

# 컴포넌트 추출기법을 상세화한 수정된 KobrA 프로세스

<sup>0</sup>장은주<sup>\*</sup> 이재현<sup>\*\*</sup> 김용성<sup>\*</sup> 유철중<sup>\*</sup> 장옥배<sup>\*</sup>

<sup>\*</sup> 전북대학교 컴퓨터과학과

<sup>\*\*</sup> 벽성대학교 컴퓨터계열

ejchang@chonbuk.ac.kr, jhlee@byuksung.ac.kr, [yskim, cjyoo, okjang]@moak.chonbuk.ac.kr

## Revised KobrA Process : Elaboration of Component Extraction Method

<sup>0</sup>Eun-Ju Chang<sup>\*</sup> Jae-Hyun Lee<sup>\*\*</sup> Yong-Sung Kim<sup>\*</sup> Cheol-Jung Yoo<sup>\*</sup> Ok-Bae Chang<sup>\*</sup>

<sup>\*</sup> Dept. of Computer Science, Chonbuk National University

<sup>\*\*</sup> Dept. of Computer, Byuksung College

### 요 약

기존 KobrA 프로세스는 컴포넌트 기반과 프로덕트 라인 공학의 개념을 모두 포괄할 수 있는 장점이 있지만, 컴포넌트 추출에 관한 명확한 가이드가 존재하지 않아서 특정 도메인에 적용하는데 어려움이 있다. 따라서 본 논문에서는 KobrA 프로세스에 컴포넌트 추출 단계를 추가하여 KobrA 프로세스를 수정하고자 한다. 본 논문은 기존의 KobrA 프로세스보다 효과적이고 실용적인 프로세스를 정의할 뿐만 아니라 컴포넌트 추출상의 모호성이 존재하는 문제점을 해결할 수 있다는데 그 의의가 있다.

### 1. 서 론

기존의 애플리케이션들은 여러 도메인들끼리의 공통적인 기능들이 있음에도 불구하고 도메인별로 별도로 구축되어 소프트웨어공학 관점에서 개발시간과 비용적인 측면에서 비효율적이라는 단점이 있었다. 독립적이고 대치 가능한 시스템의 일부분 또는 상호 작용하는 기능적 시스템을 형성하는 독립적 단위로 정의되는 컴포넌트는 그 기능별로 요구사항에 따라 각각을 조립하여 원하는 애플리케이션을 만들 수 있다[1,2]. 그렇기 때문에 개발 및 유지보수, 생산성 등 소프트웨어 품질을 획기적으로 향상시킬 수 있다. 또한, 소프트웨어 프로덕트 라인은 컴포넌트가 플러그인 될 수 있는 프레임워크를 제공하는 아키텍처를 기반으로 필요한 컴포넌트를 선택적으로 조합함으로써 시장의 요구사항에 맞는 시스템을 생산하는 방식을 말한다. 즉, 높은 품질의 프로덕트를 좀더 빠르고, 적은 비용으로 만들기 위해서 멀티 프로덕트 계획 사이클과 교환할 수 있는 혁신적인 재사용을 강조하는 접근 방식이다[3].

컴포넌트 기반 개발에서 설명하고 있는 활동과 개념들은 컴포넌트 모델링과 구현 또는 단일 시스템 환경에서의 재사용에 초점을 두고 있다. 컴포넌트 기반 개발은 어느 정도의 재사용의 문제는 해결할 수 있지만 프로덕트 라인 공학의 이점과 컴포넌트가 가져올 수 있는 재사용성을 향상시키는데는 한계가 있다. 이에 본 연구에서는 소프트웨어 산출물을 재사용하기 위해서 프로덕트들간의 공통성과 가변성을 상세하게 조사하여 개발 프로세스에 적용하기 위해, 기존의 KobrA 프로세스에 컴포넌트 추출기법을 상세화함으로써 KobrA 프로세스를 수정하고자 한다. 그리하여 보다 효과적이고 실용적인 프로세스를 정의한다.

본 논문의 구성은 다음과 같다. 2장에서는 본 연구의 기반이 되는 KobrA 프로세스에 대해 언급하고, 3장에서는 KobrA 프로세스의 수정된 기법을 제안한다. 마지막으로 4장에서는 연구 내용에 대한 결론과 향후 연구를 제시한다.

### 2. KobrA 프로세스

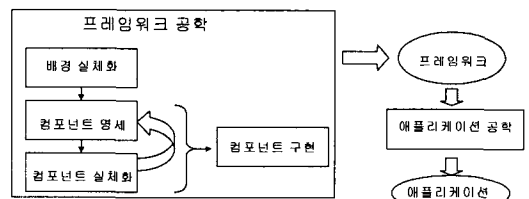
최근에 산업계에서 널리 쓰이고 있는 대표적인 컴포넌트 기

반 소프트웨어 개발 방법론은 CBD96, UML Components 방법 등이 있다. 이러한 컴포넌트 기반 시스템 구축을 위한 기존의 컴포넌트 개발 방법론은 컴포넌트 식별을 위하여 객체를 추출하는 부분이 비 효율적이고 시스템 컴포넌트를 추출하기 위한 방법이 제시되어 있지 않다. 또한 시스템의 전체 도메인을 중심으로 비즈니스 컴포넌트 식별을 위한 절차와 방법만을 제시하고 있으며 컴포넌트 식별을 위하여 대부분 개발자의 직관과 경험에 의존하는 문제점이 있다[4].

따라서 본 논문에서는 KobrA 프로세스 방법론을 이용하여 컴포넌트 추출기법을 상세화하여 보다 효율적이고 실용적인 프로세스를 정의한다.

KobrA는 보다 진보된 프로덕트 라인의 개념을 통해서 컴포넌트 공학(component engineering)을 확장하고 있다. KobrA 프로세스에서는 프로덕트 라인 개념을 사용한 컴포넌트 공학을 프레임워크 공학(architecture engineering)이라고 부른다. KobrA는 이러한 프레임워크 공학과 이것을 통해 제공된 프레임워크를 이용하여 애플리케이션을 만드는 애플리케이션 공학(application engineering)으로 구성된다[5].

[그림 1]은 KobrA에서 프로덕트 라인 공학의 라이프사이클을 나타낸다[6].



[그림 1] KobrA 프로덕트 라인 공학 라이프사이클

프레임워크 공학은 유사한 애플리케이션의 군을 분류하기 위해서 공통성과 가변성을 고려하여 이것을 명시적으로 표현한다. 즉, 프레임워크 공학에서는 도메인 분석을 통해 프로덕트들을 식별하고 프로덕트의 기능들을 각 프로덕트들의 공통적인 기능들과 가변적인 기능들로 구분한다. 이러한 과정의 프레임

워크 범위 분석이 끝나면 시스템 군들의 배경(context)을 표현하도록 요구사항을 분석하고, 비즈니스 모델링, 구조적 모델링, 액티비티 모델링, 상호작용 모델링, 결정 모델링의 기초적인 배경 작업들을 수행한다. 다음으로 컴포넌트 모델링의 작업으로 컴포넌트 추출과 명세를 거쳐, 실체화 단계를 통해 컴포넌트를 구현한다. 프레임워크 공학에서는 컴포넌트 모델링과 구체화(embodiment)활동을 일반화하고, 특정 프로덕트를 만드는 것이 아니라 범용 프로덕트를 만드는 것에 목적을 둔다.

애플리케이션 공학은 프레임워크 공학 동안에 개발된 프레임워크를 특별한 애플리케이션을 만들기 위해 사용한다. 애플리케이션 공학의 목적은 특정 시스템의 요구를 만족시키는 산출물들의 특정 집합을 생성하는 것이 목적이다.

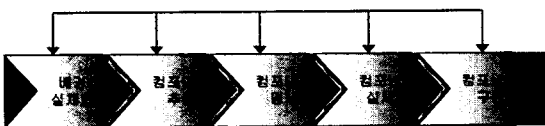
3. Kobra 프로세스의 수정

Kobra 프로세스는 컴포넌트 기반과 프로덕트 라인 공학의 개념을 모두 포괄하고 있는 장점이 있지만 컴포넌트 추출에 관한 단계가 명확하게 정의되지 않아서 특정 도메인에 대한 컴포넌트 기반 소프트웨어를 구축함에 있어서 컴포넌트 추출상의 모호성이 존재한다.

따라서 본 연구에서는 컴포넌트 추출 프로세스의 추가 및 이에 따른 각 프로세스의 조절을 통하여 Kobra 프로세스를 수정하였다. 즉, 기존의 Kobra 프로세스에 어떤 정보가 관리되어야 하는지 그 정보를 관리하기 위해 어떤 인터페이스가 필요하며, 그러한 인터페이스를 제공하기 위해 어떤 컴포넌트가 필요한지 그리고 이들 컴포넌트들이 어떻게 조립될 것인가를 추출해내는 작업이 중심이 되는 컴포넌트 추출 단계를 추가하여 기존 Kobra 프로세스를 수정하였다.

뿐만 아니라 컴포넌트 추출이 원활하게 진행되기 위해 배경 실체화 단계에서 프레임워크 범위 분석과 동시에 시스템의 군들의 배경을 표현하도록 요구사항을 분석하고, 비즈니스 모델링, 구조적 모델링, 액티비티 모델링, 상호작용 모델링, 결정 모델링의 기초적인 배경 작업들이 수행되도록 하였다. 또한 다음에 이어지는 컴포넌트 명세 단계에서는 앞의 컴포넌트 추출 단계에서 식별된 인터페이스들에 대한 컴포넌트를 명세하며 컴포넌트 아키텍처를 구성하는 구조적 모델링을 수행한다. 또한, 기능적 모델링을 통해 컴포넌트의 오퍼레이션을 명세하고, 행위 모델링에서 상태 다이어그램을 생성한다.

[그림 2]는 프레임워크 공학의 GNSS 개발 프로세스를 나타내며, 다음과 같은 단계로 수행된다.



[그림 2] 프레임워크 공학 프로세스

첫 번째 단계에서는 해당 도메인 분석을 통해 프로덕트들을 식별하고, 프로덕트의 기능들을 프로덕트 라인 공학의 개념으로 공통적인 기능과 가변적인 기능을 식별하는 배경실체화 단계를 거치게 된다.

두 번째 단계인 컴포넌트 추출단계에서는 초기 인터페이스와 컴포넌트 명세를 생성하여 컴포넌트 아키텍처의 초안을 만든다. 이러한 단계를 거침으로써 다음에 이어지는 컴포넌트 명세에서 명세와 아키텍처를 정제하고 보강하기 위한 기초를 제공하게 된다. 결국 컴포넌트 추출 단계에서는 어떤 정보가 관리

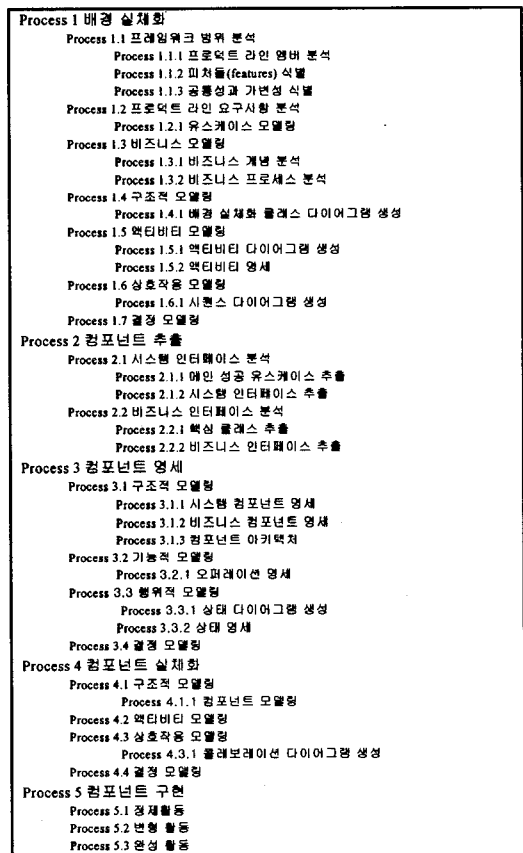
될 필요가 있고, 그 정보를 관리하기 위해 어떤 인터페이스가 필요하며, 그러한 인터페이스를 제공하기 위해 어떤 컴포넌트가 필요한지 그리고 이들 컴포넌트들이 어떻게 조립될 것인가를 추출해 내는 작업을 중점적으로 진행하게 된다.

세 번째 단계인 컴포넌트 명세 단계에서는 외부에 드러난 컴포넌트의 속성과 행위를 정의한다. 이 명세는 컴포넌트 군(family)의 멤버가 기대하는 요구사항들의 집합이다. 대부분의 경우 추출된 인터페이스 각각에 대해 별도의 컴포넌트 명세를 생성한다. 서로 다른 인터페이스에 의해 표현되는 내용이 동시에 생성되고 소멸되거나, 인터페이스간의 상호작용이 복잡하고 빈번하게 일어나거나, 그 상호작용을 위해서 많은 양의 데이터가 처리되어야 하는 경우에 하나의 컴포넌트 명세가 여러 개의 인터페이스를 지원한다.

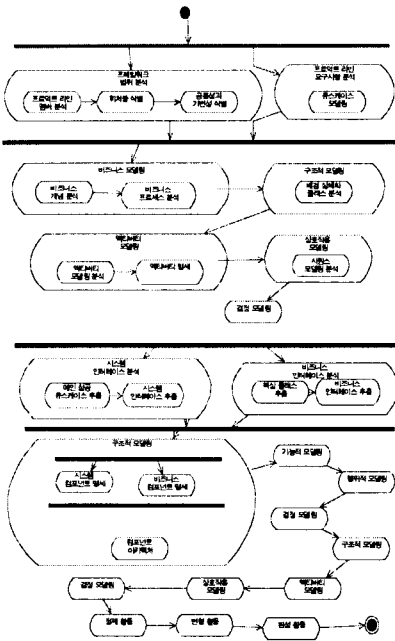
네 번째 단계인 컴포넌트 실체화 단계에서는 프레임워크 내에서 어떻게 컴포넌트가 명세를 만족시키는지를 정의하는 것과 컴포넌트가 어떻게 동작하는지를 정의한다.

마지막 컴포넌트 구현 단계는 컴포넌트 구체화 단계로 컴포넌트 모델링 단계의 산출물을 실행가능한 배포물로 만들어주는 단계이다.

[그림 3]은 Kobra 프로세스를 수정한 상세 프레임워크 공학 개발 프로세스를 나타내며, [그림 4]는 이러한 프레임워크 공학 프로세스를 나타낸 액티비티 다이어그램(activity)이다.



[그림 3] 상세 프레임워크 공학 프로세스



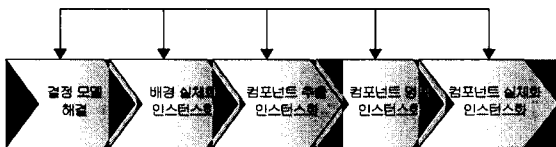
[그림 4] 프레임워크 공학 프로세스를 나타낸 액티비티 다이어그램

[그림 3]에서 기존의 Kobra 프로세스에 컴포넌트 추출 단계를 삽입한 것을 볼 수 있다. 컴포넌트 추출단계는 시스템 인터페이스 분석과 비즈니스 인터페이스 분석으로 나뉜다.

먼저 시스템 인터페이스 분석 단계는 시스템 인터페이스와 시스템 컴포넌트를 식별하기 위한 단계이며, 또 다시 메인 성공 유스케이스와 시스템 인터페이스를 추출한다.

비즈니스 인터페이스 분석 단계는 핵심적인 비즈니스 정보, 업무 규칙 그리고 변환을 구현하고 있으며 여러 시스템에서 재사용될 수 있다. 이 단계는 핵심 클래스 추출과 비즈니스 인터페이스를 추출한다.

[그림 5]는 애플리케이션 공학 프로세스를 나타내며, 보는 바와 같이 애플리케이션 공학은 결정 모델 해결, 배경 실체화 인스턴스화, 컴포넌트 추출 인스턴스화, 컴포넌트 명 인스턴스화, 컴포넌트 실체화 인스턴스화



[그림 5] 애플리케이션 공학 프로세스

첫 번째 단계인 결정 모델 해결 단계에서는 특정 애플리케이션 구축 시, 관련된 결정 모델들을 해결하는 단계로 프레임워크 공학에서 산출된 결정 모델들을 토대로 자신의 애플리케이션에 맞는 결정 모델의 가변성이 고려된다.

두 번째 배경 실체화 인스턴스화 단계에서는 프레임워크 공학에서 산출된 배경 실체화의 모든 산출물들을 특정 애플리케이션에 맞도록 인스턴스화 시킨다.

세 번째 단계인 컴포넌트 추출 인스턴스화 단계에서는 프레임워크 공학에서 산출된 컴포넌트 추출의 모든 산출물들을 특정 애플리케이션에 맞도록 인스턴스화 시킨다.

네 번째 컴포넌트 명세 인스턴스화 단계에서는 프레임워크 공학에서 산출된 컴포넌트 명세의 모든 산출물들을 특정 애플리케이션에 맞도록 인스턴스화 시킨다.

마지막 다섯 번째 단계인 컴포넌트 실체화 인스턴스화 단계에서는 프레임워크 공학에서 산출된 컴포넌트 실체화의 모든 산출물들을 특정 애플리케이션에 맞도록 인스턴스화 시킨다.

#### 4. 결론 및 향후연구

Kobra 프로세스는 컴포넌트 기반과 프로덕트 라인 공학의 개념을 모두 포괄하고 있는 장점이 있지만, 컴포넌트 추출에 관한 단계가 없이 컴포넌트의 개념을 사용하고 있어서 특정 도메인에 적합하지 않는 문제점이 있었다.

본 연구에서는 RUP, Catalysis, UML Components, Business Component Factory 등 여러 가지 컴포넌트 기반 개발 방법론 중 컴포넌트 기반의 개발 패러다임과 프로덕트 라인 공학의 패러다임을 통합하고 확장할 수 있는 방법인 Kobra 프로세스 방법을 컴포넌트 추출 단계의 상세화를 통해 수정하여 컴포넌트 기반 개발 및 프로덕트 개념을 포함한 애플리케이션 개발 프로세스를 정의하였다.

기존의 Kobra 프로세스에 컴포넌트 추출 단계를 추가시킬 뿐만 아니라 컴포넌트 추출이 원활하게 진행되도록 하기 위해 컴포넌트 추출 단계의 앞부분과 뒷부분 역시 특정 컴포넌트 기반 소프트웨어에 맞도록 수정하였다. 이렇게 함으로써 특정 도메인에 대한 컴포넌트 기반 소프트웨어를 구축함에 있어서 컴포넌트 추출상의 모호성이 존재하는 문제점을 해결할 수 있다는 데에 연구목적은 둔다.

향후 연구 방향으로는 컴포넌트 기반 개발 프로세스에 관련한 기존의 많은 방법론들을 검토함과 동시에 프로덕트 라인 기반 GNSS(Global Navigation Satellite System) 개발 프로세스를 실제 적용해 보는 것이 필요하며, 각 단계별로 상세한 지침의 제안이 필요하다.

#### 참고문헌

- [1] A. W. Brown and K.C. Wallnau, "The Current State of CBSE", IEEE Software, Vol. 16, No. 5, pp.37-46, Sept./Oct. 1998.
- [2] P.Brereton and D. Budgen, "Component-Based System: A Classification of Issues", IEEE Computer, Vol. 33, No. 11, pp. 54-92, Nov. 2000
- [3] John D. McGregor, Linda M. Northrop, Salah Jarrad, Klaus Pohl, "Initiating Software Product Lines", IEEE Software, Vol. 19, No. 4, pp. 24-31, July/August 2002.
- [4] 최미숙, 윤용익, 박재현, "RUP 기반의 컴포넌트 식별 방법에 관한 연구", 한국정보처리학회논문지, Vol. 9, No. 1, pp. 91~102, 2002.
- [5] John Dodd, "Identifying & Scoping CBD96 Components", Texas Instruments Inc., 1999.
- [6] Colin Atkinson, Joachim Bayer, Christian Bunse, Erik Kamsties, Oliver Laitenberger, Roland Laqua, Dirk Muthing, Barbara Peach, Jurgen Wust, Jorg Zettel, Barbara Paech, Component-Based Product Line Engineering with UML, Addison-Wesley, 2001.