 다음과 같은 식을 구했다.

〈표 1〉 클래스 도출 비율

사례모델	분석수준(평균값)	기본 RUP	본 연구 제안 적용
의료클리닉	초보	68.5%	82.7%
	중간	75.5%	87.2%
현금지급기	초보	70.2%	88.3%
	중간	82.5%	92.0%

$$class_extraction_time = \frac{required_time}{The_ratio_of_extraction}$$

클래스의 도출 시간은 클래스 도출에 걸린 시간에 대한 클래스 도출의 정확도의 비율로써, 이는 정확한 클래스를 도출하는데 걸린 시간을 구하는 의미이다. required_time은 실험 대상자가 클래스를 도출하는데 걸린 시간을 측정하는 것이고,

The_ratio_of_accurate_extraction은 실험 대상자가 도출한 클래스와 실제 모델에서 사용되고 있는 클래스를 비교하여 적중비율(hot ratio)을 구한 값이다. 결국, 절대적으로 시간이 많이 걸리지만 정확한 클래스가 많이 도출된 경우와 시간이 적게 걸렸지만 클래스가 정

〈표 2〉 클래스 도출 시간

사례모델	분석수준(평균값)	기본 RUP	본 연구 제안 적용
의료클리닉	초보	100	118
	중간	65	71
현금지급기	초보	95	118
	중간	60	69

〈표 3〉 정확한 클래스 도출시간

사례모델	분석수준(평균값)	기본 RUP	본 연구 제안 적용
의료클리닉	초보	1.46	1.43
	중간	0.86	0.81
현금지급기	초보	1.35	1.34
	중간	0.76	0.75

확하지 않은 경우를 비교한다면, 전자의 경우 본 연구의 제안을 적용한 효과가 있다고 판단할 수 있다. <표 3>은 <표 1>과 <표 2>의 값을 식에 대입한 결과를 보여준다.

<그림 6>은 정확한 클래스를 도출하는데 걸린 시간을 그래프로 보여준다. 초보그룹이 두 모델에서 중간그룹 보다 정확한 클래스 도출 시간이 더 많이 필요했음을 알 수 있다.

5. 결론 및 활용방안

본 연구에서는 기존 객체지향 및 컴포넌트 지향 소프트웨어 개발 방법의 분석단계에서 발생하는 개념적 갭으로 인한 불완전한 클래스 도출을 지원하기 위해 세가지 다이어그램을 제안하고 이를 실험하고 분석해 보았다.

그 결과, 본 연구의 제안을 적용했을 경우 절대적으로 시간은 많이 걸리지만 정확한 클래스를 도출할 수 있으므로 본 연구의 제안이 효과적으로 적용될 수 있음을 보였다.

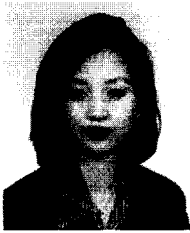
이 방법은 클래스 도출의 과정이 포함되는 개발과정에 모두 적용할 수 있는 완충기능으로 특히 분석과정이 과다하게 복잡하여 문제가 되거나 개념적 갭이 상대적으로 크고, 분석결과물(artifact)사이의 일치성이 떨어지는 경우에 더욱 효과적으로 이용될 수 있다. 또한, 최근의 방법론들은 과거에 비해 방법론 자체의 규모 면에서 커지는 상황이다. 두 사람이상의 분석가들이 팀워크를 하는 경우 분석관점이나 기준의 모호함을 줄이기 위해 본 연구의 세가지 다이어그램을 활용할 수 있다.

참 고 문 헌

- [1] 박재년, "정보구조 모델링에 의한 시스템 분석," 숙명여자대학교 논문집, 제33집, pp.677-692, 1992
- [2] 이서정, "사용자중심의 일관성을 보장하는 객체지향 시스템 분석 및 설계 방법론," 숙명여자대학교 박사학위 논문, 1998
- [3] 정정인, "비즈니스 업무에서의 정보구조 모델링을 이용한 객체추출 방법에 관한 연구," 숙명여자대학교 석사학위 논문, 2000
- [4] A. Evans, "Rigorous Development in UML," ETAPS'99, FASE Workshop, LNCS, pp.129-144, 1999
- [5] John Vu, "Software Process Improvement," KIPA S/W 프로세스 개선 실무자 교육 과정, 2004
- [6] K.D.Schewe, "UML: A Mordern Dinosaur? A Critical Analysis of the Unified Modeling Language," proceeding of Information Modeling and Knowledge Bases XII, pp.185-202, 2001
- [7] UML DISTILLED: A Brief Guide to the Standard Object Modeling Language, Martin Fowler, Addison?wesley, chap1-2, 1999
- [8] "The UML ? Two years after adoption of the standard," <http://home.earthlink.net/~salhir/theumltwoyearsafteradoptionofthestandard.html>

- [9] Wolfgang Hesse. "Dinosaur Meets Archaeopteryx? or: Is There an Alternative for Rational's Unified Process," Proceeding of Software and Systems Modeling, vol.2, 2003 http://www.mathematik.uni-marburg.de/~hesse/papers/Hes_01b.pdf

저 자 소 개



이서정

1989.

1991.

1993 ~ 1998.

1998 ~ 2003.

2003 ~ 현재

관심 분야

(E-mail : sjlee815@sookmyung.ac.kr)

숙명여자대학교 전산학과(이학학사)

숙명여자대학교 대학원 전산학과(이학석사)

숙명여자대학교 대학원 전산학과(이학박사)

동덕여자대학교 강의전임교수

숙명여대 및 명지대 대학원 강의 / 숙명여대 연구원

Software Achitecture, Mobile Contents Delivery,

Middleware Repository