

# Agile 방법론과 MDA/MDD 기법을 활용하여 전자정부 표준 프레임워크 프로젝트의 품질 향상 사례

이승한\*, 박재표<sup>2</sup>

<sup>1</sup>엔키소프트, <sup>2</sup>송실대학교 정보과학대학원

## A Case Study of Software Development Quality Improvement by Agile Methodology and MDA/MDD Technology

Seung-Han Lee<sup>1\*</sup>, Jae-Pyo Park<sup>2</sup>

<sup>1</sup>Department of Research & Development, Enkisoft

<sup>2</sup>Graduate School of Information Science, Soongsil University

**요약** 본 논문에서는 전자정부 표준 프레임워크 프로젝트 내에서 MDD/MDA 기술 적용으로 인한 생산성과 품질 향상 효과를 기능점수 (Function Point) 관점과 코드 인스펙션 관점에서 분석하여 제시한다. 또한 본 사례의 프로젝트에서 적용한 MDD/MDA 적용 방법을 소개한다. MDD 기법의 발전과 더불어 그 효과를 검증하기 위해 다양한 연구가 진행되고 있으나, 대부분은 사례 연구를 통해 얻어지는 교훈이나, 제한된 측정 데이터로 분석된 효과를 제시하고 있다. 본 연구에서는 효율적인 개발 방법론과 전자정부 표준 프레임워크를 활용하여 MDD를 실제 정보시스템 개발 프로세스에 적용한 사례 연구를 통해 새로운 측정지표로 정량적 효과성을 측정하였다. 더불어 Application Life-Cycle 환경에서 모델링 된 정보들을 통합 저장소를 통하여 관리함으로써 추적성과 가시성이 확보된 방안을 제시한다.

**Abstract** This paper analyzes the effect on productivity by measuring Function Point per MM(man-month) in a software development project and code inspection which performed in Agile methodology, and introduce the method of applying MDA/MDD technology in the e-government standard framework project. MDA/MDD which recently raising its head is technology that increase software development productivity through automatic transformation form abstract model into lower abstract model, or from model to code. but, many Lessons obtained through the case studies, the analysis suggests a limited effect measurements. In this study, efficient development methodology and utilizing a e-government framework to measure the effectiveness of the MDD to the new measures through a case study applied to real information system development process. In addition to, managing information modeling through integrated repository presents a traceability and visibility.

**Key Words** : UML, Agile Methodology, MDA, MDD, Electronic government standard framework

### 1. 서론

본 논문에서는 빠른 속도로 급변하는 IT 환경에서 사용자의 서비스 요구가 다양해지고, 요구사항의 변경이

빈번한 소프트웨어 산업에서 이를 적시에 반영하고, 품질 및 생산성 향상을 위한 SW 기술인 MDA(Model Driven Architecture : 모델 지향 아키텍처, 이하 MDA) 및 Agile 개발 방법론에 의한 구축 전략과 도메인에 적

This work was supported by the ICT R&D program of MSIP/IITP. [10047178, Development a MDD Tool based on Open Source Software, for the expansion of E-Government Standard Framework usage]

\*Corresponding Author : Seung-Han Lee(Enkisoft)

Tel: +82-2-419-6362 email: seunghan.lee@enkisoft.co.kr

Received March 3, 2015

Revised March 26, 2015

Accepted April 9, 2015

Published April 30, 2015

합한 아키텍처를 구성하여 SW 개발 프로세스 및 모델링 도구를 활용한 전자정부 표준 프레임워크 기반의 정보 서비스를 개발하고 연관된 컴포넌트 시스템의 아키텍처를 구성하여 다양한 모델링 정보들을 저장하는 통합 저장소로 통합함으로써 데이터의 재활용 가치를 향상시키는 방안을 제시한다.

### 1.1 설계이론

SW 개발에 있어서 추상화(abstraction)와 자동화(automation)라는 개념이 객체(Object)를 중심으로 결합되기 시작하고 이러한 두 개념의 결합이 가져오는 무수한 잠재력들이 하나 둘 씩 드러나면서 MDD(Model-driven development)라고 통칭되는 새로운 모델링 기술로 발전되고 있다. MDD의 여러 가지 특징들 중의 한 가지는 소프트웨어 개발의 전 과정에서 Model을 가장 중요한 정보로 여긴다는 점이다. 이는 컴파일러가 특정 소프트웨어 시스템의 복잡성으로부터 개발자들을 자유롭게 했던 것처럼, 소프트웨어의 모델이 프로그램 코드가 가지는 복잡성으로부터 벗어날 수 있다는 가능성을 의미한다. MDD는 모델로부터 프로그램의 소스 코드를 자동 생성이 가능하게 하는 기술에 대한 청사진을 제시하는 개념이다. 오늘날 MDD 개념을 사용한 코드의 자동 생성 수준은 간단한 스킴레턴 코드의 생성으로부터 컴파일러가 그렸던 것처럼 완벽히 실행 가능한 코드의 생성에 이르기까지 천차만별이다. 자동화 수준과 모델의 정확도는 상호간에 영향을 주고받으면서 지속적으로 개선될 것이며, 이러한 과정을 통하여 MDD라는 개념이 소프트웨어 세계에 끼치는 영향은 컴파일러가 가져왔던 파급효과를 증가하게 될 가능성이 크다. 사실, 소프트웨어 개발에 있어서 MDD라는 개념의 적용은 새로운 것이 아니며 과거에도 수도 없이 시도되어 왔다. 그럼에도 불구하고 MDD가 최근 들어서 관심의 초점이 되는 이유는 과거와는 비교할 수 없을 정도로 MDD를 성공적으로 적용시키기 위한 기반 기술이 발전되고 준비되어 있다는 것이다. 이러한 기반 기술의 성숙은 MDD에 관한 표준의 제정을 가능하게 했으며, 이로 인하여 관련 도구의 통합이나 사용성 측면에서 급격한 발전을 초래하게 되었다. 이러한 기반기술의 선두에 서 있는 것이 바로 MDA이다.

MDA는 MDD 개념을 실제 구현 가능하도록 하는 기반기술로서 2002년 OMG에서 공식적으로 발표되었다. OMG에서 제안한 MDA의 궁극적인 목표는 소프트웨어

개발의 전 과정에서 필요한 모델링 기술에 대한 표준을 제정하는 것이다. 소프트웨어 개발자는 이렇게 모델 표준에 근거하여 특정 개발 도구에 종속되지 않으면서 소프트웨어 개발 과정에서 생성되는 모든 모델을 해당 모델 표준에 근거하여 통합하고 제어할 수 있게 된다. MDA의 기본을 이루고 있는 사상은 소프트웨어 자체에 대한 명세와 소프트웨어가 플랫폼을 사용하는 방법에 대한 명세를 분리하려는 데서 출발하고 있다. 이를 위하여 OMG에서는 소프트웨어 모델을 추상화 수준에 따라 CIM(Computer Independent Model), PIM(Platform independent Model), PSM(Platform Specific Model), Code Model의 네 가지 유형으로 분리하고 있으며, 모델 간의 Mapping을 통하여 변환(Transformation)을 위한 기반을 만들어 가고 있다[1].

본 논문에서는 이렇게 MDA/MDD 기술을 적용한 Agile 방법론 프로젝트에서의 개발 생산성을 기능점수(Funtion point) 단위로 적용하여 비교 분석함으로써 MDA/MDD 기술 적용 후의 개발 생산성 향상 효과를 살펴본다. 또한 전자정부 표준 프레임워크에서 수행된 MDA/MDD 기술 적용 방법을 소개한다.

## 2. 관련연구

### 2.1 대상 프로젝트 개요

본 논문에서 제시하는 사례는 대학교의 학사 행정도메인으로 하는 D대학의 학사 행정 시스템 구축 프로젝트이다. 주요한 개발 범위는 D대학의 통합 게시판 및 설문조사와 학사 행정 시스템의 운영 프로세스를 포함한다. 개발방법론은 Enkisoft에서 구축한 MDD 방법론을 활용하였으며, Fig. 1과 같이 SINVAS UML1.0 모델링 도구를 활용하여, 분석과 설계를 수행하였다.

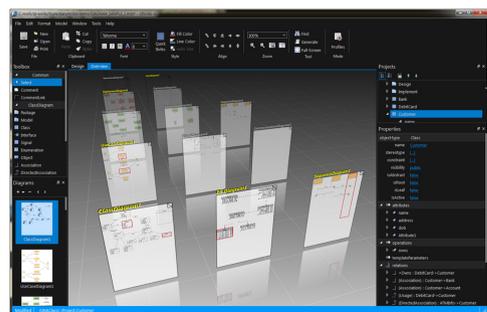


Fig. 1. SINVAS UML Tool of Enkisoft

### 2.2 개발 생산성 향상 효과의 측정

본 논문에서는 개발 규모를 기능점수(Function Point)로 산출하여 개발 생산성을 측정하였으며, 이를 위하여 정보통신부에서 고시한 [소프트웨어 대가기준(제2005-22호)]을 활용하였다.

MDD 기반의 프로젝트를 추진한 후, 사례 프로젝트와 국내 SW 공학박서와의 개발 생산성을 비교 하였을 때, 30% 수준의 개발 생산성 효과를 보았으며, 통합 게시판의 자동 코드 생성으로 인하여 프로젝트 비용을 절감하는 효과를 가져왔다. 사례 프로젝트는 SW 공학박서에서 발행하는 데이터와 비교하여 중소기업의 개발 생산성 평균인 21.9(FP/MM)보다 35.8% 향상된 29.74(FP/MM)으로 측정되었으며, 국외 프로젝트의 평균 개발 생산성 대비 30% 이상의 향상된 생산성 향상률을 보이고 있다. 이러한 생산성 향상의 효과는 본 사례 프로젝트에 MDA/MDD 기술을 적용하여 모델 및 코드를 자동 생성함으로써 얻어진 것으로 추정해 볼 수 있다. 다음 Fig. 2는 사례 프로젝트에 적용된 아키텍처이다.

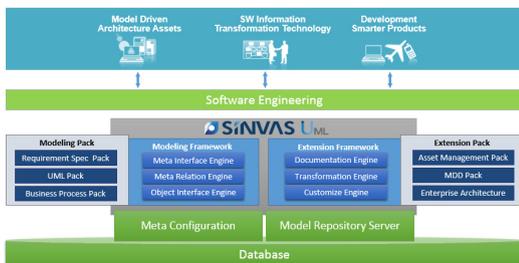


Fig. 2. System Architecture

### 3. 제안하는 MDA/MDD의 기술

본 프로젝트에서는 MDD의 기본 개념과 OMG에서 발표한 MDA 가이드를 프로젝트 환경에 적합하게 변경하여 적용하였다. 특히 비즈니스의 성과 향상을 위하여 Web-site 프로젝트 구축에 있어서 전자정부표준프레임워크를 활용함으로써 SW의 표준화, 품질 및 재 사용성을 향상시킴으로써 서비스의 품질 향상 및 투자 효율성을 높일 수 있는 것이 가능해질 수 있었고, PIM과 PSM을 분리함으로써 PIM을 변경하지 않고도 기술 플랫폼의 변화나 요구사항에 빠르게 대처할 수 있게 되었다.

다음 Fig 3에서 볼 수 있는 것처럼 MDA/MDD 기술

로 모델 변환의 구현뿐만이 아니라 통합 저장소 환경을 제공함으로써, 효과적인 협업 환경을 지원하여 대규모 프로젝트 및 신규 프로젝트 추진 시 재사용이 가능할 수 있도록 하였으며, 개발 조직의 개발 환경 표준화와 개발 생산성 향상에 크게 기여할 수 있음이 확인되었다.

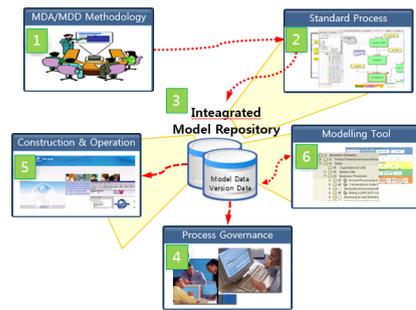


Fig. 3. Collaboration Environment for Modelling

이러한 통합 저장소 환경은 IT거버넌스 프로세스를 제공하는 정보시스템의 활용이 확대되면서 인력, 프로세스 등 총괄적으로 관리할 수 있는 Enterprise Architecture, Compliance, 서비스관리, 투자대비효과(ROI)등의 업무 추진 환경에 크게 기여할 수 있을 것이다.

또한, 모델의 변환 규칙으로 매우 중요한 컴포넌트 구성에 있어서, PSM 자동 생성 시 Verify 기능을 활용하여 실행 가능한지에 대한 모델 검증을 함으로써, 아래 Fig. 4와 같이 Boundary, Control, Entity의 모델 변환 컴포넌트로 구축되었으며, 이는 메타모델을 정의함으로써 가능할 수 있음을 확인할 수 있다.

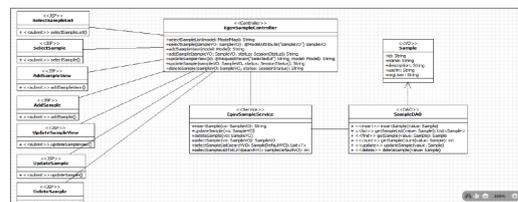


Fig. 4. Transformation to Component Model

이와 같이 모델링 환경에서 메타 모델로 정의되어지는 영역은 매우 광범위 하며, 이를 통하여 Database 설계, Business Process Management등의 SW 아키텍처의 분석, 설계 단계에서 중요한 환경을 효과적으로 제공할 수 있다.

### 4. 성능평가 및 비교분석

본 장에서는 제안하는 MDA/MDD 기술로 기존의 프로젝트 생산성과 SW 공학센터에서 발행하는 데이터의 중소기업 개발생산성과 비교하여 MDD를 적용한 사례 프로젝트의 개발 생산성이 크게 향상되었음을 확인할 수 있었다. 다음의 Fig. 5에서 이에 대한 정량적 지표를 보여주고 있다.

Title	SW .Center	Case Project	increase
Function Point	-	267.7	-
Productivity	21.9	29.74	35.8%↑
Man/Month	-	9 (M/M)	-

Fig. 5. SW Productivity indicator

Fig. 5와 같이 정량적 지표를 평가하기 위한 SW개발 생산성 계산 공식은 다음과 같다.

$$SW개발 생산성 = \frac{프로젝트 전체 FP}{프로젝트 투입 MM} \text{ (단위 } FP/MM)$$

본 논문의 대상 프로젝트의 SW 개발 생산성은 267.7 / 9 = 29.74 로서 기준 생산성인 21.9(FP/MM)과 비교하여 35.8%의 향상된 효과를 보여준다. 또한, SW공학백서의 결함밀도 측정 기준과 비교하였을 때에도 기준 결함밀도인 0.12와 비교하여 적용된 결함밀도는 0.06으로 운영 결함 밀도가 50% 이상 수준으로 향상되었음을 확인할 수 있었다. 그리고 Fig. 6에서와 같이 기존 프로세스의 경우 11개의 모델을 수작업으로 작성해야 했지만, MDA/MDD 방법론에서는 5개의 모델만 작성하고 나머지는 모델링 도구에 의해 자동으로 생성되기 때문에 5개의 모델에 대해서는 재활용이 가능할 뿐만 아니라, 주요 프로세스가 자동화 되어 프로세스의 결과를 예상하기가 쉽고 일정하며, 동일한 플랫폼 환경에서 동일하게 아키텍처를 적용할 수 있게 된다.

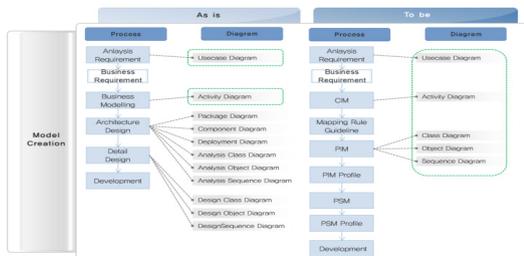


Fig. 6. Difference between existing Methodology

이 결과들은 MDA/MDD 방법론 적용의 우수성을 입증할 수 있을 뿐만 아니라, 자동 생성된 소스코드들의 역공학 및 재활용이 가능하기 때문에 기술적인 가치가 크다고 볼 수 있다.

### 5. 결론

MDA/MDD 기술은 정보화 시스템의 구현 시 프로그램 코드를 자동 생성하는 기술로서, 모델의 변환, 코드의 자동 생성, 코드의 역 변환, 산출물의 자동 생성 및 다양한 시스템 환경으로의 산출물 변환과 모델 검증 등의 분야에서 기술을 적용하였다. 그 결과 개발 생산성 및 코드의 품질을 향상시키는 효과를 얻을 수 있었고, 코드와 산출물을 모델을 통하여 자동 생성함으로써 설계와 코드, 그리고 산출물간의 일관성을 유지하기가 용이해졌으며 모델 검증의 자동화를 통하여 소프트웨어의 설계 품질을 보증하는 활동을 체계적으로 수행하는 것이 가능해졌다.

특히 국내 프로젝트의 평균 개발 생산성과 비교하여 100% 이상의 생산성 개선 효과를 가져왔다.

다만, 아직 국내의 경우 반복적인 개발 방법론 적용에 익숙하지 않고, 다양한 산업별 모델링 기술에 대한 기술적 투자 및 관심도가 부족하여 향후 이러한 성공 사례들이 잘 전파되어야 IT 환경을 변화 시키는데 크게 기여할 수 있을 것이다.

### References

- [1] Choi, Jeongil, "A Case Study of SW Development Productivity Improvement by MDD/MDA Technology Application in CBD Project", 2006.
- [2] Model Transformation: The Heart and Soul of Model-Driven Software Development: Shane Sendall, Wojtek Kozaczynski, IEEE Software, 2003.
- [3] MDA Guide Version 1.0.1:Joaquin Miller and Jishnu Mukerji, OMG, 2003
- [4] Model-Driven Development:A Metamodeling Foundation:Colin Atkinson, Thomas Kuhne, IEEE software, 2003.
- [5] Model Driven Process Engineering:Erwan Breton, jean Bezivin, Proceedings of the 25th Annual International Computer Software and Applications Conference

- (COMPSAC.01), 2001.
- [6] MDA Distilled:Principles of Model-Driven Architecture :Stephen J.Mellor, Kendall Scott, Axel Uhl, Dirk Weise, Addison Wesley, 2004
- [7] MDA Explained: The Model Driven Architecture Practice and Promise:Anneke Kleppe, Jos Warmer, Wim Bast, Addison Wesley, 2003.
- [8] An MDA Manifesto:Grady Booch, Alan Brown, Sridhar Lyengar, James Rumbaugh, Bran Selic, MDA Journal, 2004.
- [9] Y. Jeon, S. Im, H. Hwang, "Design of Open Gateway Framework for Personalized Healing Data Access", The Journal of The Institute of Internet, Broadcasting and Communication, Vol. 15, No. 1, pp.229-235, Feb. 28, 2015.  
DOI: <http://dx.doi.org/10.7236/JIIBC.2015.15.1.229>
- [10] Y.-H. Kim, S.-H. Cho, "Design and Implementation of Lightweight ESBUS Engine for Service Oriented Architecture", The Journal of The Institute of Internet, Broadcasting and Communication, Vol. 14, No. 6, pp.131-137, Dec. 31, 2014.  
DOI: <http://dx.doi.org/10.7236/JIIBC.2014.14.6.131>
- [11] K. Yoon, H. Kwon, " A study on Deployment of the Optimized WBS and Effective Small and Medium Enterprise Informatization System using Standardized PSDM", The Journal of The Institute of Internet, Broadcasting and Communication, Vol. 14, No. 6, pp.199-205, Dec. 31, 2014.  
DOI: <http://dx.doi.org/10.7236/JIIBC.2014.14.6.199>
- [12] D. Lee, "A Software Framework Design for Providing Presence Information of SMS Users to RCS-e App", The Journal of The Institute of Internet, Broadcasting and Communication, Vol. 14, No. 5, pp.145-151, Oct. 31, 2014.  
DOI: <http://dx.doi.org/10.7236/JIIBC.2014.14.5.145>
- [13] S.-U. Lee, M.-B. Choi, "An Establishment of the Process System for Software Requirements Engineering", The Journal of The Institute of Internet, Broadcasting and Communication, Vol. 14, No. 1, pp.37-49, Feb. 28, 2014.  
DOI: <http://dx.doi.org/10.7236/JIIBC.2014.14.1.37>

---

## 이 승 한(Seung-Han Lee)

[정회원]



- 2006년 2월 : 경희대학교 e-Business학과 학사 수료
- 2011년 2월 : 숭실대학교 정보과학 대학원 석사 수료
- 2015년 3월 ~ 현재 : 숭실대학교 컴퓨터학과 박사 과정
- 2014년 2월 ~ 현재 : 엔키소프트 개발팀장

<관심분야>

SW Architecture, MDA/MDD, UML, SW 생명주기

---

## 박 재 표(Jae-Pyo Park)

[정회원]



- 2004년 8월 : 숭실대학교 컴퓨터학과 공학박사
- 2008년 9월 ~ 2009년 8월 : 숭실대학교 정보미디어 기술연구소 전임연구원
- 2010년 3월 ~ 현재 : 숭실대학교 정보과학대학원 교수

<관심분야>

컴퓨터통신, 보안, 암호학, 멀티미디어 통신