도메인 컴포넌트 개발에서의 산출물간 추적성 검증 기법

장수호[0], 김수동
숭실대학교 컴퓨터 학과
shchang@otlab.ssu.ac.kr, sdkim@comp.ssu.ac.kr

Methods to Verify Traceability among Artifacts in Developing Domain Components

Soo Ho Chang[0], Soo Dong Kim
Dept. of Computing, Soongsil University

요약 (Abstract)

컴포넌트 기반 소프트웨어 개발 (CBD) 기술은 재사용 단위의 컴포넌트를 조합하여 소프트웨어를 개발함으로써 개발 노력과 상품화 시간을 줄여주는 새로운 기술로 정착되고 있다. 이러한 CBD에서의 재사용 단위로서 컴포넌트의 개발은 여러 활동(Activity)들을 통해 산출물들을 생성한다. 각 산출물은 이전 단계의 산출물을 기반으로 하므로 컴포넌트의 정확성을 보증할 산출물간의 추적성(Traceability)이 중요하다. 그러나 컴포넌트의 재사용성으로 그 정확성을 보증할 추적성이 더욱 중요함에도 이에 대한 연구가 미흡하다. 본 논문에서는, CBD 개발활동에서의 주요 산출물을 제시하고 그 관계에 따른 산출물 항목의 전이관계를 통한 일관성 검증, 즉 추적성을 검증하는 기법을 제안한다.

1. 서론

컴포넌트 기반 소프트웨어 개발 (CBD) 기술은 재사용 가능한 컴포넌트를 조합하여 효율적으로 소프트웨어를 개발함으로써 개발 노력과 상품화 시간을 줄여주는 새로운 기술로 정착되고 있다. 소프트웨어 개발은 여러 활동(Activity)들을 통해 산출물들을 생성한다. 각 산출물은 이전 단계의 산출물을 기반으로 하므로 산출물 간의 일관성과 정확성을 검증할 수 있는 추적성(Traceability)이 중요하다. 특히 CBD에서는 재사용성으로 컴포넌트의 정확성을 보증할 추적성이 더욱 중요시되나 이에 대한 연구가 미흡하다.

본 논문에서는, 주요 개발활동에서의 산출물을 제시하고 그 관계에 따른 산출물 항목의 전이관계를 통한 일관성 검증, 즉 추적성을 검증하는 기법을 제안한다.

논문의 2장에서는 기반연구로서 CBD, 도메인 컴포넌트와 가변치 가변점을 조사하고 3장에서는 컴포넌트 개발을 위한 활동 및 산출물을 제시하였으며 이를 기반으로 4장에서는 그들간의 관계에서 추적성 검증 기법을 제안한다.

2. 기반 연구

비즈니스 컴포넌트란 내부에 비즈니스 도메인에 관한 기능을 가지는 컴포넌트이다[1]. 이러한 비즈니스 컴포넌트는 제공해주는 기능의 레벨에 따라 두 가지 유형으로 구분된다. 비즈니스 개체를 직접 다루는 컴포넌트를 Fine-grained 컴포넌트라 하며, 도메인의 순수한 기능을 제공하는 컴포넌트를 Coarse-grained 컴포넌트라 한다.

가변성이란 공통적인 기능 안에서 각 어플리케이션이나 회사별로 요구되는 세부적인 차이점을 의미한다[2,3]. 이러한 가변성은 가변점(Variation Point)과 가변치(Variant)로 나누어질 수 있다.가변점이란 도메인에서 가변성이 발생하는 지점을 의미하며 주로 기능으로 표현된다. 또한 가변치란 가변점에서 기능을 요구하는 어플리케이션이나 회사가 다르게 요구하는 규칙이나 값들을 의미한다.

추적성이란 설계나 코드에서 요구사항으로 반대로 추적할 수 있는 성질을 의미한다[4]. 이러한 추적성은 개발되는 소프트웨어의 기능에 대한 정확성을 위한 품질 조건이다. 소프트 웨어 개발에서 추적성이란 한 개발 산출물에 대한 입력으로 사용한 이전 산출물의 연관 관계를 규정할 수 있는 정도를 나타낸다. 소프트웨어 개발에 대한 산출물 간의 규칙성을 정의하고 확인할 수 있는 규칙을 가짐으로 보다 정확한 일관성 있는 소프트웨어 개발이 가능해 진다.

3. CBD프로세스의 주요 활동 및 산출물

3장에서는 산출물간의 추적성 규칙을 도출하기 위해, 도메인 컴포넌트를 개발하는 주요 활동과 그에 따른 산출물을 소개한다.

3.1. 요구사항 정규화

도메인 컴포넌트를 개발하는데 있어서 어려운 점들 중 하나는 그 요구사항 명세서에서 사용되는 용어의 의미가 패밀리 멤버들간에 서로 다르다는 것이다. 이러한 요구사항의 패밀리 멤버간 이질성을 해결하기 위해 요구사항 정규화 단계에서는 패밀리 멤버들의 용어들을 추출하여 표준이 되는 용어를 정의한다.

표 1. 표준 용어 사전

표준용어	정의
CT_1	
:	:

그러한 표준 용어들을 표 1과 같이 표준용어 사전 형태로을 만들 수 있으며 이를 이용하여 각 패밀리 멤버의 요구사항 명세서를 재작성 한다.

3.2. 공통성 식별

도메인 컴포넌트로부터 패밀리 멤버가 기능성을 제공받기 위해 패밀리 멤버들간에 공통되는 기능성들이 추출되어야 한다. 공통 기능성을 추출하기 위한 방법으로 정규화된 패밀리 멤버의 요구사항 명세서로부터 표 2와 같은 기능 비교표를 제안한다.

표 2. 기능 비교 표

기능	멤버				공통성정도	공통기능 (Y/N)
	M_1	M_2	...	M_n		
F_1						
F_2						
...						
F_m						

'기능'열은 패밀리 멤버가 가지는 모든 기능의 합집합이다. 이러한 기능을 멤버가 가지는 경우 체크( )표시 한다. 이러한 체크개수나 도메인에서의 중요성 정도 도메인 표준 등을 고려하여 공통성 정도를 표시한다. 공통성 정도를 기반으로 공통기능 여부를 결정하며, 결정된 공통 기능들로 표 3과 같은 공통성 명세표를 작성 할 수 있다.

표 3. 공통성 명세 표

기능 ID	기능 명	설명
CF_1		
CF_2		
:		

3.3. 가변성 식별

공통성 식별 단계를 통해 추출된 공통성 내부에는 패밀리 맴버에 따라 조금씩 다르게 표현될 가변성들이 존재한다. 이러한 가변성은 가변적인 기능 타입에 따라 로직(Logic)또는 워크플로우(Workflow)의 타입으로 구별되며, 다르게 구현될 실제 값인 가변치를 가진다. 멤버에 따라 가변치가 정의된 경우도 있으나 아직 정의되지는 않았으나 예상 되어지는 경우도 있으므로 이는 Closed/Open으로 각각 표현한다. 이러한 가변성을 표 4과 같이 표현할 수 있다.

표 4. 가변성 설계표

가변점	가변성 타입	가변치집합	Open/ Closed	초기값	비고
CF_1	Logic	$\{V_{1.1}, V_{1.2}\}$	Closed	$V_{1.1}$	
CF_3	Workflow	$\{V_{3.1}, V_{3.2}\}$	Open	$V_{3.2}$	
...					
CF_n	Logic	{ }	Open	None	

3.4. 컴포넌트 모델링

이전 단계를 통해 얻어진 공통성과 가변성을 기초로 도메인 컴포넌트가 설계될 수 있다. 컴포넌트 모델링 단계에서는 그림 1과 같이, 여러 기준과 규칙으로 공통 기능을 그룹핑하여 컴포넌트를 추출하고, 그 컴포넌트에 식별된 가변성을 투영한다.

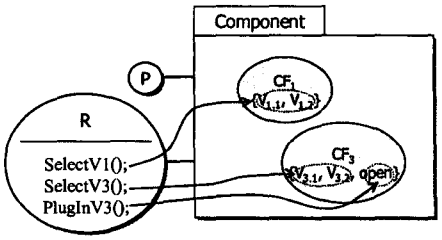

그림 1. 가변성 투영

컴포넌트는 외부로 그 기능 제공하기 위하여 Provide Interface를 가지며, 내부에 필요한 환경 설정을 위하여 Required Interface를 가진다. 가변성은 컴포넌트 내부에 멤버에 따라 다른 값이 설정되므로 Required Interface를 이용하며 가변치의 확정 유무에 다라 *Select()*또는 *PlugIn()*을 사용한다.

4. 산출물간의 추적성 검증

3장에서는 도메인 컴포넌트를 개발하기 위한 활동들과 그 결과로 생성된 산출물들을 소개하였다. 4장에서는 3장의 산출물들을 기초로, 산출물 내의 항목들이 개발 활동에 따라 전이되는 데 초점이 맞춰진 추적성 검증 기법을 제안한다.

도메인 컴포넌트를 개발하는데 있어서 주요 개발 활동에 대한 산출물의 관계를 그림 2와 같이 표현 할 수 있다.

그림 2에서, Mapping (1)은 멤버들의 이질적인 용어들로부터 만들어진 표준 용어가 사용자 요구사항 명세서에 반영되는 관계이다. 이 관계로부터 멤버들의 재 작성된 사용자 요구사항 명세서는 서로 비교 가능할 수 있게 된다.

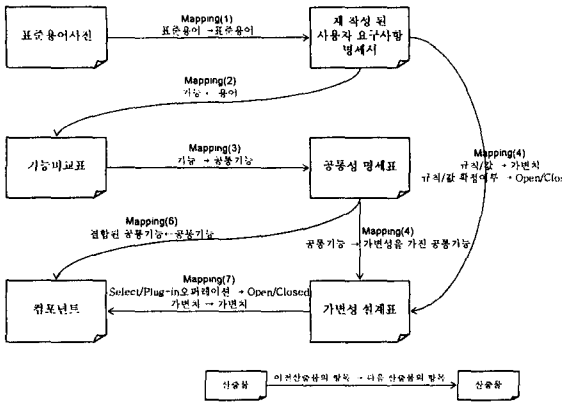

그림 2. 산출물간의 항목 전이도

Mapping (2)에서 모든 멤버들의 사용자 요구사항 명세서에 기술된 기능들은 기능 비교표의 기능 컬럼에 들어가게 된다. 작성된 사용자 요구사항의 기능을 보는 시각에 따라 기능의 크기가 다를 수 있으므로 요구사항의 기능이 기능 비교표의 여러 기능으로 표현될 수 있고 반대의 경우도 가능하다.

Mapping (3)에서는 기능 비교표에서 공통기능으로 결정된 기능들이 공통성 명세표에 기입된다. 또한 Mapping (4)에서는 공통성 명세표의 기능들 중 그 내부에 가변성을 가진 기능들만이 가변성 설계표의 가변점으로 전이된다

Mapping (5)는 재 작성된 사용자 요구사항 명세서내의 기능 들이 멤버들간에 서로 비교되어 추출된 정보가 가변성으로 설계되는 관계이다. 멤버별로 작성된 요구사항의 기능들은 사전 조건, 기능의 입출력 값, 기능방식, 또는 기준이 되는 규칙 등에 따라 기능성내에 가변성으로 추출될 수 있다. 따라서 사용자 요구사항 명세서 내에는 기능과 기능을 위한 규칙, 값 등은 가변성 테이블의 가변치 집합 중 한 요소로 대응된다. 또한 규칙, 값 등이 결정된 경우는 Open/Closed의 Closed로, 결정되지 않은 경우는 Open으로 내용이 전이된다.

Mapping (6)으로 공통성 명세표의 기능들은 그 성격에 따라 컴포넌트 단위로 그룹 지어질 수 있다. 공통성 명세표의 여러 기능들은 하나의 컴포넌트에 포함되게 되고, 또한 하나의 기능이 쪼개져서 두 개 이상의 컴포넌트에 포함될 수 있다. 특히 공통성에서의 기능들은 순수한 비즈니스 프로세스 즉 워크플로우와 사용되는 데이터를 함께 가지게 된다. 이러한 기능들은 비즈니스 프로세스와 데이터로 나뉘어 Coarse-grained컴포넌트와 Fine-grained 컴포넌트로 대응된다. Coarse-grained 컴포넌트의 경우 비즈니스 프로세스에 따라 여러 컴포넌트에 동시에 존재하는 기능도 존재할 수 있다. 또한 어느 컴포넌트에도 속하지 않는 기능도 존재할 수 있다. 후자의 경우는 가장 비즈니스 프로세스상 가까운 기능이 있는 컴포넌트에 포함되거나 유틸리티성 컴포넌트에 포함 될 수 있다.

Mapping (7)은 식별된 가변성이 컴포넌트 내부로 설계되는 관계이다. 설계를 위해 컴포넌트 인터페이스, 즉 Required Interface를 사용한다. 가변성 설계표의 Open/Closed의 구분에 따라 Required Interface의 Operation이 Open인 경우 *PlugIn()* 오퍼레이션으로 Closed인 경우 *Select()*오퍼레이션으로 설계된다. 또한 가변성 설계표의 가변치 집합은 *Select()* 오퍼레이션 내에 포함되며 포함된 가변치를 선택하는 값을 *Select()* 오퍼레이션의 매개변수로 받게 된다. 가변치가 설정되어 있지 않은 경우는 응용프로그램에서 사용하기 위해 컴포넌트를 설정하는 단계에서 *PlugIn()*의 매개변수로 가변치가 들어가게 된다.

5. 맺음말

본 논문에서는 도메인 컴포넌트를 개발하기 위한 CBD 프로세스의 주요 활동들과 활동의 결과로 생성되는 산출물들은 제안하였고, 그 산출물들의 정확성을 보증하기 위한 방법으로 산출물간의 추적성 관계를 정의하고 일관성을 검증 할 수 있는 기법을 제시 하였다. 제시된 추적성을 사용한 검증기법을 사용하면 산출물의 정확성과 일관성을 보증할 수 있다.

참고문헌

[1] Carey J., Carlson B., "Business Components", Chapter 16 of *Component-Based Software Engineering,* Addison Wesley, 2001.

[2] Atkinson C., Bayer J., Bunse C., Kamsties E., Laitenberger O., Laqua R., Muthig D., Paech B., Wüst J, Zettel J., *"Product Line Concepts"*, chapter 14 of *Component-based Product Line Engineering with UML,* Addison Wesley, 2001.

[3] Griss, M., "Product-Line Architectures", Chapter 22 of *Component-Based Software Engineering,* Addison Wesley, 2001

[4] McCall, J.A., Richards, P.K. and Walters, G.F. Factors in software quality, Vols I-III, Rome Air Development Centre, Italy 1977