

컴포넌트 기반 시스템 개발 방법론 개발

하수정*, 조진희*, 함동한*, 김진삼*

*전자통신연구원 컴퓨터.소프트웨어연구소

e-mail : hsj@etri.re.kr, chojh@etri.re.kr, dhham@etri.re.kr, kjs@etri.re.kr

Development of Methodology for Component-Based Systems Development

Su-Jung Ha*, Jin-Hee Cho*, Dong-Han Ham*, Jin-Sam Kim*

*Computer & Software Research Laboratory, ETRI

요 약

소프트웨어의 위기를 극복하고자 하는 하나의 해결책으로 등장하기 시작한 소프트웨어 개발 방법론이다. 국내에서는 '80년대 후반부터 방법론의 도입이 시작되었고 시스템 통합 사업의 팽창으로 '90년대부터는 더욱 가속화되었다. 구조적, 정보공학, 객체지향으로 이어진 국내 방법론 시장은 최근에는 작은 단위의 소프트웨어 부품을 조립하여 더 큰 소프트웨어 시스템을 구축하자는 컴포넌트 기반 개발에 관심을 가지기 시작하면서 컴포넌트 기반 시스템 개발 방법론에의 관심이 커지고 있다. 이에 본 고에서는 본 연구원에서 개발한 컴포넌트 기반 시스템 개발 방법론, 마르미-III 버전 2.0을 소개하고자 한다.

1. 서론

소프트웨어의 위기를 극복하고자 하는 해결책 중의 하나로 시작된 소프트웨어 개발방법론은 선진국에서는 정부나 공인 기관, 기업체 주도로 구조적 방법론이나 정보공학 방법론에 대한 연구가 이루어지고 개발 모델로 정립시켜 실제 많은 프로젝트에 적용하여 왔다. 또한 '80년대 말부터는 객체지향 방법론의 연구 및 활용도 급속히 확산되어 오고 있다. 그러나 소프트웨어의 재사용성을 보장해 주리라 여겨졌던 객체지향의 언어나 방법론들이 우수한 개념과 연구 성과에도 불구하고, 소프트웨어의 재사용 측면에서는 주목할만한 결과를 나타내지 못하면서, 학계나 산업계는 새로운 개념에 눈을 돌리기 시작하였다. 기존의 다른 산업 분야에서는 일반적인 개념으로 사용되고 있는 작은 부품을 조립하여 제품을 개발하는 방식으로 소프트웨어 산업에 이 방식을 적용하고자 하는 시도가 바로 그것이다. 물론 객체지향 방법의 경우도 같은 개념에서 출발한 방법이지만, 성공적으로 적용하기 힘든 많은 요인이 있었다. 타인이 개발한 소프트웨어에 대한 개발자 상호간의 불신과 잦은 변경 및 블랙박스 형태의 부품을 사용하는 데서 오는 부담감 등으로 소프트웨어 재사용에 어려움을 겪고 있다. 그러나 갈수록 복잡해지고 대형화 되어가고 있는 소프트웨어 개발에 지금까지의 방식으로는 더 이상 주문 제작 형태의 소프트웨어 생산 수준에서는 감당하기 어려움에 따라 품질이 보장된 소프트웨어 부품, 컴포넌트를 만들고 그 부품, 컴포넌트를 이용한 시스템 개발에 대한 수요가 늘고 있는 현실이다.

컴포넌트는 객체보다 확장된 재사용의 단위로서 지속성 있는(Persistent) 데이터와 공용 오퍼레이션을 가지고 캡슐화 되어 있다는 점에서는 객체와 유사하나, 보다 큰 서비스의 집합으로서, 응집도(Cohesion)는 높고 결합도(Coupling)는 낮은 개발단위라고 말할 수 있다. 인터페이스를 통하여 외부 어플리케이션에게 서비스를 제공한다는 점에서는 기존의 라

이브리리와 유사하나, 블랙박스형태로 커스터마이징이 가능하며, 영구 자료를 가지고 있다는 점에서 차별된다. 검증된 컴포넌트의 사용은 컴포넌트를 기반으로 하는 시스템의 생산성 뿐만 아니라, 안정성도 보장해 주는 소프트웨어 재사용의 대안으로 부각되며 연구되고 있다.

또한 소프트웨어 산업이 급속하게 발전해감에 따라 정보 기술 업체간 경쟁이 더욱 심화되어 소프트웨어 재사용성, 적시성(time to market), 유지 보수성 등이 업체의 생명력으로 대두되면서 컴포넌트 기반 기술이 점차 각광을 받고 있다. 컴포넌트 기반 기술은 컴포넌트 단위로 독립적인 개발이 가능하므로 병행적/단계적 소프트웨어 개발을 통해 개발 기간을 단축시킬 수 있다. 또한 컴포넌트의 수행은 컴포넌트 플랫폼 아키텍처 상에서 이루어지므로 아키텍처가 제공하는 트랜잭션, 보안성, 지속성 등의 다양한 서비스를 이용할 수 있으며 또한 컴포넌트 단위로 수행되므로 컴포넌트의 대체 및 유지보수가 수월한 장점이 있다.

현재 산업계에서 쓰이고 있는 대표적인 컴포넌트 기반 시스템 개발 방법론으로는 Rational의 RUP (Rational Unified Process)[1], Computer Associates사의 CBD96[2], Compuware사의 UNIFACE[3], PrincetonSoftech사의 Select[4] 등이 있다.

국내의 경우, 한국전자통신연구원에서 국내의 개발 여건을 반영하여 국내 개발 기술준에 의해 개발된 한국형 정보 시스템 구축 방법론, 마르미((MaRMI, Magic and Robust Methodology Integrated),[5] 객체지향 정보통신용 시스템 개발 방법론, 마르미-III[6]를 개발하여 산업체에 보급하고 있다. "마르미"는 우리말의 '마름질(하다)'를 어원으로 하며, 국내에서 단편적으로는 널리 사용되고 있지만 종합적으로는 체계적인 구성을 갖추고 있지 못한 개념, 방법, 기법, 절차 및 경험적 지식들을 쉽고 편리하게 사용할 수 있도록 재구성하여 '우리의 옷' (개발 여건에 보다 적합하도록 만들어진 한국형 정보시스템 개발 방법론이다

본 논문에서는 컴포넌트 기반 시스템 개발 방법론, 마르미-III의 특징, 구성, 공정에 대하여 상세히 설명하고자 한다. [8]

2. 마르미

2.1 마르미-III의 특징

마르미-III는 다음과 같은 특징을 가지고 있다. 우선, 마르미-III는 소프트웨어 아키텍처 중심의 개발 방법을 채택하였다. 재사용성이 높은 컴포넌트를 개발하기 위해서는 범용성이 있으면서 확장가능하고 운영체제와 독립적인(Platform Independent) 아키텍처 위에서 구현해야 한다. 또한 모든 컴포넌트가 객체지향적인 소프트웨어 구조로 개발하여야 하는 것은 아니지만, 재사용성이 높고 요구변화에 영향이 적은 견고한 구조를 만들기 위해서 객체지향적 프레임워크 기술을 이용하여 아키텍처를 구현하는 것이 바람직하다. 또한 CORBA(Common Object Request Broker Architecture), EJB, DCOM(Distributed Component Object Model)과 같은 분산기술 구현 메커니즘 위에 객체지향적 응용 아키텍처를 구현함으로써 향후 발생 가능한 기술변화에 대처할 수 있다.

마르미-III는 컴포넌트 기반 시스템 개발 공정은 물론 관리 공정도 함께 정의하고 있다. 기존의 마르미 방법론을 포함한 개발 방법론은 관리 공정이 제대로 정의되어 있지 않거나, 정의되어 있더라도 개발공정의 일부분으로 최소한의 관리 공정, 즉 계획을 수립하고 결과를 검토하는 작업을 위주로 정의하고 있으나 마르미-III에서는 그림 1과 같이 관리공정과 개발 공정의 연계를 통하여 프로젝트 시작부터 프로젝트 진행중에 품질 및 위험 관리를 포함한 모든 관리요소를 정의하고 있다. 또한 프로젝트가 완료되면 프로젝트 평가 및 개발 시스템의 인도 및 사후관리를 위한 작업도 정의하고 있다.

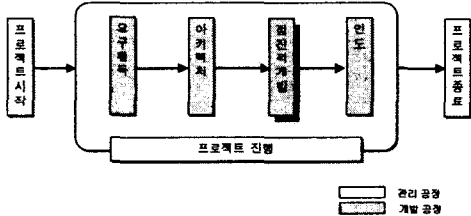


그림 1. 마르미-III 공정 구성도

마르미-III의 개발 대상 플랫폼은 현재는 SUN사의 EJB(Enterprise Java Beans)를 고려한 지침이 제공되고 있으며, 개발공정은 반복적이고 점진적으로 수행이 되는데, 반복적인 개발 활동의 단위를 미니프로젝트(mini-project)라고 정의하고 있는데, 각 미니프로젝트에서는 타임박스(time box)의 개념으로서 유스케이스를 구현하게 된다. 타임박스는 소규모의 개발인원이 엄격히 제한된 시간 내에 사용할 수 있는 시스템을 개발하는 것으로 효과적인 개발 관리를 위한 방법이다. 이것을 통해 개발자 중심이 아닌 사용자 중심으로, 동적으로 변화는 요구사항과 위험을 효과적으로 관리할 수 있고 개발자들에게는 주기적으로 동기유발을 할 수 있는 장점이 있다.



그림 2. 개발공정 수행

2.2 마르미-III 구성

마르미-III는 절차서, 기법서, 양식정의서 및 적용사례서로 구성되어 있다. 공정은 컴포넌트 개발 공정과 관리 공정으로 구분되어 정의하였으며, 개발 공정은 요구획득단계(R0000), 아키텍처단계(A0000), 점진적 개발단계(D0000), 인도단계(T0000), 4개의 단계로 이루어져 있고, 관리공정은 프로젝트시작단계(S0000), 프로젝트진행단계(C0000), 프로젝트종료단계(E0000)로 구성되어 있다. 단계는 논리적으로 서로 연관된 작업을 하나로 묶은 활동으로 구성되어 있다. 각 작업은 하나 이상의 산출물을 만들어 내며, 이를 위한 상세한 세부 절차가 정의되어 있다. 산출물을 위한 양식은 양식정의서에 정의되어 있다.

- 마르미-III 절차서 (7개 단계/25개 활동/93개 작업)
- 마르미-III 기법서
- 마르미-III 양식정의서
- 마르미-III 적용사례서

마르미-III의 개발공정의 4개의 단계는 위의 그림 2와 같이 요구획득단계, 아키텍처 단계의 각 작업을 순차적으로 진행한 후에 점진적 개발단계에서는 유스케이스를 기준으로 나누어진 미니프로젝트를 반복적으로 수행함으로써 전체 시스템을 점진적(incremental)으로 개발한 후에 인도단계에서 시스템에 사용자에게 인도하는 흐름으로 개발 공정이 진행된다. 세 번째 단계인 점진적 개발단계 내에 있는 미니프로젝트는 활동들의 집합으로 제한된 시간 내에서 사용자가 사용 가능한 형태의 시스템을 생성하는 단위이다.

일반적으로 방법론은 소프트웨어 생명주기를 바탕으로 관리, 방법, 환경적 요소로 구성되어 있지만 이는 상위 수준에서의 개념적인 설명이다. 방법론 절차서에는 이러한 구성 요소들이 보다 세분화되어 그들간의 관계가 유기적으로 연계되어서 설명되어야 하고 서술하고자 하는 수준(내용의 정도)에 따라 세분화 정도가 달라질 수 있다. 마르미-III의 메타 모형은 그림 3과 같다.

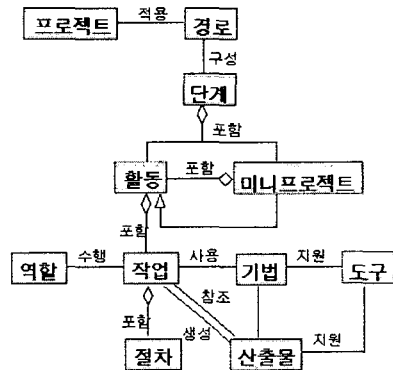


그림 3. 마르미-III 메타모형

마르미-III의 메타모형은 OMG(Object Management Group)에 의해 명세화된 SPEM(Software Process Engineering Metamodel)[10]과 호환성을 가지도록 정의되어 있다. 메타모형은 프로세스가 어떤 구성을 가지는지, 그 구성요소간에 어떤 관계를 가지는지 나타낸다.

2.3 프로젝트시작 단계

첫 번째 단계인 프로젝트시작 단계의 목적은 다음과 같다.

- 프로젝트의 수행목적과 목표, 범위를 결정하여 프로

젝트를 기안한다.

- 프로젝트 제안요청에 따라 제안서를 제출받고 가장 적합한 용역 수행업체를 선정한다.
- 정보시스템 구축 계약 내역에 따라 프로젝트를 확정하고 상세한 수행계획을 수립하여 승인을 얻는다.

프로젝트시작 단계는 모두 9 개의 작업으로 이루어져 있는데, 다음과 같이 2 개의 활동으로 그룹핑할 수 있다. [그림 4]

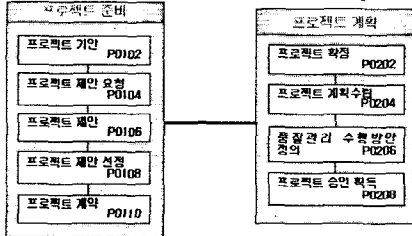


그림 4. 프로젝트시작 단계 구성도

2.4 프로젝트진행 단계

프로젝트진행단계의 목적은 다음과 같다.

- 개발 공정(요구 획득, 아키텍처, 점진적 개발, 인도)의 각 단계에 대한 착수를 준비한다.
- 개발 공정 별 작업계획을 수립하여 진척관리의 기준을 수립한다.
- 미니프로젝트 계획을 수립하여 점진적 개발의 효과를 최적화한다.
- 프로젝트 진행을 추적하고 평가하여 최적의 프로젝트 진행을 보장한다.
- 품질관리를 수행하여 프로젝트 산출물의 품질을 확보한다.
- 단계점검을 통하여 개발공정 단계별 프로젝트 결과를 평가하고 후속조치를 계획하여 프로젝트 계획을 수정하고 완전성을 도모한다.

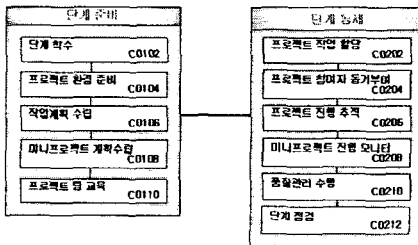


그림 5. 프로젝트진행 단계 구성도

프로젝트진행 단계는 모두 11 개의 작업으로 이루어져 있는데, 다음과 같이 2 개의 활동으로 그룹핑할 수 있다. [그림 5]

2.5 요구획득 단계

개발 공정의 첫 번째 단계인 요구획득단계로서 목적은 다음과 같다.

- 사용자 업무를 지원하기 위한 시스템의 비전을 개발한다.
- 시스템 비전을 기반으로 실제 시스템의 사용자로부터 요구사항을 수집하고 경우에 따라서는 시스템의 개발 배경을 명확히 이해하기 위해 업무 모형을 작성한다.
- 개발 시스템의 범위를 명확히 한 후, 사용자 요구사

항을 유스케이스 모형과 개념 모형으로 정리한다. 그리고 UI 프로토타이핑을 통하여 사용자와 함께 이 모형들이 요구사항에 맞게 작성되었는지 검증한다.

- 작성한 유스케이스 모형 및 개념 모형을 기반으로 시스템 아키텍처 초안을 정의하고 재사용 가능한 컴포넌트를 조사한다.

요구획득 단계는 모두 8 개의 작업으로 이루어져 있는데 다음과 같은 3 개의 활동으로 구성되어 있다. 계획 단계의 작업 구성도는 그림 6과 같다.



그림 6. 요구획득 단계 구성도

2.6 아키텍처 단계

아키텍처 단계의 목적은 다음과 같다.

- 개념모형을 기반으로 유스케이스 분석을 통하여 추가적인 객체 및 클래스를 도출하고 객체 모형을 작성한다.
- UI 프로토타입을 기반으로 화면 및 보고서와 같은 사용자 인터페이스를 설계한다.
- 사용자 요구사항을 기반으로 소프트웨어 및 비즈니스 아키텍처를 설계하고 이러한 아키텍처 기반에서 재사용 가능한 컴포넌트를 조사하여 획득한다.
- 프로토타이핑을 통하여 선정된 아키텍처를 검증하고 평가한다.
- 점진적 개발단계에서 개발되는 시스템에 대한 테스트 계획을 수립하고 데이터 이관 및 전환 계획을 수립한다.

아키텍처 단계는 모두 14 개의 작업으로 이루어져 있는데 다음과 같은 4 개의 활동으로 정의되어 있다. 아키텍처 단계의 작업 구성도는 아래 그림 7과 같다.

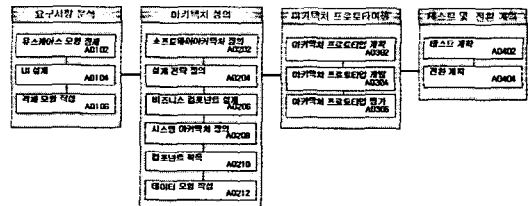


그림 7. 아키텍처 단계 구성도

2.7 점진적 개발 단계

세 번째 단계인 점진적 개발 단계의 목적은 다음과 같다.

- 미니프로젝트 수행계획에 따라 컴포넌트, 데이터베이스, GUI 등을 구현하여 점진적으로 시스템을 개발한다.
- 각 미니프로젝트에서 유스케이스를 개발하고 나면 사용자의 검토를 받아 요구사항을 만족하는지 확인한다.
- 계획된 모든 미니프로젝트를 수행하고 나면 통합된 시스템의 기능성 및 성능이 사용자 요구사항을 만족하는지 확인하기 위한 시스템테스트를 수행하고 본 단계의 수행결과를 평가한다.

점진적 개발 단계는 모두 30 개의 작업으로 이루어져 있는데 아래 그림 8 과 같이 8 개의 활동으로 그룹핑되어 있다.

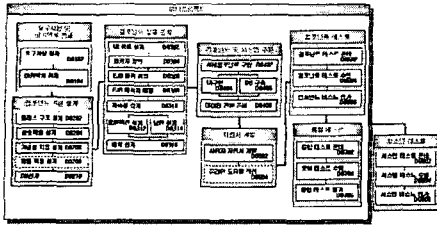


그림 8. 점진적 개발단계 구성도

2.8 인도 단계

- 개발 과정의 마지막 단계인 인도 단계의 목적은 다음과 같다.
- 개발자 환경에서 개발된 결과물을 컴포넌트인 경우에는 컴포넌트 리포지터리에, 컴포넌트 기반 시스템인 경우에는 실제 시스템이 운영될 사용자 환경에 설치한다. 기존에 운영되고 있는 리포지터리나 시스템이 있을 경우 신규 시스템으로 전환하여 원활한 운영이 가능하도록 한다.
 - 개발된 컴포넌트 또는 시스템에 대하여 최종적으로 사용자 요구사항과의 일치 여부에 대하여 승인을 얻고 프로젝트의 모든 전달물을 사용자에게 전달하고 인계한다.
 - 본 단계는 사용자의 업무에 영향을 주게 됨으로 사용자의 적극적인 참여가 필수적이며, 사용자는 인수와 함께 시스템을 운영할 수 있도록 준비하여야 한다.

인도 단계는 사용자의 업무에 영향을 주게 됨으로 사용자의 적극적인 참여가 필수적이며, 사용자는 인수와 함께 시스템을 운영할 수 있도록 준비하여야 한다.

인도 단계는 모두 10 개의 작업으로 이루어져 있는데 3 개의 활동으로 그룹핑되어 있다.[그림 9]

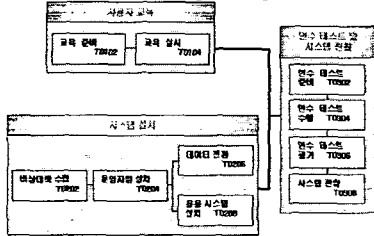


그림 9. 인도 단계 구성도

2.9 프로젝트종료 단계

마르미-III 의 마지막 단계인 프로젝트 종료 단계의 목적은 다음과 같다.

- 프로젝트 결과를 평가하고 종료를 보고한다.
- 프로젝트 교훈을 정리하여 다음 프로젝트에서 활용 가능하게 준비한다.
- 프로젝트 산출물들을 재사용 가능하게 정리한다.
- 프로젝트 비용을 정산한다.
- 프로젝트 참여자의 원활한 업무부귀를 지원한다.
- 유지보수 계획을 수립하고 운영계약을 체결하여 운영경으로 자연스럽게 전환시킨다.

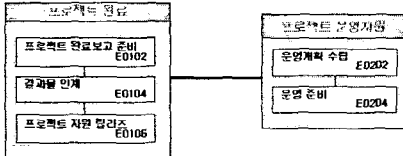


그림 10. 프로젝트종료 단계 구성도

프로젝트종료 단계는 모두 5 개의 작업으로 이루어져 있는데 2 개의 활동으로 그룹핑되어 있다.[그림 10]

3. 결론

마르미-III 는 객체지향 모형화 언어인 UML(Unified Modeling Language)[9]에 근거하고 있으며, 컴포넌트 기반의 시스템 개발을 위한 절차, 산출물, 기법, 지침을 제공하고 있다. 마르미-III 의 절차모형은 반복적이고 점진적인 모형을 채택하고 있으며 요구사항의 변경, 개발 여건의 변화에 민첩하게 적용할 수 있고, 사용자의 요구를 반영한 컴포넌트 기반 시스템을 개발할 수 있도록 지원하고자 노력하였다. 또한 마르미-III 에는 초기 아키텍처의 중요성, 유즈케이스 개념의 활용, 관리 공정내에 품질관리, 위험관리 기법을 적용할 수 있도록 정의되어 있다.

마르미-III 는 방법론 사용자에게 손쉽게 방법론에 접근할 수 있도록 국내의 어떤 방법론보다 상세한 설명과 지침을 제시하고 있다. 즉, 타 방법론에 비해서 국내 기업들이 빠른 시간 내에 채택하여 손쉽게 적용할 수 있는 방법론이 되도록 노력하였다. 현재 방법론 개발에 참여하고 있는 여러 공동 연구기관을 통하여 실제 컴포넌트 기반 시스템 개발에 본 방법론을 적용하여 방법론 개선 사항을 추출하여 하지만 마르미-III 방법론도 다른 방법론들과 마찬가지로 여러 분야의 다양한 형태의 프로젝트에 적용을 통해서 계속적으로 진화하는 과정이 필요하다. 다시 말하면, 마르미-III 방법론을 국내 다양한 기관과 업체에서 활발하게 적용하고 활용하여, 그 경험들이 피드백 되어 계속적으로 방법론의 보완으로 이어지는 순환 과정이 필요하다. 그리고 향후에는 EJB 뿐만 아니라 Microsoft 사의 COM+(Component Object Model plus), CORBA 의 CCM (CORBA Component Model) 등 다양한 플랫폼을 지원하고, 현재는 프로젝트 관리를 위한 공정만이 분리되어 있으나 품질관리, 위험관리 등의 공정을 분리하여 컴포넌트 기반의 통합 방법론을 개발하고자 한다. 또한 기존의 소유하고 있는 프로세스(마르미 방법론)을 프로젝트에 맞추어 커스터마이징을 하고 프로젝트 적용 후에 그 결과를 축적하여 조직의 프로세스를 정립해 나갈 수 있는 프로세스 지원 도구도 개발 예정이다.

참고문헌

- [1] Rational Software Corporation, Rational Unified Process, <http://www.rational.com/products/rup/index.jsp>
- [2] Computer Associates, CBD96, <http://www.sterling.com/cbdedge1/cbd96.htm>
- [3] CompuWare Corporation, UNIFACE Development Methodology, <http://www.compuware.com/>
- [4] PrinceTonSoftech, Select Perspective, <http://www.princetonsoftech.com/index.asp>
- [5] 전진욱 외, 소프트웨어 생산기술 개발 - 정보시스템 구축기술 개발, 연구보고서, 과학기술처, 1997.
- [6] 전진욱 외, 정보통신용 시스템 개발방법론 구축에 관한 연구(II), 연구보고서, 정보통신부, 1997.
- [7] 박창순 외, 컴포넌트기반 시스템 개발방법론 마르미-III, <http://www.component.or.kr/>, 2001.8.
- [8] 김진삼 외, 컴포넌트기반 시스템 개발방법론 마르미-III, <http://www.component.or.kr/>, 2001.8.
- [9] G.Booch, J.Rumbaugh, and I.Jacobson, The Unified Modeling Language : User Guide, Addison-Wesley, 1999.
- [10] Software Engineering Process Metamodel Specification, OMG, 2001.12.