

객체지향 시스템 모델링 차원: 이론 및 탐색적 평가

안 준 모*

A Study on the Dimensions of Object-oriented Systems Modeling : Theory and an Exploratory Evaluation

Joon M. An*

Abstract

This study proposes evaluation dimensions of object-oriented systems modelling tools and activities available in object-oriented systems development practices. The dimensions are developed based on previous research in cognitive psychology, information systems modeling study, and object-oriented systems analysis and design areas. The proposed dimension is composed of two dimensions. The one dimension includes abstraction levels of modelling and the other includes process and representation in modeling activities.

Experts on object-oriented modeling were selected to evaluate the practical validity of the proposed dimensions and applications of major object-oriented modeling tools during systems development project. Most of the tools were observed to be used for representing objects rather than for modeling the process of related objects. The proposed modeling dimension will be evaluated for acquiring general validity in future empirical research.

※ 본 연구는 건국대학교 2000년도 학술연구진흥비 지원에 의해 수행되었음.
* 건국대학교 경영·경영정보 학부 부교수

1. 서론

정보시스템 개발 생산성 향상, 품질확보, 유지보수 비용의 절감은 정보시스템관리 분야의 핵심이슈이다. 특히 21세기 지식·정보 사회에서 정보시스템의 신속하며 효과적 개발 및 유지보수는 기업의 경쟁 민첩성(agility)을 결정짓는 주요한 요소로 등장하게 될 것이다. 기업의 경쟁 민첩성을 지원하는 정보시스템 개발을 위해서 최근 객체지향 정보시스템 분석기법 활용이 확산되어가고 있다. 객체에 기반한 정보시스템 개발환경에서 객체기반 모델링 틀의 효율적 활용 여부는 정보시스템 개발 생산성, 양질의 정보시스템 구축, 나아가서는 기업의 경쟁력을 결정짓는다고 할 수 있다.

정보시스템 모델링 활동 및 절차의 자동화가 진행되면서 정보시스템 모델링 활동에 대한 이해와 이를 지원하는 모델링 틀의 평가에 대한 관심은 실무와 이론적 연구의 주요한 관심사가 되었다[Essink, 1986]. 특히 실무에서는 모델링 틀의 활용이 일반화되면서 프로젝트 관리자는 모델링 활동을 프로젝트의 특성에 따라 구성하고 이를 지원하는 모델링 틀을 적절히 활용함으로써 시스템 개발의 생산성과 최종 사용자의 만족도를 높이고자 한다. 정보시스템 모델링 활동은 성공적인 정보시스템 구축 즉, 양질의 정보시스템 구축, 생산성 향상 및 유지보수 비용의 절감에 결정적인 영향을 미치는 것으로 평가되고 있다[Winogard, 1995].

본 연구의 목적은 객체지향 모델링 활동을 체계화하여 모델링 활동을 구성하는 차원을 제시하고 이에 근거하여 최근에 활용되는 객체지향 모델링 틀을 평가하는 데에 있다. 구체적으로 첫째, 객체지향 모델링 틀에 의해 형성되는 시스템개발 모델링 활동의 개념을 정립하고 둘째, 이론에 근거한 모델링 활동에 대한 평가 체계를

제시하고 셋째로, 도출된 모델링 체계에 근거하여 기존에 활용되는 객체지향 모델링 틀을 평가함으로써 제시된 평가차원의 활용성을 제시한다. 마지막으로는 평가차원의 실무적 의미와 향후 연구를 위한 이론의 정립 방향을 제시하는데 있다.

본 연구결과는 모델링 활동에 대한 평가 차원을 제시함으로써 기존에 활용된 모델링 틀의 활용을 최적화 시킬 수 있는 이론적 근거를 제시하게 된다. 또한 시스템 개발 프로젝트 관리자는 프로젝트에 활용되는 다양한 객체지향 설계 및 분석 틀을 활용하여 프로젝트 별 최적 모델링 활동을 구성할 수 있을 것이다. 이론적인 측면에서는 본 연구에서 제시된 객체지향 모델링 체계는 향후 객체지향 모델링 연구를 위한 연구의 방향, 가설 도출 기반을 제공할 것으로 기대된다.

본 논문은 아래와 같이 구성된다. 2절에서는 객체지향 모델링 활동, 모델링 단계, 모델링 대상에 관한 기존 연구를 분석하여 객체지향 모델링 활동을 구성하는 모델링 차원을 도출한다. 3절에서는 2절에서의 이론적 연구를 기반으로 하여 두 개의 모델링 차원으로 구성된 모델링 활동 체계를 제시한다. 제 4절에서는 전 절에서 제시된 모델링 체계에 근거하여 기존에 객체지향 모델링 활동에 활용되는 틀을 평가하고 마지막으로 5절에서는 향후 이론적 연구의 방향과 실무적 활용가능성, 연구의 한계를 설명한다.

2. 기존 연구 정리 및 분석

객체지향 시스템 개발 관리자는 모델링 틀을 활용하여 시스템 개발 프로젝트의 복잡성을 감소시키고 이를 통하여 시스템의 효율적 구축과 사용자 만족의 극대화를 이루고자한다. 실무에서 객체지향 모델링 틀이 광범위하게 확산, 활

용되고 있음에도 불구하고 이론적 측면에서 다양한 객체지향 모델링 틀이 어느 단계의 모델링 활동에 활용되고 있으며 각 모델링 단계에서 어떤 활동에 활용되는지에 대한 체계적이며 통합적 연구가 부재한 것이 사실이다.

본 절에서는 객체지향 모델링 활동에 관한 기존 연구에 대한 분석을 통하여 기존 연구의 범위 및 한계점을 제시한다. 본 절의 첫 부분에서는 기존 객체지향 모델링 활동에 관한 연구를 분석, 요약하여 제시한다. 두 번째 부분에서는 객체지향 모델링 활동 단계에 관한 기존 연구를 통합 제시한다. 마지막 절에서는 객체지향 모델링 활동 유형에 대한 기존 연구를 요약, 제시한다. 이러한 기존 연구를 분석함으로써 모델링 활동의 구체적 차원과 유형을 제시하고 이를 기반으로 하여 제 3절에서 객체지향 모델링 활동 평가 차원을 제시하게 된다.

2.1 객체지향 모델링 활동에 관한 기존 연구

객체지향 개념이 제시되고 이와 관련된 프로그래밍언어가 활용된 이후로 객체지향 시스템 개발 관련 연구가 다각적으로 이루어졌다. 객체지향 모델링 활동을 구성하는 차원을 도출, 제시하기 위해서 기존 객체지향 모델링 틀나 활동과 관련한 연구를 분석해 봄으로써 모델링 차원에 대한 연구의 필요성을 제시하고자 한다.

박광호[1999]는 분석단계에서 활용되는 유스케이스(use case) 모델링 방법을 시스템 개발의 마지막 단계인 테스트 단계까지 확장하여 사용함으로써 개발 과정의 일관성과 연속성을 유지할 수 있음을 강조한다. 이 연구는 분석, 설계 단계에서 작성한 유스케이스를 단위테스팅과 통합테스팅 레벨까지 확장하여 활용하여 시스템 개발 성공 가능성을 높일 수 있음을 제시하였다. 객체지향 모델링 활동이 전통적 모델링

활동에 비하여 상위 모델링 활동과 하위 모델링 활동 사이의 일관성과 상호 검증성을 가능케 하는 모델링 틀에 의해 지원될 수 있음을 사례 연구를 통하여 입증하였다.

Kjaer & Madsen[1995]의 연구에서는 병행방사선과 응용시스템 개발에 있어서 유연성에 초점을 맞춘 참여적 개발 방법론을 제안하였다. 모델링 방법으로 청사진 매핑(blueprint mapping)과 조직 게임(organizational game) 방법을 제시하였다. 청사진 매핑 방법은 부서에서 행해진 다양하고 예측할 수 없는 상황을 지도 형태로 표현한다. 조직 게임 방법은 특정 상황을 묘사하는 카드를 활용하여 요구사항의 변화에 유연성 있게 대처할 수 있는 모델링 방법이다. 이러한 기법은 기존의 접근 방법에서는 작업 활동만을 강조했던 것에 비해, 작업 활동의 다양성과 유연성(flexibility)을 고려한 모델링 기법이다.

Bohrer, Johnson, Nilsson & Rubin[1998]은 비즈니스 분산 객체 어플리케이션 구축을 위한 객체 컴포넌트(components) 모델 구성 체계인 샌프란시스코(San Francisco)체계를 제시하였다. 이 체계는 분산 어플리케이션 개발 및 확장을 지원하는 상위 비즈니스 객체 컴포넌트부터 하위의 인프라 객체 컴포넌트를 제공하는 모델링 체계이다. 이를 통하여 표준화된 객체 즉 컴포넌트 재사용을 촉진하고 생산성을 향상시킬 수 있다. 이러한 모델링 체계는 특히 유사한 객체의 재사용성을 강조하여 유지보수 및 시스템 개발 프로젝트의 생산성 향상에 초점을 두고 있다.

Parson & Wand[1997]는 객체지향 시스템 분석 단계에서의 모델링 활동을 존재론적(ontology)이고 인지적인(cognitive) 측면에서 연구하였다. 객체 개념의 적용은 구현(implementation), 생성보다는 객체를 표현(representation)함에 적합하다는 것을 보여주고 이에 근거한 시스템 분석을 위한 지침(guideline)을 제시하였

다. 이 연구는 객체지향 시스템 분석 활동이 상위 단계 모델링에 영향을 미침으로서 시스템의 존재론적 의미와 최종 사용자의 시스템과의 인지적 상호작용을 증진시킬 수 있음을 제시하였다.

Purao, Jain & Nazareth[1998]는 객체지향 시스템 개발을 위한 효율적인 객체 분산 모델에 대해 연구했다. 효율적 분산시스템을 개발을 위해서 객체지향 시스템의 구성개념인 상속성(inheritance), 캡슐화(encapsulation)를 활용하였다. 제안된 분산 시스템 모델링 체계는 객체 상호작용을 모델링하는 위치간 분산(intersite distribution)과, 어플리케이션 기능간의 상호작용을 모델링하는 내부간 분산(intrasite distribution)의 2단계(phases)를 포함한다. 이는 기존의 모델링 기법이 객체사이의 모델링에 치우친 점을 고려하여 분산환경 하에서 위치간 또는 내부 객체간의 분산을 고려한 확장된 모델링 기법을 제시하였다.

Kaindl & Carroll[1999]에 의하면 모델링은 주어진 목적을 위해 특정 분야의 정보와 지식을 추상화하는 활동으로서 이는 인간과 컴퓨터 인터페이스(human computer interface), 인공지능(artificial intelligence), 소프트웨어 공학에서의 객체지향 분야 등 여러 분야에서 활용되고 있다. 기존의 연구를 분석한 결과, 객체 지향 모델링에 공통적으로 이해될 수 있는 폭 넓은 심볼릭 모델링 틀이나 기법의 필요성을 강조하였다. UML(Unified Modeling Language)과 같은 다이어그램 언어는 시각적 모델링을 제공함으로써 다양한 시나리오에 상응하는 표현 방법을 제공할 수 있다고 주장한다.

Butler, Esposito & Hebron[1999]에 의하면 소프트웨어구축은 최종 사용자가 작업상 필요한 목록, 형식, 일련의 정보를 주요 대상으로 다루기 때문에 사용자의 작업모델이 필요하다. 작업모델(work model)과 소프트웨어 모델(soft-

ware model)은 둘 다 상호작용적인 면을 고려하여 디자인되어야 하지만, 근본적인 차이점을 가지고 있다. 작업모델의 목적은 더 나은 자원 유용성, 시간(time)과 차이(variance)의 단축, 비용과 낭비의 감소와 같은 물질적인 면에 있다. 그에 반해 소프트웨어 모델은 완벽성과 일관성, 실행할 코드의 효율성에 중점을 맞추고 있다. 이러한 차이점을 통합하는 일은 어려운 일이지만 작업모델에서 고려되는 기술 및 비용과 소프트웨어모델에서 고려되는 이점과의 합리적인 절충이 필요하다. 이와 같은 절충을 위한 모델링 틀의 대표적 예로는 UML에서 사용사례(use-case)와 사용자 활동(user activity) 다이어그램, 사용자 인터페이스 개발을 위한 사용자 작업 시나리오 표현을 대표적인 예로 들 수 있다. 작업과 소프트웨어 모델 연결의 주된 이점은 시스템 활용의 목적을 구현 활동으로부터 분리시킴으로써 디자인 활동 단계에서 통찰력(insight)이나 독창성(creativity)을 증가시킬 수 있다는 점이다.

Robillard[1999]는 시스템 분석 전문가는 프로그래밍 이전에 특정한 지식 구조에 따라 지식을 설명하고 구성할 수 있어야 함을 강조한다. 소프트웨어 개발은 지식 집약적(knowledge-intensive)인 지식 처리 과정이라 할 수 있다. 즉 컴퓨터에 의해 읽혀지고 실행될 수 있는 언어 안에 연속적인 지식을 구체화(crystallization)시킨다고 말할 수 있다. 지식의 구체화는 어플리케이션 도메인 지식으로부터 소프트웨어 아키텍처와 알고리즘 디자인 지식으로 옮겨지고, 결국은 프로그래밍 언어로 구체화된다. 소프트웨어 엔지니어는 지식 구체화 처리를 쉽게 해줄 방법과 기법(practice), 틀을 개발해 왔다. 인지과학자들은 다양한 관점을 가지고 지식의 속성을 연구했다. 이러한 두 분야에서의 접근방법을 결합하기 위해서 인지과학(cognitive sci-

ence)에서는 지식 구조와 표현을 연구함으로써 소프트웨어 공학의 새로운 방향을 찾고 있음을 주장하며 소프트웨어 개발은 스키마 생성, 스키마 디폴트값의 검증, 주 지식, 에피소드 지식, 계획 활동, 문제 정의를 위한 형식적 명세서, 다양한 사건(phenomenon)을 관리하기 위한 틀 활용에 의해서 지식의 구조나 표현의 향상을 기할 수 있음을 강조하였다.

Winograd[1995]는 전통적인 모델링 활동에서 강조되었던 하위 단계 모델링 활동인 상호작용적 프로그래밍(interactive programming), 명세(specifications), 재사용 코드(reusable code), 상호작용적 디버깅(interactive debugging), 프로그래밍보다는 프로토타이핑 방법(responsive prototyping), 사용자 개념 모델(user conceptual models), 디자인 언어(design languages), 참여적인 디자인(participatory design)과 같은 상위단계 모델링 활동을 지원하는 틀들의 활용

을 강조하였다. 이 연구에서는 객체지향 모델링 환경으로 갈수록 기존의 절차적 프로그래밍보다는 객체 즉 모듈 프로그래밍과 같은 표준화된 객체에 의한 시스템 구현이 가능해지기 때문에 이와 같은 시스템 구축 환경에서는 프로그래밍과 같은 하위 수준의 모델링 활동보다는 사업설계, 시스템분석, 시스템 디자인과 같은 상위 수준의 모델링 활동의 중요성이 높아짐을 강조한다.

Beyer & Holzblatt[1995]는 사용자 요구사항을 파악함에 있어서 인터뷰와 설문 외에 좀더 새로운 접근방법의 필요성을 주장하였다. 디자이너와 사용자 사이의 새로운 관계성을 알아냄으로써 좀더 나은 요구사항을 찾을 수 있는 접근방법인 견습 모델(apprenticeship model) 방법을 제시한다. 견습 모델 방법은 디자이너가 작업 종류, 구조를 관측하고, 사용자의 경험으로부터 배우고, 관측하고 이해한 바에 따라 응답하고 명확하게 전달하고, 작업을 향상시킬 수

〈표 1〉 객체지향 모델링 활동 관련 연구 요약

저 자 [연 도]	연 구 내 용
박광호[1999]	· 객체지향 방법론의 모델링 활동이 개발 전 과정에 일관성 있게 적용될 수 있음을 사례 분석을 통하여 제시
Kjaer & Madsen[1995]	· Blueprint mapping, 조직게임과 같은 모델링 활동의 유연성을 보장하는 모델링 활동지원을 사례연구를 통하여 제시함
Bohrer, Johnson, Nilsson & Rubin[1998]	· 분산 객체 어플리케이션을 위한 San Francisco 체계 개발 · 기존 시스템의 객체지향 시스템으로의 변환 체계 제시
Parson & Wand[1997]	· 시스템 분석단계에서의 객체 사용 · 구현보다는 표현방법에 적합함을 제시 · 시스템 분석을 위한 지침 제시
Purao, Jain & Nazareth[1998]	· 클라이언트/서버 기반의 객체지향 어플리케이션의 효율적인 분산처리 방법 제시
Kaindl & Carroll[1999]	· 객체지향 모델링에 공통적으로 이해될 수 있는 표준화된 모델링 틀의 필요성 강조
Butler, Esposito & Herbon[1999]	· 작업모델과 소프트웨어 모델의 상호작용을 지원할 수 있는 모델링 환경의 필요성 강조
Robillard[1999]	· 지식 구체화(crystallization)를 지원하는 모델링 활동의 중요성 강조
Winograd[1995]	· 객체지향 환경에서 상위 수준 모델링 활동의 중요성 강조
Bayer & Holzblatt[1995]	· 사용자와 시스템 디자이너 사이의 상호 작용을 촉진시키는 견습 모델(apprenticeship model)제시함. 이는 학습을 통한 업무 패턴과 객체를 발견하기 위한 모델링 기법이다.

있는 방법을 생각하며 지원 작업 등 특정분야에 대해 사용자와 이야기함으로써, 사용자의 작업을 함께 이해하고 검토하여 상호작용적으로 시스템을 개발 할 수 있게 한다. 디자이너는 인터뷰, 전문가의 도움, 사용자 대화를 통해서 업무 패턴 즉, 업무 관련 객체를 발견하고 이를 해석, 조율하게 된다.

<표 1>은 위에서 논의된 이론적 연구 결과를 요약적으로 정리, 제시하고 있다. 대부분 연구가 객체지향 모델링 툴의 유효성과 기능적 차원에 대한 체계를 제시하고있다. 객체지향 모델링에 관한 개념적 연구는 주로 모델링 활동의 일부 단계나 일부 측면에 국한하여 연구가 진행되어 왔다. 또한 실증 및 사례 연구에서도 모델링 활동에 대한 정의와 체계를 제시하지 못한 상황에서 모델링 활동의 단편적 부분에 집중하여 연구가 진행되어서 그 연구 결과에 대한 일관성 있는 이론적 해석이 어려운 것이 사실이다.

2.2 객체지향 모델링 활동 단계에 관한 연구

전 절에서 살펴 본 바와 같이 기존의 관련 연구는 객체지향 모델링 활동 분야 및 툴의 기능적 유용성에 초점을 둔 연구가 주를 이루고 있음을 알 수 있었다. 이와 같은 연구는 객체지향 모델링 활동의 일반적 유용성이나 그 효과성을 입증하는 데는 부분적으로 공헌하였다고 할 수 있으나 객체지향 모델링 툴이 어떠한 단계의 모델링 활동에 영향을 미치고 있으며 모델링 대상 객체의 어떠한 측면을 모델링 대상으로 하고 있는지를 밝혀주지 못함으로써 객체지향 모델링 툴의 구체적 활용분야나 평가 체계를 구성하는데 도움을 주지 못하고 있다. 객체지향 설계 및 분석툴이 제공하는 시스템개발 모델링 활동이 각 추상화 층(abstraction layers)의 어느 차원에 공헌을 하고 이러한 공헌이 대상 모델의

어떠한 특성을 모델링 대상으로 하고 있는 지에 체계적인 개념, 체계를 제시하지 못하고 있다.

객체지향 소프트웨어 공학(object oriented software engineering)은 복잡한 문제를 해결하는 인간의 인지적 과정을 소프트웨어 개발 모델링 활동에 적용한 개발 방법론으로서 이는 인간의 인지적 모델링 체계를 바탕으로 하여 상위 객체부터 이에 속한 하위 객체까지를 단계적으로 모델링하는 소프트웨어 개발 및 분석기법이다. 객체지향 시스템개발 모델링 툴이 객체지향 프로그래밍 언어와 같은 코딩을 위한 하위(low-level) 모델링 툴로부터 사업 아키텍처(business architecture)설계, 정보전략 계획, 업무 흐름분석을 위한 상위(high-level) 모델링 툴까지를 포함한다고 할 때에 각 모델링 활동 단계를 구체화하고 이러한 단계별 지원 활동을 체계화할 필요가 있다[Ericsson & Jacobson, 1995].

인지심리학에서 제시된 추상화 모델링 이론(abstraction theory of modeling)은 모델링 활동의 계층화는 단계적으로 모델링 활동의 구체성을 증가시킴으로서 인지적 복잡도를 감소시킨다는 것이다. 시스템 개발을 위한 모델링 활동이 상위 모델링 활동부터 단계별로 하위 모델링 활동까지 계층화됨으로서 모델링 활동과 관련된 인지적 복잡도(complexity)를 감소시킬 수 있다는 것이다. 이를 위해서 상위 모델 도출부터 하위 실행모델 도출까지를 단계적으로 추상화한다. 객체 지향 모델링 툴은 상위 객체부터 하위 관련 객체까지를 일관성 있게 체계화하여 분석, 설계, 구현하는 모델링 활동을 지원한다는 것이다.

정보시스템 구축을 위한 객체지향 설계 및 분석툴은 프로그램 코딩단계에서의 모델링이나 시스템 분석 및 설계의 수준을 뛰어넘어서 비즈니스 아키텍처(business architecture) 설계, 정보전략 계획, 업무 흐름분석, 비즈니스

재구축을 위한 모델링 활동 즉, 상위(high-level) 모델링 활동을 지원하게 되었다[Ericsson & Jacobson, 1995]. 또한 모델링 대상에 있어서도 기존의 프로세스 중심의 즉 절차 중심의 모델링 활동뿐만 아니라 모델링 대상 시스템의 정적인 측면, 동적인 측면을 표현(representation)하는 단계에 이르게 되었다. 객체지향 모델링 활동은 이와 같이 모델링 차원에 있어서의 확대뿐만 아니라 모델링 대상에 있어서의 모델링 활동의 확장 및 다양화를 이루게 되었다.

Howden[1982]은 모델링 툴이 지원할 수 있는 모델링 활동의 단계를 4단계로 구분하여 제시하고 있다. 이는 요구사항(requirement) 분석 단계, 디자인 단계, 코딩 단계, 검증 단계로 구성된다. 객체지향 시스템 분석 및 설계 프로젝트에서 활용되는 모델링 툴은 모델링 상위 단계에서 하위 단계까지 일관성 있게 객체를 모델링하기 위해 사용된다.

Iivari & Koskela[1987]는 추상화 모델링 단계를 실용적(pragmatic)단계, 입력/출력(input/output) 단계, 구축(constructive)단계로 상위 조직의 목적을 위한 모델링 단계부터 하위 구축을 위한 모델링 단계로 제시하고 있다. 이 연구는 객체지향 모델링 활동에 국한하지 않고 시스템 분석 및 설계 시에 구성될 수 있는 일반적 모델링 단계를 제시하고 있다. 이들은 정보시스템의 조직적 목적 단계의 모델링부터 최종 프로그래밍 단계를 제시하고 모델링 활동이 이러한 단계적 일관성을 유지할 때 성공적으로 구축될 수 있음을 주장한다.

Olive[1983]는 외부(external)단계, 개념(conceptual)단계, 논리적 아키텍처(logical architectural)단계, 물리적(physical) 단계로 모델링 차원을 계층화하여 제시하고 있다. 외부 단계는 모델링의 주 대상 시스템과 인터페이스하고 있는 관련 환경 요인에 대한 시스템 모델링 단계

를 나타내며 개념적 단계는 모델링 시스템의 개념 단계에서의 상위 모델을, 이를 구현하기 위한 논리 단계의 구성 요인사이의 관계도인 논리적 아키텍처와 이를 하드웨어적 시스템으로 구체화한 물리적 단계로 구분하여 제시하고 있다.

Essink[1986]은 객체 시스템 모델(object system model), 개념적 정보 시스템 모델(conceptual information system model), 데이터 시스템 모델(data system model), 구현 모델(implementation model)을 포함한 4단계로 모델링 단계를 구분하였다. 이러한 단계는 객체시스템 모델을 최상위 모델링 단계로 제시한 점에서는 여타의 모델링 단계에 비하여 진일보한 연구라 할 수 있으나 하위 단계는 이러한 객체 모델의 일관성 있는 변환 단계라기 보다는 전통적 모델링 활동에 근거하여 제시된 모델링 활동으로 구성되었다.

Cervený, Garrity & Sanders[1990]에 의하면 상위 모델링 활동(1. 사업객체 모델링 2. 엔터프라이즈 모델링 3. 주요사업 프로세스 모델링, 4. 개념적 데이터/프로세스 모델링)을 위한 객체지향 설계 및 분석 툴의 활용은 최종 사용자 요구사항의 포괄성 있는 반영, 사업 업무 프로세스의 캡슐화 및 분석을 도와줌으로써 최종 개발된 시스템의 사용자 측면에서의 시스템 성공도를 높일 수 있다. 하위 모델링 활동(1. 에디터/프로세스의 통합 2. 객체지향 프로그래밍/테스팅)을 지원하는 객체지향 설계 및 분석 툴의 활용 정도는 소프트웨어, 프로그램 모델링, 시스템 테스트를 도와줌으로써 시스템 개발 측면에서의 효율성 즉, 프로그래머 및 시스템 분석 전문가의 생산성, 유지/보수비용의 절감, 시스템의 신뢰성 증가, 시스템 모듈의 재사용 등에 영향을 미친다고 한다. 다시 말해서, 추상화의 상위 레벨에서의 시스템 개발 모델링 활동은 문제 해결 활동이나 사용자 정보 만족의 유효에 영향을 주

게 된다는 가설이다. 하위 단계에서의 모델링 틀 적용은 프로젝트의 효율성이나 생산성에 영향을 주는 것으로 주장한다. 예를 들어 구조화된 프로그래밍, 의사코드(pseudo code), 그리고 의사결정(decision)테이블 등의 시스템 구현단계에서 활용되는 모델링 틀들은 컴퓨터 스키마 형성을 도와줌으로써 시스템 구축의 효과성을 높일 수 있다는 것이다. 상위 모델링 수준에서 활용되는 틀, 예를 들어 데이터 흐름 다이어그램, 전략적 데이터 플랜 등은 시스템의 개념적 모델 형성을 지원하고 이를 기반으로 다양한 구성에 대한 대안을 제시해줌으로써 최종적으로는 사용자가 지향하는 시스템을 효과적으로 개발하는 데에 공헌할 수 있다는 주장이다.

Fayad & Schmidt[1997]는 객체지향 어플리케이션 체계(object-oriented application dimension), 즉 모델링 체계를 그 영역에 따라 시스템 인프라구조 체계(system infrastructure dimension), 미들웨어 통합 체계(middleware integration dimension), 엔터프라이즈 어플리케이션 체계(enterprise application dimension)으로 구분하였다. 이러한 모델링 체계는 소프트웨어 품질의 향상과 개발 노력 감소 등을 통한 생산성 및 시스템 품질 향상을 가져올 수 있으나 개발 노력, 학습곡선(learning curve), 통합성, 유지보수, 검증과 결점제거, 효율성, 표준안 등에 대한 체계적인 지원 없이는 바라는 효력을 발휘할 수 없다고 하였다. 객체지향 어플리케이션 체계는 분산 시스템 환경에서 확장성과 재사용성을 보장하나 성공적인 재사용을 위해서는 인프라구조 개발자와 엔터프라이즈 어플리케이션 개발자간의 체계 통합이 필요함을 강조한다. 즉 상위 모델링 활동을 위한 체계와 구현을 위한 모델링 체계 간의 일관성과 통합성의 중요성을 강조하고 있다.

Scheer & Hars[1992]는 기업 전체 차원에서

의 통합 데이터모델(enterprise-wide data models)의 필요성을 제시하였다. 데이터모델은 거시적 단계(macro-level), 중간 단계(medium-level), 세부단계(micro-level)모델로 나눌 수 있다. 데이터 통합 방법론도 데이터 내에서 상위 수준의 모델링 활동과 하위 수준의 모델링 활동이 통합되어 일관성을 유지할 때에 프로젝트 성공을 도출할 수 있음을 보여주고 있다. 이는 데이터 모델링 활동에 국한하여 통합의 필요성을 강조하고 있으며 이를 위한 모델링 활동이 상위 수준부터 하위 수준까지 일관성 있게 진행되어야함을 강조하고 있다.

Baumer et al.[1997]은 객체지향 프로젝트들을 대상으로 한 모델링 체계를 제시하였다. 이 체계는 어플리케이션 도메인과 비즈니스 구조간의 대응을 위해서 비즈니스 도메인, 비즈니스 섹션, 어플리케이션으로 그 범주를 구분하였고, 그들간의 의존성을 관리하고 중복되는 부분을 제거하기 위해 계층적으로 분류하였다. 비즈니스 도메인은 전체 비즈니스의 핵심개념을 포함하고, 비즈니스 섹션은 각 비즈니스 섹션을 위한 특정 클래스를 가진 체계으로 이루어져 있으며, 어플리케이션은 서로 다른 작업공간을 위한 소프트웨어로 구성된다.

Hackathorn & Karimi[1989]는 정보 시스템을 모델링하기 위한 추상화 단계로서 글로벌 엔터티 관계 모델링(global entity relation modeling), 개념적 데이터 모델링(conceptual data modeling), 프로세스 모델링(process modeling), 데이터/프로세스 통합 모델링(data/process integration modeling) 4단계로 나누었다. 이는 인지적 추상화 모델링 단계이론을 정보시스템 분석 및 디자인 단계에 활용한 연구로서 정보시스템 모델링 환경을 추상화 이론 측면에서 체계화 할 수 있는 체계를 제시하였다.

<표 2>는 객체지향 모델링 활동을 모델링 단

〈표 2〉 객체지향 모델링 활동 단계에 관한 연구 요약

저 자 [연 도]	모 델 링 단 계
Howden[1982]	<ul style="list-style-type: none"> · 요구사항 분석 단계 · 디자인 단계 · 코딩 단계 · 검증 단계
Iivari & Koskela[1987]	<ul style="list-style-type: none"> · 실용적 단계 · 입력/출력 단계 · 구축 단계
Olive[1983]	<ul style="list-style-type: none"> · 외부 단계 · 개념 단계 · 논리적 아키텍처 단계 · 물리적 단계
Essink[1986]	<ul style="list-style-type: none"> · 객체시스템 모델 · 개념적 정보시스템 모델 · 데이터시스템 모델 · 구현 모델
Cervený, Garrity & Sanders[1990]	<ul style="list-style-type: none"> · 상위 모델링 환경 <ul style="list-style-type: none"> - 사업객체 모델링 - 엔터프라이즈 모델링 - 사업프로세스 모델링 - 개념적 데이터/프로세스 모델링 · 하위 모델링 <ul style="list-style-type: none"> - 엔티티/프로세스 모델링 - 프로그래밍/테스팅
Fayad & Schmidt[1997]	모델링 단계 <ul style="list-style-type: none"> - 엔터프라이즈 어플리케이션 체계 - 미들웨어 통합 체계 - 시스템 인프라 체계
Scheer & Hars[1992]	기업 통합 데이터 모델 제시 <ul style="list-style-type: none"> - 거시적 단계 (macro-level) - 중간 단계 (medium-level) - 세부 단계 (micro-level)
Baumer et al.[1997]	객체지향 프로젝트의 모델링 단계 제시 <ul style="list-style-type: none"> - 비즈니스 도메인 - 비즈니스 섹션 - 어플리케이션
Hackathorn & Karimi[1989]	인지 이론에 근거하여 모델링 단계 도출 <ul style="list-style-type: none"> · 글로벌 엔티티 관계 모델링(global entity relation modeling) · 개념적 데이터 모델링(conceptual data modeling) · 프로세스 모델링(process modeling) · 데이터/프로세스 통합 모델링(data/process integration modeling)

계별로 제시한 기존 연구를 요약, 제시하고 있다. 각 연구는 객체지향 모델링 단계를 다양한 측면에서 제시하고 있으며 대부분의 연구가 객

체지향 모델링 활동의 모델링 단계를 상위 모델링 단계부터 하위 모델링 단계까지로 제시하고 있다.

2.3 객체지향 모델링 활동 유형에 관한 연구

전 절에서 토의된 모델링 단계에 관한 연구는 주로 시스템 모델링의 추상화 단계에 초점을 맞춘 연구였다. 추상화 단계와 더불어 모델링 활동의 주요대상은 각 단계별로 모델링 활동의 '무엇'을 지원하느냐로 나누어 볼 수 있다. 모델링 대상이 되는 객체의 속성 자체를 표현하는 활동을 지원하는 틀과 속성을 발견하고 이를 배치시키며 이를 구체화하는 객체의 프로세스를 지원하는 모델링 활동으로 나누어 볼 수 있다. 본 절에서는 객체지향 모델링 틀이 지원하는 활동 유형에 관한 연구를 정리, 분석한다.

Colter[1984]와 Pressman[1995]은 다양한 분석 및 설계 틀을 평가한 후에 틀의 특성은 모델링 대상에 대한 프로세스(process) 모델링과 표현(representation) 모델링 활동으로 구성되어 있다고 주장한다. 이러한 모델링 틀의 특성은 모델링 단계별로 구체화될 수 있다고 하였다. 객체지향 분석 및 설계 프로세스는 분석 및 설계를 위한 절차적 기법을 말한다. 객체지향 시스템 분석 및 설계에서의 표현이란 시스템의 구조, 기능, 제어를 추상화하여 표현하는 기법으로 정의하였다.

Olle et. al[1988]은 시스템 모델링 활동의 주요 대상으로 데이터(data), 프로세스(process), 행위(behavior)를 제시하고 있다. 각 주요 모델링 대상의 추상화 단계별로 모델링의 표현틀로 활용될 수 있다. 데이터와 프로세스는 전통적으로 모델링 틀에 의해 지원되는 대상이었으며 최근에 이러한 데이터나 프로세스의 시간적 역동성을 표현하기 위한 행위(behavior)에 대한 모델링 활동의 중요성이 점증되고 있다.

Essink[1986]는 시스템 이론에 기반을 둔 추상화 대상으로서 8가지 대상을 나열했다. 이는 목적구조(goal structure), 환경적 상호작용(en-

vironmental interaction), 내부 기능 구조(internal functional structure), 엔터티 구조(entity structure), 프로세스 구조(process structure), 시스템 역학(systems dynamics), 배치 관점(allocation aspect), 실현 관점(realization consideration)등을 포함한다. 이는 모델링 활동에 근거한 귀납적 모델링 활동의 주요 대상을 도출하기보다는 시스템 이론에 근거한 연역적 방법에 근거하여 모델링 활동의 주요 대상을 도출하였다고 할 수 있다.

Batra & Davis[1989]는 모델링 활동의 주요 대상으로 엔터티의 처리, 엔터티, 속성, 카테고리 사이의 관계를 제시하였다. 이들은 주로 데이터 모델링 차원에서의 모델링 대상을 강조하였다. 이러한 데이터 속성이 비교적 모델링 대상의 안정적 파라미터를 제공한다고 할 지라도 이들이 구체적으로 활용되는 프로세스 모델링에 대한 접근이 부족한 연구이다.

Henderson & Coopridge[1990]은 객체, 관계, 프로세스를 모델링의 주요 대상으로 제시하였다. 이들은 비교적 모델링 대상에 있어서의 포괄성을 강조하였다. 그러나 이러한 모델링 대상 내에서의 구체적 특성을 나열하지 못함으로써 개념적 연구의 출발점을 제시하고는 있으나 실무적 차원에서의 모델링 틀의 평가 체계를 제시하지는 못하고 있다.

Martin & McClure[1985]은 추상화 각 단계에서의 모델링 대상으로 데이터(data)와 활동(activity)를 제안했다. 이들의 연구는 모델링 대상으로서 전통적 대상이었던 데이터와 활동을 강조하여 비교적 모델링 활동을 시스템 분석 및 설계 차원에서 분석, 제시하였다. 최근에 모델링 활동이 엔터프라이즈 수준으로부터 프로그래밍 단계까지 확장되고 이를 지원하는 모델링 틀이 확산되면서 모델링 단계별 모델링 대상의 확장도 필요할 것이다.

Parson & Wand[1997]는 시스템 분석단계에서의 객체 사용에 대하여 연구했다. 그들은 객체지향 시스템 분석을 위한 함축적이고 실제적인 결과를 보여주는 존재론적(ontology)이고 인지적인(cognitive) 표현의 모델을 사용하여 객체지향 개념을 해석하였다. 그 결과 시스템 분석에서 객체 개념의 적용은 구현, 생성보다는 객체 표현(representation)함에 적합하다는 것을 보여주고 이에 근거한 시스템 분석을 위한 지침(guideline)을 제시하였다. 이는 모델을 표현하는데 있어서 캡슐화(encapsulation), 동질성(homogeneity), 특수화/상속성(specialization/inheritance), 합성(composition), 의사소통(communication)의 요소를 포함하고 있다. 이는 객체의 프로세스에 대한 모델링 보다는 표현에 치우친 모델링 활동을 제시하고 있다.

Shelton[1996]은 객체지향 모델링 활동 중 상위 활동인 비즈니스 모델링(business modeling)에 대하여 연구하였다. 상위 모델링 활동은 시스템 활용분야에 대한 이해와 의사소통을 촉진 시킴으로서 사업모델 수준의 프로세스나 데이터 표현을 촉진시킨다. 이는 지속적인 개선을 통해 특정 타겟 비즈니스에 적합한 표현으로 변형될 수 있다. 비즈니스 객체는 개념, 규칙, 플레이어, 자원, 장소나 생각을 표현하는 비즈니스 엔티티(entity)와 비즈니스 활동(객체간의 상호 작용)을 표현한 프로세스, 그리고 사건이나 시간에 따라 일어나는 객체를 표현하는 이벤트로 구성될 수 있다. 비즈니스 모델링 활동은 상위 단계 모델링 활동으로서 비즈니스 아키텍처를 논리적 표현하는 모델링 프로세스라 할 수 있다.

〈표 3〉 객체지향 모델링 활동 유형에 관한 연구 요약

저 사 [연 도]	연 구 내 용
Colter[1984], Pressman[1987]	모델링 틀은 프로세스와 표현을 지원한다.
Olle et. al[1988]	모델링의 주요 대상은 데이터, 프로세스, 행위로 구분
Essink[1986]	목적구조, 환경적 상호작용, 내부기능구조, 엔티티 구조, 프로세스 구조, 시스템 역할, 배치관점, 실현 관점
Batra & Davis[1989]	엔티티의 처리 엔티티, 속성, 카테고리 사이의 관계
Henderson & Cooperider[1990]	객체, 관계, 프로세스
Martin & McClure[1985]	데 이 터 활 동
Parson & Wand[1997]	캡슐화 동질성 특수화/상속화 합성 의사소통
Shelton[1996]	비즈니스 엔티티 : 개념, 규칙, 플레이어, 자원, 장소를 표현 프로세스 : 객체간의 상호 작용 표현 이벤트 : 객체의 시간에 따른 변화를 표현
Monarchi & Puhr[1992]	지금까지의 객체지향 방법론 연구 정리, 분류, 구체화. 모델링 활동을 프로세스와 표현으로 나누고 각 활동에서의 구체적 활동의 세부 항목을 제시 프로세스 : 식별 활동, 클래스 배치, 명세화, 클래스 최적화 활동 표현 : 정적 뷰, 동적 뷰, 제약사항 표현

Monarchi & Puhr[1992]는 지금 까지 객체지향 모델링 툴의 모델링 활동을 정리하여 프로세스(process)와 표현(representation) 차원에서 분류하고 이를 더욱 구체화하였다. 객체지향 모델링의 단계를 분석, 디자인, 표현, 복잡성 관리 활동으로 구분하고 각 모델링 활동 내에서 표현 체계를 구체화하였다. 또한 이러한 체계를 기반으로 하여 현재까지 활용되어온 객체지향 모델링 툴의 지원 여부를 확인하였다. 이러한 관찰을 통해서 현 객체지향 분석 및 설계 관련 연구가 클래스, 속성, 행위의 식별, 메소드의 배치(placing method), 일반화와 집단화의 식별과 표현, 정적 뷰(view)에서의 표현부분에서의 많은 지원을 하였지만, 인터페이스, 어플리케이션, 시스템 클래스 식별, 속성, 관계성(행위)의 클래스 결정, 클래스 배치, 다른 종류의 관계의 식별과 표현 관계를 위한 일관성 있고 올바른 의미 관리, 동적 뷰의 표현, 정적 모델과 동적 모델의 통합, 추상화 단계의 일관성 유지 관리에서는 취약점을 보임을 제시하였다. 이 연구는 객체지향 모델링 툴을 평가하는 최초의 연구라는 점에서 그 의의가 있으며 실무 차원에서 평가 체계를 제시하였다는 점에서도 공헌을 찾을 수 있다. 그러나 이론적인 측면에서 모델링 활동을 단지 시스템 분석 활동과 디자인 활동 차원에 제한하여 평가하였다는 문제점을 지니고 있다. 모델링 활동은 이 이외에도 기업 전체 차원에서의 모델링 활동의 중요성이 부각되어야 하며 또한 디자인 활동도 구현 단계의 모델링 활동 즉, 프로그래밍 활동과 연관되어야 할 것이다.

3. 객체지향 모델링 활동 평가 차원

객체지향 모델링 기법은 전통적 구조적 모델링 기법보다 추상화 이론에 근거한 모델링 방법

으로서 각 추상화 단계별 대상이 객체에 한정되기 때문에 모델링 측면에서 상위부터 하위 단계까지 일관성을 유지할 수 있을 뿐만 아니라 모델링 대상에 대한 표현이나 모델링 프로세스가 투명화될 수 있다. 객체지향 모델링 활동에 대한 연구를 분석해 볼 때에 모델링 활동 측면에서 두 가지 연구 흐름으로 대분될 수 있음을 알 수 있다. 모델링 활동 단계(levels)를 상위부터 하위 단계로 모델의 추상화 단계에 관한 연구 흐름과 각 모델링 활동 단계에서 모델링 활동 유형(type), 즉 프로세스(process)와 표현(representation) 활동에 관한 연구 흐름이다.

Berzin, Gary & Naumann[1986]은 상위 모델링 단계에서 모델화된 프로세스나 표현활동은 하위 모델링 단계에서 대상 객체를 구체화하게 되고 이러한 활동은 프로세스 활동과 표현 활동으로 나누어 볼 수 있다고 한다. 예를 들어 기능 명세 도출이 주요 목적인 시스템 분석 및 설계 단계에서의 모델링 활동은 대상 모델의 구체적인 내부 활동 및 작업(task)에 대한 모델을 숨기는 대신 전체 기능들 사이의 프로세스나 객체 표현에 모델링 활동을 집중하게 된다. 각 추상화 단계에서 객체지향 시스템 개발 모델링 대상은 프로세스나 표현활동을 통하여 구체화된다고 주장한다. 이 연구는 개념적 차원에서 객체지향 모델링 툴의 평가 차원을 제시하여 주었으며 특히 추상화 차원에서의 평가 차원과 모델링 활동 유형을 결합한 체계를 도출할 수 있는 이론적 근거를 제시하고 있다.

본 연구에서 객체지향 모델링 툴을 평가하기 위한 체계는 두 개의 차원으로 구성된다. 모델링 단계는 객체지향 엔터프라이즈 모델링 단계, 객체지향 시스템 분석 단계, 객체지향 시스템설계단계, 객체지향 시스템구현단계로 구성된다. 각 모델링 단계 내에서 모델링 활동의 유형은

객체지향 프로세스(process)모델링 활동과 객체지향 모델 표현(representation) 활동으로 모델링 활동을 구분된다.

객체지향 프로세스(process) 활동은 해당 문제영역에 대한 분석 절차(procedure)를 말한다. 모델링 프로세스란 객체지향분석 및 설계를 수행하기 위하여 사용되는 절차적 방법(procedural methods)을 말한다. 이는 클래스, 속성, 행위와 같은 모델링 대상에 대한 시별(identification), 배치(placement), 명세화(specification), 클래스 최적화 활동을 포함한다[Monarch & Puhr, 1992]. 이는 객체지향 분석 및 설계를 위한 다이어그램이나 표현법을 포함하지 않는다. 이는 모델링 툴나 방법론을 구조화하는 방법과 사용자가 하나의 모델을 만들 때, 어떻게, 언제, 무엇을, 왜라는 절차적 방법을 제시한다. 여기에서 강조되는 활동은 “어떻게” 문제를 분석하느냐에 대한 절차이지 문제를 표현에는 그 중요성이 주어지지 않는다.

그에 반해 객체지향분석에서 표현(representation) 활동은 모델을 표현하는 방법에 그 중점을 맞춘다. 객체지향 표현(representation)이란 객체지향분석 및 설계 활동의 결과를 표현하기 위한 다이어그램이나 도식적 표현법(graphical notation) 활동을 말한다. 여기에서는 디자인이나 분석 활동을 표현(representation)에 그 활동의 초점이 있지 어떻게(how to) 디자인이나 분석 내용을 도출하느냐 에는 그 활동의 관심이 없다. 이는 객체에 대한 정적 뷰(static view), 동적 뷰(dynamic view), 제약 사항의 표현에 그 활동의 주요 목표가 제한된다.

실제 객체지향 방법론, 모델링 툴, 프로그래밍언어는 그 특성에 따라서 프로세스 활동과 표현 활동 모두를 지원하는 경우도 있고 그 중에 어느 하나만을 지원하는 경우도 있다. 또한 프로젝트 수행시 같은 모델링 툴을 활용할 지라도

어떠한 차원에 활용하느냐는 프로젝트 환경, 프로젝트 관리자의 프로젝트 특성에 대한 이해, 프로젝트 요원들의 모델링 툴에 대한 학습 정도에 따라서 달라질 수 있다.

본 연구는 모델링 활동을 지원하는 모델링 툴을 평가하기 위한 차원을 2개의 차원으로 제시한다. 한 차원은 상위 모델링 단계로부터 하위 모델링 단계로 구성되며 이는 구체적으로 엔터프라이즈 모델링 단계, 시스템분석 단계, 시스템디자인 단계, 구현 단계로 구성된다. 다른 한 차원은 각 모델링 활동 유형 차원으로 모델링 대상을 모델링 하기 위한 프로세스 활동과 표현활동으로 나누어진다. 이는 프로세스는 클래스 확인, 배치, 명세화, 클래스 최적화 활동으로 구성되며, 표현 활동은 정적 표현활동, 동적 표현활동, 제약 사항 표현으로 구성되었다. 위에서 검토된 추상화 모델링 단계에 관한 연구결과를 개념적으로 도표화하여 시스템 모델링 개발 모두와 관련시켜서 제시하면 아래(그림 1)과 같다.

객체지향 모델링 활동에 활용된 툴은 아래 그림에서 제시된 모델링의 두 개 차원에 대하여 평가할 수 있으며 이러한 모델링 활동의 지원 활동에 대한 확인과 이로 인한 시스템 성공에 대한 공헌 여부는 프로젝트 관리 활동의 주요한 측정치로서 향후 프로젝트 성공에 지대한 영향을 미칠 수 있다.

4. 모델링 차원 구체화 및 타당성 평가

4.1 모델링 툴(tools) 평가 도구(instrument)

(그림 1)에서 제시된 객체지향 모델링 툴 평가 차원은 객체지향 모델링 툴(tools), 객체지향 프로그래밍언어 등 엔터프라이즈 모델링 단계부터 프로그래밍 단계까지 모델링 활동을 지원

모델링 활동 유형	모델링 활동 단계			
	엔터프라이즈 단계	시스템분석 단계	시스템디자인 단계	구현 단계
프로세스(process) 활동 - 식별 활동 - 배치 활동 - 명세화 활동 - 최적화 활동				
표현(representation) 활동 - 정적 뷰(view) - 동적 뷰 - 제약사항				

(그림 1) 객체지향 모델링 툴 평가 체계

하는 객체지향 시스템 모델링 툴을 총체적으로 평가할 수 있는 도구(instrument) 도출하기 위한 기반을 제공한다. 모델링 활동 유형을 구성하는 세부 활동에 대하여는 Monarch & Puhr [1992]이 제시한 모델링 활동을 대상으로하여 구성하였다. 도구를 활용하여 전문가의 타당성 검증은 거처기 이전에 모델링 툴의 모델링 체계를 검토하기 위해서 상업적 툴¹⁾ 판매업자가 제시하는 대표적 모델링 툴의 적용분야를 조사하였다. 먼저 모델링 활동 단계를 구성하기 위해서 상업적 툴이 지원하는 모델링 단계를 검토한 결과 이론적 측면에서 검토된 4개의 차원 즉, 엔터프라이즈 모델링 단계, 시스템분석 단계, 시스템 디자인단계, 시스템 구현 단계의 구성 타당성을 확인하였다. 모델링 활동 유형에 대하여는 프로세스 모델링 활동과 표현 모델링 활동 내에서의 상세한 모델링 활동을 구성하기 위해서는 Monarch & Puhr[1992]이 제시한 세부 모델링 활동을 중심으로 기존 툴이 지원하는 대표적 활동을 선별하여 포함하였다. 이 모델링 활동 평가 도구(instrument)의 현실적 타당성을

검증하기 위해 먼저 시스템통합 업체의 프로젝트 관리자 5명을 순차적으로 인터뷰하여 차원의 적정성, 세부 모델링 활동 유형의 차별성 등을 대상으로 타당성을 검토하였다.

평가를 위한 도구를 선정하기 위해서는 먼저 기존에 객체지향 툴에 대한 논문, 서적, 상업적 툴을 광범위하게 조사하였다. 이를 대상으로 하여 가장 광범위하게 활용되는 툴을 선택하기 위해서 객체지향 연구전문가 2인, 객체지향프로젝트 경험 프로젝트 관리자 2명, 객체지향 분석 및 설계 툴 전문 컨설턴트 2명으로 연구대상 툴 선정팀을 구성하여 최종적으로 18개 툴을 선정하였다(선정된 툴과 이에 대한 설명은 부록 참조).

평가도구의 타당성 및 주요 모델링 툴 평가를 위하여 전문가를 선별하였다. 본 연구가 이론의 검증보다는 이론 개발을 위한 연구체계/framework)를 제시하는 데에 있기에 전문가 채택에 있어서 연구의 타당성(relevance)을 높이기 위한 소수의 전문가를 선별하여 포함시켰다. 전문가를 구성함에 있어서 객체지향 모델링 분야의 연구 전문가 2명, 객체지향 프로젝트 5회 이상 경험 프로젝트 관리자 2명, 객체지향 모델링 컨설턴트 2명을 포함시킴으로서 전문가 사이의 다양한 견해를 수용하고 견해의 편견(bias)을

1) 사이트

http : \\www.rational.co.kr

http : \\www.oracle.com

http : \\www.plasticsoftware.com

http : \\www.nextech.co.kr

최소화하였다. 이를 반영하여 아래 (그림 2)에서 제시하는 구체적 평가도구(instrument)를 구성하였다.

4.2 모델링 툴 활용 분야 평가

(그림 2)에서 제시된 평가도구를 활용하여 주요 객체지향 모델링 툴을 평가하였다. 평가 전문가는 평가 도구 도출한 전문가를 중심으로 아래와 같은 평가지침과 과정을 거쳐서 모델링 도구별로 활용된 분야를 도출하였다. 평가에 포함된 객체지향 다이어그램에 대한 자세한 설명은 부록에 정리되어 있다.²⁾

평가를 위한 지침으로는 먼저 본 연구에서 제시된 모델링 활동과 모델링 활동 유형에 대한 의미상의 혼란을 최소화하기 위해서 주 연구자가 평가 이전에 이에 대한 설명을 하여 평가에 있어서의 의미상 혼란을 최소화하였다. 다음으로 본 연구에서 평가를 위하여 채택된 도구(부록 참조)에 대하여 요약 설명지를 배포하고 최종적으로 이해도를 파악한 후에 난상토론 합의(brainstorming consensus) 절차를 설명하였다. 난상토론을 위한 지침으로는 연구자가 평가 대상 툴에 대하여 간략한 설명을 한 이후에 이에 대하여 사실(fact)에 근거한 자유로운 의견 표명을 주문하였다.

평가하기 위한 과정은 본 연구의 목적이 객체지향 모델링 툴 평가 체계의 도출과 이에 대한 타당성 평가에 있기에 전문가의 전문적 판단능력에 근거한 난상토론 합의(brainstorming consensus)에 의해 모델링 활동 적용 분야를 결정하는 방법을 선택하였다. 평가과정에서 주요한 사실(significant fact) 즉 모델링 툴의 모델링 활동 적용분야를 발견함[Kuhn, 1996]에 평가의

목표를 두었다. 합의를 도출하기 위해서는 먼저 개별 전문가별 적용분야를 제시하고 이를 근거로 하여 전문가들 공히 동의하는 적용분야를 합의에 의해 도출하게 함으로써 일부 전문가의 특이한 결정에 의한 바이어스(bias)를 줄이도록 하였다.

객체지향 시스템 모델링에서 활용되는 다이어그램 툴들이 어떤 분야의 모델링에 활용될 수 있는지를 위의 지침과 과정을 거쳐서 평가한 결과는 (그림 3)부터 (그림 6)에서 정리되었다. 엔터프라이즈 모델링 단계에서는 사용사례 다이어그램, OverView 다이어그램, 객체흐름다이어그램이 활용되고 있으며 이러한 다이어그램도 주로 객체의 표현활동 중심으로 활용되고 있다. 시스템 분석 및 설계 단계에서는 실행다이어그램을 제외한 다이어그램 툴이 전반적으로 활용되고 있으나 이 단계에서도 주로 프로세스 모델링 활동보다는 표현 활동에 주로 활용되고 있는 것으로 나타났다. 시스템 실행단계에서는 실행다이어그램이 활용되고 있으며 이 툴은 프로세스와 표현 모델링 활동에 공히 활용되고 있는 것으로 확인되었다.

객체지향 모델링 툴의 활용 평가 결과 엔터프라이즈 모델링 단계부터 실행 단계까지 툴 활용 측면에서의 일관성 있는 모델링 도구 활용은 아직은 확산되지 못한 상황이며 프로세스 모델링 활동은 주로 시스템 실행 단계에서 수행되고 있음을 알 수 있다. 객체지향시스템 모델링 툴이 확산 단계에 있으나 아직도 모델링 단계나 모델링 분야를 통합적으로 지원하는 환경을 제공하지는 못하고 있는 것으로 나타났으며 이러한 모델링의 절차적 일관성이나 통합성은 객체지향 방법론이나 툴에 의한 모델링 일관성 검증보다는 인지적 모델링에 의해 지원되어야 함을 추론할 수 있다[김진우, 1996].

2) 기존에 객체지향 모델링 실무에서 자주 사용되는 툴을 기존 서적과 전문가의 검토를 통하여 선별하였다.

모델링 활동 유형	모델링 활동 단계			
	엔터프라이즈	시스템 분석	시스템 디자인	시스템 구현
프로세스(PROCESS) 활동				
• 식별(Identification of)				
클래스(Semantic class)				
속성(Attributes)				
행위(Behavior or method)				
객체간의 관계(Relationships) :				
일반화(Generalization)				
집단화(Aggregation)				
기타(Other) ³⁾				
인터페이스 클래스 ⁴⁾ (Interface classes)				
어플리케이션 클래스 ⁵⁾ (Application classes)				
베이스/유틸리티 클래스 ⁶⁾				
• 배치(Placement of)				
클래스(Classess)				
속성(Attributes)				
행위(Behavior or method)				
• 명세(Specification)				
동적행위(Dynamic behavior) (예, 메세지전달)				
• 클래스 최적화 ⁷⁾				
표현(REPRESENTATIONS) 활동				
• 정적 뷰(Static View)				
객체(Objects)				
속성(Attributes)				
행위(Behavior or method)				
관계(Relationships) :				
일반화(Generalization)				
집단화(Aggregation)				
기타(Other)				
• 동적 뷰(Dynamic View)				
커뮤니케이션(Communication)				
제어/시간(Control/Timing)				
• 제약사항(Constraints)				
구조적(Structure)				
동적행위(Dynamic behavior)				

(그림 2) 객체 지향 모델링 툴 평가 체계

3) i.e., 상속성(Inheritance), 조합(association)
 4) 인터페이스 클래스(Interface classes) : UI(User Interface). 의미화 된 객체의 사용자 뷰를 나타낸다.
 5) 어플리케이션 클래스(Application classes) : 시스템을 조절하거나, 감독할 수 있는 객체. 절차적 언어에서 주 모듈에 상응한다.
 6) 베이스/유틸리티 클래스(Base/Utility classes) : application

과 독립적인 요소로서, 베이스 클래스의 예는 string, array 와 number이고, 유틸리티 클래스의 예는 Smalltalk의 디버거 윈도우(debugger window)이다.
 7) 클래스 최적화(Optimization of classes) : 공통된 행위(behavior or method)와 속성(attributes)을 추상화하거나, 클래스를 명확하게 하고, 디자인을 올바르게 하는 것과 관련된다.

	사용 사례	CRC	클래스	Fern	Box	객체관계	Composed-of	Overview	순차	협력	상태	활동	상태변이	이벤트	객체흐름	컴포넌트	전개	실행
프로세스(PROCESS) 활동																		
• 식별(Identification of)																		
클래스(Semantic class)																		
속성(Attributes)																		
행위(Behavior or method)																		
객체간의 관계(Relationships) :																		
일반화(Generalization)																		
집단화(Aggregation)																		
기타(Other)																		
인터페이스 클래스(Interface classes)																		
어플리케이션 클래스(Application classes)																		
베이스/유틸리티 클래스(Base/utility classes)																		
• 배치(Placement of)																		
클래스(Classess)																		
속성(Attributes)																		
행위(Behavior or method)																		
• 명세(Specification)																		
동적행위(Dynamic behavior) (i.e., message passing)																		
• 클래스 최적화(Optimization of classes)																		
표현(REPRESENTATIONS) 활동																		
• 정적 뷰(Static View)																		
객체(Objects)	X							X							X			
속성(Attributes)																		
행위(Behavior or method)	X																	
관계(Relationships) :																		
일반화(Generalization)	X							X							X			
집단화(Aggregation)	X							X										
기타(Other)	X							X							X			
• 동적 뷰(Dynamic View)																		
커뮤니케이션(Communication)	X							X										
제어/시간(Control/Timing)	X																	
• 제약사항(Constraints)																		
구조적(On structure)	X							X										
동적행위(On dynamic behavior)	X														X			

* 모델링 틀이 해당 모델링 활용 유형과 관련이 있을 경우 ×로 표시하였음.

(그림 3) 객체 지향 엔터프라이즈 모델링 단계 다이어그램 활용평가*

	사용 사례	CRC	클래스	Fern	Box	객체관계	Composed-of	Overview	순차	협력	상태	활동	상대변이	이벤트	객체호름	컴포넌트	전개	실행
프로세스(PROCESS) 활동																		
● 식별(Identification of)																		
클래스(Semantic class)																		
속성(Attributes)																		
행위(Behavior or method)																		
객체간의 관계 (Relationships) :																		
일반화(Generalization)																		
집단화(Aggregation)																		
기타(Other)																		
● 배치(Placement of)																		
클래스(Classess)																		
속성(Attributes)																		
행위(Behavior or method)																		
● 명세(Specification)																		
동적행위(Dynamic behavior) (i.e., message passing)																		
표현(REPRESENTATIONS) 활동																		
● 정적 뷰(Static View)																		
객체(Objects)	X	X	X	X	X	X	X	X	X	X	X				X	X	X	
속성(Attributes)		X	X															
행위(Behavior or method)	X	X	X						X	X	X	X	X	X	X	X	X	
관계(Relationships) :																		
일반화(Generalization)	X		X	X	X	X	X	X							X			
집단화(Aggregation)			X	x	x	X	x	X										
기타(Other)	X		X			X	x	X							x	X	X	
● 동적 뷰(Dynamic View)																		
커뮤니케이션(Communication)	X		X					X	X	X	X	X		X				
제어/시간(Control/Timing)	X		X						X	X	X	X	X	X				
● 제약사항(Constraints)																		
구조적(On structure)	X		X	X	X	X	X	X								X	X	
동적행위(On dynamic behavior)	X		X						X	X	X	X	X	X	X			

(그림 4) 객체 지향 시스템 분석 단계 다이어그램 활용평가

	사용 사례	CRC	클래스	Fern	Box	객체관계	Composed-of	Overview	순차	협력	상태	활동	상태변이	이벤트	객체흐름	컴포넌트	전개	실행
프로세스(PROCESS) 활동																		
● 식별(Identification of)																		
클래스(Semantic class)																		
속성(Attributes)																		
행위(Behavior or method)																		
객체간의 관계(Relationships) :																		
일반화(Generalization)																		
집단화(Aggregation)																		
기타(Other)																		
인터페이스 클래스(Interface classes)																		
어플리케이션 클래스(Application classes)																		
베이스/유틸리티 클래스(Base/utility classes)																		
● 배치(Placement of)																		
클래스(Class)																		
속성(Attributes)																		
행위(Behavior or method)																		
● 명세(Specification)																		
동적행위(Dynamic behavior (i.e., message passing))																		
● 클래스 최적화(Optimization of classes)																		
표현(REPRESENTATIONS) 활동																		
● 정적 분(Static View)																		
객체(Objects)	X	X	X	X	X	X	X	X	X	X	X				X	X	X	
속성(Attributes)		X	X															
행위(Behavior or method)	X	X	X						X	X	X	X	X	X	X	X	X	
관계(Relationships) :																		
일반화(Generalization)	X		X	X	X	X	X	X							X			
집단화(Aggregation)			X	X	X	X	X	X										
기타(Other)	X		X			X	X	X							X	X	X	
● 동적 분(Dynamic View)																		
커뮤니케이션(Communication)	X		X					X	X	X	X	X		X				
제어/시각(Control/Timing)	X		X						X	X	X	X	X	X				
● 제약사항(Constraints)																		
구조적(On structure)	X		X	X	X	X	X	X								X	X	
동적행위(On dynamic behavior)	X		X						X	X	X	X	X	X	X			

(그림 5) 객체 지향 시스템 디자인 단계 다이어그램 활용평가

	사용 사례	C R C	클 래 스	F e r n	B o x	객 체 관 계	Com posed - of	Over view	순 차	협 력	상 태	활 동	상 태 변 이	이 벤 트	객 체 호 름	컴 포 넌 트	전 개	실 행
프로세스(PROCESS) 활동																		
● 식 별(Identification of)																		
클래스(Semantic class)																		X
속 성(Attributes)																		
행 위(Behavior or method)																		X
객체간의 관계(Relationships) :																		
일반화(Generalization)																		X
집단화(Aggregation)																		
기 타(Other)																		
인터페이스 클래스 (Interface classes)																		
어플리케이션 클래스 (Application classes)																		
베이스/유틸리티 클래스 (Base/utility classes)																		
● 배 치(Placement of)																		
클래스(Class)																		
속 성(Attributes)																		
행 위(Behavior or method)																		X
명 세(Specification)																		
동적행위(Dynamic behavior) (i.e., message passing)																		X
● 클래스 최적화 (Optimization of classes)																		
표현(REPRESENTATIONS) 활동																		
● 정적 뷰(Static View)																		
객 체(Objects)																		X
속 성(Attributes)																		X
행 위(Behavior or method)																		X
관 계(Relationships) :																		
일반화(Generalization)																		X
집단화(Aggregation)																		
기 타(Other)																		X
● 동적 뷰(Dynamic View)																		
커뮤니케이션(Communication)																		X
제어/시간(Control/Timing)																		X
● 제약사항(Constraints)																		
구 조 적(On structure)																		X
동적행위(On dynamic behavior)																		X

(그림 6) 객체 지향 구현 단계 다이어그램 활용평가

5. 연구결과, 연구의 한계, 향후 연구 방향

객체지향 모델링 틀이 객체지향시스템 개발 프로젝트에 활용되기 시작하였으나 모델링 환경이 모델링 차원을 평가하기 위한 개념적 틀이나 평가에 대한 연구는 매우 미진한 상태이다. 본 연구는 기존에 이루어진 모델링에 관한 연구를 기반으로 하여 객체지향 모델링 활동에 대한 개념적 틀을 제시하고 이러한 틀에서 제시된 구체적 모델링 차원을 전문가 중심으로 검토해 봄으로써 향후 모델링 분야의 이론적 연구를 위한 틀을 제시하였다. Kuhn[1996]의 과학적 탐구과정에서 '주요한 사실'을 발견하고자하는 것이 본 연구의 주요 목표이자 공헌이라 할 수 있다.

연구 결과 엔터프라이즈 모델링 단계에서는 모델링 다이어그램 모두 프로세스 모델링 활동에 활용되지 못하고 단지 표현 활동에 활용되고 있음을 알 수 있다. 특히 사용사례, 오버뷰(overview), 객체흐름 다이어그램이 주로 활용되고 있음을 알 수 있다. 상위 단계에서의 모델링 활동에 활용되는 다이어그램이 객체의 확인, 배치, 명세화, 최적화 활동에 활용되고 있지 못하다. 이는 아직도 객체지향 모델링 활동이 비즈니스 기능이나 엔터프라이즈 단계에서의 객체를 도출하고 이를 체계화할 수 있는 표준 프로세스나 컴포넌트화가 미진함을 반증하고 있다 할 수 있다.

시스템 분석 단계에서는 실행다이어그램을 제외한 연구 대상 모든 틀이 모델의 표현 활동에 광범위하게 활용되고 있음을 보여주고 있다. 그러나 이 단계에서도 프로세스 모델링 활동을 지원하지는 못하는 것으로 나타나고 있다. 디자인 단계에서도 분석 단계와 같은 패턴을 보여주고 있다. 이는 객체에 대한 표현(representation)에 대하여는 모델링 규칙이 형성, 지원되고 있으나 객체 자체의 확인이나, 배치 즉 프로세스

모델링 활동은 틀에 의해서 충분히 지원되지 못하고 있음을 알 수 있다. 이와 같은 연구 결과는 김진우, 한형미[1996]의 연구 결과와 같이 객체지향 모델링 활동에서 모델링 전문가의 인지적 통합에 의한 프로세스 모델링 활동의 중요성을 보여준다. 이는 프로세스 모델링 활동은 표현 활동과는 달리 인지적 통합과 유연성이 더욱 요구되는 모델링 활동으로서 틀에 의해서라기 보다는 인간의 인지적 과정에 의해 수행되기에 적합한 활동이라 할 수 있다.

구현 단계에서는 프로그램코드로 변환시킬 수 있는 실행 다이어그램만이 프로세스와 표현 모델링 활동을 지원하고 있음을 보여주고 있다. 기존의 여타 모델링 다이어그램은 실행 단계에 모델링 활동에 활용되지 못하고 있음을 알 수 있다. 실행다이어그램이 엔터프라이즈 모델링 단계, 분석, 디자인 단계에서 표현된 객체를 대상으로 실행 단계에서 객체 프로세스를 모델링 할 수 있는 환경을 제공한다는 점이다. 이는 객체표현에 있어서 일관성은 비교적 확보할 수 있는 모델링 환경을 구성할 수 있으나 프로세스 모델링은 엔터프라이즈 단계부터 실행 단계까지 일관성 있게 구현할 수 있는 환경을 구성하기는 비교적 힘들다는 점이다.

연구결과 기존 객체지향 모델링 틀은 주로 객체의 표현(representation)위주로 활용되고 있음을 알 수 있으며 프로세스 모델링 지원에는 거의 활용되지 못하고 있음을 알 수 있다. 프로세스 모델링 즉 객체의 확인 및 배치와 같은 활동은 전문가의 모델링 경험이나 전문성에 의해서 수행되어야 함을 알 수 있다. 기업 엔터프라이즈 또는 업무 기능 분야의 업무 프로세스의 객체화가 진행되지 못한 상태에서 모델링 틀에 의한 객체화가 불가능하다는 점을 시사한다.

객체지향 모델링이 확산되고 이를 지원하는 모델링 틀의 활용이 보편화되고 있음에도 불구하고

하고 이러한 모델링 활동이 시스템 개발의 생산성이나 최종시스템 개발 성공에 미치는 영향에 대하여는 일관성 있는 연구 결과를 나타내고 있지 못하는 이유도 이와 같은 모델링 활동의 통합이나 일관성 있는 모델링 활동 지원이 충분히 모델링 틀에 의해 지원되지 못함에서 그 연유를 찾을 수 있다. 즉, 아직도 객체지향 모델링 활동 시스템분석 및 설계분야의 표현 활동에 집중되어 수행되고 있는 상황에서 객체지향 모델링 틀과 시스템 개발 성공과의 인과 관계는 그 강도가 약할 수 밖에 없을 수 없다.

실무에서도 현재 객체지향 모델링 틀은 모델링 단계별 또는 활동별로 체계적으로 활용되기 보다는 모델링 단계별로 단절된 상황에서의 모델링 활동이 진행되고 있는 것이 현실이다. 부분적으로는 기존에 활용된 구조적 방법론을 활용하여 상위 모델링을 수행하고 단순히 객체지향 프로그래밍 언어를 활용하여 구현하는 경우가 많다. 객체지향 모델링 활동이 아직 성숙되지 못한 상황에서 이의 활용이 시스템 개발 성공에 미치는 영향은 그리 클 수 없음을 예측할 수 있다. 현재 객체지향 시스템 모델링 활동은 대다수 프로젝트가 객체지향 프로그램 언어에 집중되어 수행되고 있는 상황이며 이는 객체지향 모델링 활동의 상위 활동과 하위 활동의 단절이 결국 시스템 성공 및 생산성 향상에 유의적인 결과를 가져오지 못할 위험 요인으로 대두될 수 있음을 의미할 수도 있다.

본 연구의 공헌은 이론적 측면과 실무적 적용 측면에서 살펴 볼 수 있다. 이론적 측면에서는 기존의 객체지향 시스템개발 모델링 체계에 대한 개념 및 체계를 제시하고 이를 측정할 수 있는 도구(instrument)를 제시하였다는 점이다. 이를 객체지향 모델링 환경에 적용함으로써 객체지향 모델링 틀이 지원하는 모델링 활동이 시스템개발 성공에 기여하는 정도를 측정할 수 있

는 이론적 기반 및 설명의 토대로 활용할 수 있을 것이다. 기존의 연구는 시스템 모델링 활동의 개념화 및 측정방법에 대한 이론적 근거가 미약함으로 인해서 실증적 연구 결과의 해석을 불가능하거나 어렵게 하였다. 또한 기존의 시스템 성공에 관한 연구가 시스템성공에 미치는 영향요인을 선택하는 과정에서 모델링 활동의 중요성을 인정하면서도 이 변수의 개념화 및 측정방법에 대한 연구가 미비하였음을 감안할 때 본 연구의 결과는 시스템 성공의 주요한 요인으로서 시스템개발 모델링 활동을 구체화하여 성공평가 모델에 포함시킴으로서 이 분야의 연구에 공헌하게 될 것이다.

실무적 측면에서는 본 연구에서 제시된 시스템 모델링 활동 체계 및 측정 도구는 다양한 모델링 틀을 선택, 활용해야하는 시스템 개발 프로젝트 관리자에게 관리 틀을 제공해 줌으로서 프로젝트의 성격에 따른 객체지향 모델링 환경의 구축 즉, 객체지향 설계 및 분석 틀의 선택을 가능케 할 것이다. 또한 시스템개발 모델링 활동의 구체적 측정방법을 제공함으로써 모델링 활동의 심도(intensity)에 대한 측정치(metrics)관리를 가능케 함으로써 다양한 개발프로젝트의 모델링 틀의 활용 정도를 벤치마킹할 수 있다.

본 연구의 주된 목적이 모델링 활동에 대한 개념적 틀을 제시하는 데 있음에도 불구하고 이러한 틀의 현실적 타당성을 확인하는 과정에서 활용된 방법론은 이론과 현실을 대조하는 연구 방법보다는 과학적 연구의 초창기에 활용되는 주요한 사실을 결정(determination of significant facts)하는 데 중점[Kuhn, 1996]을 두었기에 일반적 타당성을 확보하는 데 어려움을 가지고 있다. 향후 연구에서는 본 연구에서 제시된 모델링 활동에 대한 측정 도구를 다양한 객체지향 프로젝트의 모델링 활동 정도를 측정하고 이

를 시스템 개발 성공에 영향을 미치는 여타요인(방법론, 프로젝트 팀 구조, 의사소통 구조 등)과 연관시키고 이들이 결과적으로 영향을 미치는 시스템 성공 변수와 연관시킴으로써 모델링 활동이 가져올 수 있는 직접적 영향에 대한 실증적 연구가 필요할 것이다.

참 고 문 헌

- [1] 김진우, 한형미, "객체지향방법론을 이용한 프로세스 모델링에 대한 연구-복수 모형의 인지적 통합과정을 중심으로", 「경영정보학연구」, 1996, 제6권 2호, pp.19-52.
- [2] 박광호, "객체 지향 정보시스템의 테스트를 위한 확장된 유스케이스의 사용과 계층적 상태 기반 테스트 방법", 「정보기술과 데이터 베이스 저널」, 1999, 제6권, 제2호, pp. 29-43.
- [3] Batra, D. and J.G. Davis, "A Study of Conceptual Data Modeling in Database Design : Similarities and Differences between Expert and Novice Designer," *Proceedings of the 10th international Conference on Information Systems*, edited by DeGross Janice, Henderson John and Konynski, Bern, 1989, pp.90-99.
- [4] Baumer, Dirk, Gryczan, Guido, Knoll, Rolf, Lilienthal, Carol, Lilienthal, Dirk, and Zullighoven, Heinz, "dimension Development for Large Systems," *Communications of the ACM*, October, 1997, Vol.40, No.10, pp.52-59.
- [5] Berzin, Valdis, Gary, Michael, and Naumann, David, "Abstraction-based Software Development," *Communications of the ACM*, May 1986, Vol.29, No.5, pp.402-415.
- [6] Beyer, Hugh R., K. Hollzblatt, "Apprenticing with the Customer," *Commnuicaions of the ACM*, May, 1995, Vol.38. No.5, pp.45-52.
- [7] Bohrer, K., V. Johnson, A. Nilsson, and B. Rubin, "Business Process Components for Distributed Object Applications," *Communications of the ACM*, June 1998, Vol. 41, No.6, pp.43-48.
- [8] Butler, Keith A., C. Esposito and R. Hebron, "Connecting the Design of Software to the Design of Work," *Commnuicaions of the ACM*, January, 1999, Vol.42. No.1, pp.38-46.
- [9] Cerveney, Robert P., E.J. Garrity and G.L. Sanders, "A Problem-solving Perspective on Systems Requirement," *Journal of Management Information Systems*, 6, 4(1990), 103-122.
- [10] Colter, M.A., "A comprehensive examination of systems analysis technique," *MIS Quarterly*, Mar. 1984, pp.51-66.
- [11] Ericsson, Maria and Jacobson, Agneta, *The Object Advantage : Business Process Reengineering with Object Technology*, 1995, Addison-Wesley Publishing Company.
- [12] Essink, Leo J.B., "A Modelling Approach to Information System Development," *Information Systems Design Methodologies : Improving the Practice*, edited by T.W. Olle, H.G. Sol and A.A. Verrijin-Stuart, Elsevier Science Publishers (1986), 55-84.
- [13] Fayad, Mohamed and Schmidt, Douglas,

- "Object-oriented application dimensions," *Communications of the ACM*, October 1997, Vol.40, No.10, pp.32-38.
- [14] Fayad, Mohamed E., W.C. Tsai and M.L. Fulghum, "Transition To Object-Oriented Software Development," *Communications of the ACM*, February 1996, Vol.39, No.2, pp.108-121.
- [15] Hackathon, R.D. and J. Karimi, "A dimension for Comparing Information Engineering Methods," *MIS Quarterly*, June 1989, 203-220.
- [16] Henderson, J.C. and Cooperider, J.C., "Dimensions of I/S Planning and Design Aids : A Function Model of CASE Technology," *Information Systems Research*, 1, 3 (1990), 227-254.
- [17] Howden, William, "Contemporary Software Development Environment," *Communications of the ACM*, May 1982, Vol.25, No. 5, pp.318-329.
- [18] Iivari, Juhani and Koskela, Erkki, "The PICO Model for Information Systems design," *MIS Quarterly*, September 1987, pp.401-419.
- [19] Kaindl, Hermann, and Carroll, John M., "Symbolic Modeling in Practice," *Communications of the ACM*, January, 1999, Vol.42. No.1, pp.28-30.
- [20] Kjaer, A. and H.K. Madsen, "Participatory Analysis of Flexibility," *Communications of the ACM*, May, 1995, Vol.38. No.5, pp. 53-60
- [21] Kuhn, T.S., *The Structure of Scientific Revolutions(3rd Ed.)*, University of Chicago Press, Chicago and London, 1996.
- [22] Martin, James and McClure, Carma, *Diagramming Techniques for Analysts and Programmers*, Prentice Hall, Englewood Cliffs, NJ, 1985.
- [23] Monarchi David E., and Puhr, Gretchen I., "A Research Typology for Object-Oriented Analysis and Design," *Communications of the ACM*, September, 1992, Vol.35 No.9, pp.35-47.
- [24] Olive, Antoni, "Analysis of Conceptual and Logical Models in Information Systems Design Methodologies," *Information Systems Design Methodologies : A Feature Analysis*, T.W. Olle, H.G. Sol, and C.J. Tully Ed., Elsevier Science Publisher (1983), pp.63-85.
- [25] Olle, T.W., Hagelsstein, J. MacDonald, I.G., Rolland, C., Sol, H.G., VanAssche, F.J.M., and Verrjin-Stuart, A., *Information Systems Methodologies : A dimension for Understanding*, Wokingham, England : Addison-Wesley, 1988.
- [26] Parsons, J. and Y., Wand, "Using Objects For Systems Analysis," *Communications of the ACM*, December 1997, Vol.40 No.12, pp.104-110.
- [27] Pressman, Roger, *Software Engineering : A Practitioner's Approach*, MacGraw-Hill, 1997.
- [28] Purao, S., H. Jain and D. Nazareth, "Effective Distribution Of Object-Oriented Applications," *Communication of the ACM*,

- August 1998, Vol.41 No.8, pp.100-107.
- [29] Robillard, Rierre N., "The Role of Knowledge in software development," *Communications of the ACM*, January, 1999, Vol. 42. No.1, pp.87-92.
- [30] Scheer, August-Wilhelm and Hars, A., "Extending Data Modeling to Cover the Whole Enterprise," *Communications of ACM*, September, 1992, Vol.35 No.9, pp. 166-171.
- [31] Shelton, Robert E., "Business Objects Modeling with Business Patterns," *Data Management Review*, May, 1996.
- [32] Winograd, Terry, "From Programming Environments to Environments for Designing," *Communications of ACM*, June 1995, Vol.38, No.6, pp.65-74.

■ 저자소개



안 준 모

Texas A&M University, College Station, Business Computing Science 석사, State University of New York, Buffalo 경영정보학 박사를 받았

으며 전문분야 경험으로는 미국 EDS 정보전략 컨설턴트 과정 수료, Motorola University 소프트웨어 프로젝트 품질관리 과정 교수, LG-EDS 컨설팅 부분 책임 컨설턴트, 정보통신부 소프트웨어 산업정책 자문 교수, 한국 IT벤처 인큐베이터 협회 회장으로 활동 중이다. 현재 건국대학교 경영정보학과와 부교수로 재직 중이며 학술 연구 결과는 *Journal of Organizational Computing*, 경영정보학 연구 등에 발표하였으며 현재 경영정보학 연구 등에 편집 위원으로 활동 중이다.

〈부 록〉

평가에 활용된 객체지향 시스템 모델링 다이어그램

다 이 어 그 램	설 명
사용사례(Use-case)다이어그램	시스템을 개발할 때, 사용자의 시스템에 대한 요건을 이해하기 위하여 전형적인 사용 시나리오를 작성하기 위한 다이어그램
CRC(class-Responsibility-Collaboration) 카드	파악된 각각의 클래스에 대한 사용사례로부터 기능을 할당하는 방법
클래스(class) 다이어그램	시스템내의 객체유형과 객체들 사이에 존재하는 각종 정적인 관계를 묘사
순차(sequence) 다이어그램	객체들이 어떻게 상호작용하는지를 메시지 순서에 초점을 맞추어 보여 준다
협력(Collaboration) 다이어 그램	상호작용이 일어나는 객체와 그들간의 연결을 중심으로 상호작용을 표현 한다.
상태(State) 다이어그램	한 객체, 서브시스템 혹은 시스템의 생성주기 동안의 일련의 상태(변화)를 표현한다.
활동(Activity) 다이어그램	대부분의 상태가 연산의 수행을 나타내는 활동(상태)이고 변환이 원천 상태에 있어서의 활동(오퍼레이션)의 완료와 더불어 시작되는 상태 다이어그램의 특수한 경우이다. 이는 클래스 태 오퍼레이션의 구현인 프로시저 자체의 상태변환을 표현한 것이다.
컴포넌트 다이어그램	소스코드 컴포넌트, 바이너리 코드 컴포넌트, 그리고 실행가는 컴포넌트 등 소프트웨어 컴포넌트간의 의존성을 표현한다.
전개(deployment) 다이어 그램	실행시 프로세싱 요소와 이들 상의 소프트웨어 요소, 프로세스, 그리고 객체의 구성항목을 보여준다
Fern 다이어그램	왼쪽에서 오른쪽으로 서브 클래스로 나누어진다.일반화에 기반을 둔 상속성과 분류하는데 유용하다. 서브타입을 보여주기 위하여 이진 tree형태로 확장하여 나타낸다.
Box 다이어그램	클래스를 박스로 나타내어 클래스를 분류하는데 유용하게 쓰인다. 서브타입을 보여주기 위한 직관적인 방법이다.
객체관계(Object-Relationship) 다이어 그램	object 간의 관계를 나타내기 위해 entity-relationship 다이어 그램과 유사하게 나타내어 object 간의 관계를 나타낸다.
Composed-of 다이어그램	서브타입을 보여주기 위한 직관적인 방법으로 box 다이어그램과는 다르게 복잡성을 감소시키기 위해 클래스 간의 서브타입의 관계임을 나타낸다.
OverView 다이어그램	전체 비즈니스 영역의 묘사하기위해 사용되는 다이어 그램.
상태변이(state-transition) 다이어그램	순차적인 상태(state)를 나타낸다.
이벤트(event) 다이어그램	상태간의 변화를 이벤트로서 표시한다.
객체흐름(object-flow) 다이어그램	high level에서 이벤트와 trigger의 동적인 표시하기 위한 다이어그램이다
실행(executable) 다이어그램	규칙을 가진 실행 다이어그램은 프로그램 코드로 변환할수 있다.