

# 컴포넌트 가변성을 위한 Required 인터페이스 설계

박지영<sup>0</sup>, 김수동  
승실대학교 컴퓨터학과  
jypark@otlab.ssu.ac.kr, sdkim@comp.ssu.ac.kr

## A Method to Design Required Interface for Component Variability

Ji Young Park<sup>0</sup>, Soo Dong Kim  
Dept. of Computing, Soongsil University

### 요 약

컴포넌트 기반 개발은 재사용 단위의 컴포넌트를 사용하여 소프트웨어의 중복개발을 지양하고 시스템을 효율적으로 개발한다. 또한 인터페이스를 통해 컴포넌트 배치시 내부의 수정없이 사용자의 요구에 맞게 컴포넌트를 특화하는 장치를 제공한다. 본 논문에서는 컴포넌트 가변성(Variability)을 위한 컴포넌트 Required 인터페이스를 제시한다. 분석단계에서 컴포넌트의 공통성 및 가변성 식별이 이루어졌음을 가정하고, 분석단계의 산출물인 ‘가변성 식별 테이블’을 수집하여 가변성을 위한 오퍼레이션 식별 및 인터페이스를 정의한다. 또한 컴포넌트 Required 인터페이스 명세서를 제공한다. 따라서 제시된 기법은 Required 인터페이스를 구성하는 오퍼레이션의 식별부터 명세까지 체계적인 Required 인터페이스의 설계 프로세스를 제시하여 블랙박스 컴포넌트의 가변성 설정을 위한 실용적인 Required 인터페이스의 기반이 된다.

### 1. 서론

컴포넌트 기반 개발은 재사용 단위의 컴포넌트를 사용하여 소프트웨어의 중복개발을 지양하고 시스템을 효율적으로 개발하는 소프트웨어 개발 패러다임이다. 이는 컴포넌트 내부 구현을 캡슐화하고 정의된 인터페이스를 통해 환경과 상호작용한다. 또한 컴포넌트 배치시 내부의 수정없이 사용자의 요구에 맞게 컴포넌트를 특화하는 장치를 제공한다.

컴포넌트 도입 이후 컴포넌트 가변성 관리를 위한 인터페이스의 설계 지침에 대한 체계적인 방법이 요구되어 왔다. 그러나 이에 대한 연구가 거의 이루어지지 않아 컴포넌트의 확장성 등 고유한 장점을 충분히 발휘하지 못하고 있다. 따라서, 블랙박스 컴포넌트 내의 가변성이 식별되었다 하더라도 이를 설정해 주는 인터페이스 개발을 위한 시간 및 비용을 절감하기 위한 실용적인 설계의 연구와 방법이 필요하다.

본 논문에서는 컴포넌트 가변성을 설정하기 위한 컴포넌트 Required 인터페이스를 제시하며, 이를 위한 체계적인 설계 프로세스를 고안한다. 요구사항 분석단계에서 컴포넌트의 공통성 및 가변성 식별이 이루어졌음을 가정하여 2장에서는 가변성 설정 및 인터페이스에 대한 기존 연구를 고찰하며, 3장은 가변성 유형을 살펴본다. 4장에서는 패밀리 멤버의 특정 사용자가 요구하는 기능성을 위한 Required 인터페이스 설계 프로세스를 제안한다.

### 2. 관련 연구

Catalysis 방법론은 컴포넌트의 확장성을 위한 개발 프로세스와 산출물을 제시한다[1]. 컴포넌트 인터페이스는 일반적으로 사용되는 상위 인터페이스와 특화를 위한 하위 인터페이스가 명시된다. 하위 인터페이스는 컴포넌트의 특화를 관리하기 위해 플러그인 기법의 파라미터화 기법을 제시한다. 이 방법론은 가변성을 위한 인터페이스가 소개되었으나, 이에 대한 추가적인 정의와 세부적인 설계방법 및 명세가 요구된다.

Katharine 기법은 복잡한 컴포넌트 설계의 재사용성 및 확장성을 위한 개발 방법을 제시한다[2]. 컴포넌트는 커스터마이제이션 인터페이스를 통해 사용자의 설정 정보에 따라 특정 목적을 가진 컴포넌트로 확장된다. 이는 블랙박스 컴포넌트를 기반으로 하며 가변성이 설정되는 가변점에 따라 위임, 파라미터화, 상속 등으로 특화를 위한 가변치를 설정하는 방법을 제시한다. 이 기법은 커스터마이제이션 인터페이스 설계를 위한 상세한 지침이 요구되며, 명세 방법 또한 필요하다.

### 3. 컴포넌트 가변성 유형

같은 도메인에 속한 패밀리의 요구사항 명세서를 통한 공통성은 컴포넌트의 Provide 인터페이스로 식별 및 설계된다[2]. 그러나 공통성 중 각 패밀리 멤버에 맞는 특화를 요구하는 가변성이 존재하게 된다. 컴포넌트 가변성은 그림 1과 같이 논리

(Logic) 가변성과 워크플로우(Workflow) 가변성으로 정의된다[3]. 논리 가변성은 패밀리 멤버가 사용되는 공통성 중 오퍼레이션의 알고리즘과 로직이 변경되어 사용되는 경우를 말하며, 워크플로우 가변성은 동일한 목적을 수행하기 위한 컴포넌트 내부의 메시지 흐름이 경우에 따라 유동적인 흐름으로 제시될 수 있음을 의미한다.

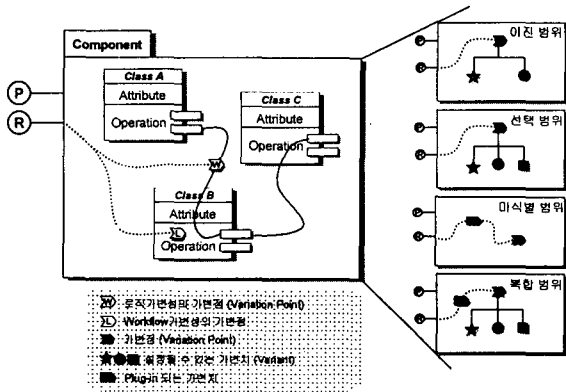


그림 1. 가변성 유형 및 범위

또한 가변성은 유형에 따라 식별 범위(Open Scope), 미식별 범위(Closed Scope)와 식별 및 미식별이 공존하는 복합 범위(Combined Scope) 내에 설정된다. 식별 범위는 컴포넌트에 존재하는 가변성의 경우에 따라 2개의 가변성이 존재하는 이진 범위(Binary Scope)와 3개 이상 존재하는 선택 범위(Select Scope)로 정의된다. 미식별 범위는 컴포넌트 내에 가변성이 정해지지 않고 컴포넌트 배치시 컴포넌트 외부에서 가변치를 설정한다. 복합 범위는 컴포넌트 내에 가변성이 정의되어 있거나, 부분적으로 외부로부터 가변성을 제공 받는 것을 말한다.

#### 4. 설계 프로세스

본 장에서는 컴포넌트 Required 인터페이스를 설계를 위해, 컴포넌트 개발 과정 중 컴포넌트의 공통성 및 가변성 분석 및 식별이 완료되었음을 가정한다.

##### 4.1. 가변성 오퍼레이션 식별

본 단계에서는 공통성과 가변성 분석단계에서 추출된 가변성 설정지점에 따른 가변치 및 가변성 유형을 전제로 하여 Required 인터페이스를 구성하는 오퍼레이션을 식별하는 단계이다. 가변성 오퍼레이션 식별을 위한 단계는 다음과 같다.

1. 완료된 컴포넌트 개발 요구사항 분석단계의 컴포넌트 가변성 관련 산출물인 '가변성 설계 테이블'(표 1 참조)을 수집한다.

2. '가변성 설계테이블'에 따라 가변성 설정을 위한 오퍼레이션을 찾는다. 즉, Provide 인터페이스 실행 시 참조되는 가변성을 정의한다.

3. Required 인터페이스에 할당되는 오퍼레이션을 위한 '가변성 오퍼레이션 할당 테이블'(표 2 참조)을 작성한다.

그림 2는 공통성을 제공하는 Provide 인터페이스 중 가변점에서 가변치 설정을 위한 Required 인터페이스가 유도됨을 나타낸다.

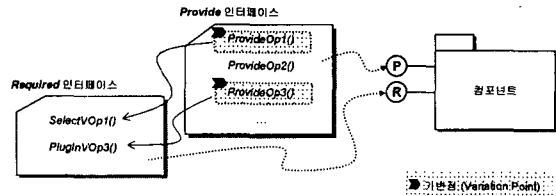


그림 2. 가변성을 위한 Required 인터페이스

가변점에 따라 Required 인터페이스를 구성하는 가변성 오퍼레이션 식별을 위해 표 1과 같은 '가변성 설계 테이블'을 요구한다. 컴포넌트 분석 단계의 산출물인 '가변성 설계 테이블'은 가변점, 가변성 유형 및 범위, 가변치 집합 및 초기값을 정의한다.

표 1. 가변성 설계 테이블

가변점	가변성 유형	가변치 집합	초기 값	가변성 범위	설정내용
VP <sub>1</sub>	논리	{V <sub>1</sub> , V <sub>2</sub> }	V <sub>1</sub>	식별	고유번호
VP <sub>2</sub>	워크플로우	{V <sub>6</sub> , Open}	V <sub>6</sub>	복합 (식별+미식별)	이차계산
VP <sub>3</sub>	논리	{V <sub>3</sub> , V <sub>4</sub> , V <sub>5</sub> }	V <sub>3</sub>	식별	...
VP <sub>4</sub>	워크플로우	{ }	없음	미식별	...

가변성 오퍼레이션은 표 1의 각 가변점으로부터 가변성 유형 및 가변성 범위에 따라 추출되며, 이는 가변점을 유도하는 Provide 인터페이스를 기본 인터페이스로 한다. 즉, 표 2와 같이 기본 인터페이스에는 하나 이상의 가변성 오퍼레이션이 할당되며, 오퍼레이션 ID로 구분하였다.

표 2. 가변성 오퍼레이션 할당 테이블

오퍼레이션 ID	기본 인터페이스	가변점	가변성 유형	가변치 집합	초기 값	가변성 범위
01	IInterface name <sub>1</sub> ()	VP <sub>1</sub>	논리	{V <sub>1</sub> , V <sub>2</sub> }	V <sub>1</sub>	식별
02	IInterface name <sub>2</sub> ()	VP <sub>2</sub>	워크플로우	{V <sub>6</sub> , Open}	V <sub>6</sub>	복합
03	IInterface name <sub>3</sub> ()	VP <sub>3</sub>	논리	{V <sub>3</sub> , V <sub>4</sub> , V <sub>5</sub> }	V <sub>3</sub>	식별
04		VP <sub>4</sub>	워크플로우	{ }	없음	미식별

##### 4.2. 가변성 유형별 인터페이스 설계

본 단계에서는 4.1에서 식별된 가변성 유형별 Required 인터페이스 설계를 수행한다. 인터페이스의 유형별 설계지침은 다음과 같다.

1. '가변성 오퍼레이션 할당 테이블'로부터 가변성 유형에 따라 Required 인터페이스를 위한 오퍼레이션의 네이밍 규칙을 정의한다.

2. 정의된 *Required* 오퍼레이션의 기능성의 범주에 따라 *Required* 인터페이스로 정제한다.

가변성 유형에 따라 사용되는 *Required* 인터페이스의 오퍼레이션은 *Select()*와 *PlugIn()*의 네이밍(Naming) 규칙을 적용한다. 만일, 사용자가 요구한 가변치가 컴포넌트 내 장착되어 있는 식별 범위일 경우, 가변치를 입력받아 영구적으로 설정하는 *Select()* 함수를 사용한다. 그러나 미식별일 경우, 외부에서 참조되는 함수나 객체를 가변치로 입력받는 *PlugIn()* 함수를 사용한다. 가변성 범위가 복합일 경우, *Select()* 및 *PlugIn()* 함수 모두를 사용하게 된다.

식별 범위 내 존재하는 논리 가변성의 오퍼레이션은 ‘가변성 오퍼레이션 할당테이블’에서 가변치 집합을 고려한다. 즉, 동일한 기능을 수행하지만, 실제 설정값에 따라 오퍼레이션의 파라미터값이 다르다. 이는 컴포넌트 내부에서는 함수의 몸체가 다른 가변성을 구현하게 된다. 식별 범위 내 워크플로우 가변성의 오퍼레이션은 사용자가 원하는 시스템을 구현하기 위한 워크플로우를 특화할 수 있는 오퍼레이션으로 논리 가변성 오퍼레이션과 동일한 방법으로 실제 설정값을 입력받아 컴포넌트 내부에 가변적인 워크플로우를 가지는 함수를 설정한다. 식별 범위 내 논리 및 워크플로우 가변성 오퍼레이션의 BNF 형식은 다음과 같다.

```
Logic Type ::=
    "Select"<Variation Point Name>(<Variant Type><Actural Argument>);
```

```
Workflow Type ::=
    "Select"<Variation Point Name>(<Variant Type><Actural Argument>);
```

미식별 범위의 가변성 오퍼레이션은 ‘가변성 오퍼레이션 할당테이블’에서 가변치에 대한 식별이 이루어지지 않았다. 이는 멤버간의 요구사항이 매우 다양하거나, 확장 가능성이 존재하는 경우 컴포넌트 내부에서 가변성을 고려하여 설정 내용만 선언한 후 외부에서 해당 클래스 또는 메소드를 Plug-in 할 수 있게 된다. 따라서, 외부에서 제공되는 파라미터 객체를 실제 파라미터 설정값으로 입력받는 네이밍 규칙을 갖게 된다. 미식별 범위 내 논리 및 워크플로우 가변성 오퍼레이션의 BNF 형식은 다음과 같다.

```
Logic Type ::=
    "PlugIn"<Variation Point Name>(<Variant Class Type><Actural object>);
```

```
Workflow Type ::=
    "PlugIn"<Variation Point Name>(<Variant Class Type><Actural object>);
```

‘가변성 오퍼레이션 할당 테이블’로부터 정의된 가변성 오퍼레이션 집합은 *Provide* 인터페이스를 기본 인터페이스로 하여 *Required* 인터페이스를 구성한다. 즉, 가변성을 유도하는 *Provide* 인터페이스에서 하나의 *Required* 인터페이스가 유도되며, 기본 인터페이스가 복수개의 가변점을 요구할 경우, 복수

개의 가변성 오퍼레이션으로 이루어진 *Required* 인터페이스가 정의된다.

### 4.3. Required 인터페이스 명세

컴포넌트 *Required* 인터페이스는 지금까지 식별 및 설계된 가변성 오퍼레이션의 집합이다. 그림 3은 *Required* 인터페이스의 명세서를 나타낸다. 인터페이스 명세서에는 *Provide* 인터페이스와 동일하지만 “IR”을 접두사로 요구하는 *Required* 인터페이스의 이름과 인터페이스의 계층 및 가변성 오퍼레이션의 사전/사후 조건과 가변성의 정보가 기록된다.

인터페이스 이름 :	"IR"<Interface Name <sub>i</sub> >
인터페이스 계층 :	시스템 서비스 계층
인터페이스 유형 :	<i>Required</i>
시그네처 :	"Select"<Variation Point Name>(<Variant Type><Actural Argument>)
사전조건 :	<i>Pre_Condition</i>
사후조건 :	<i>Post_Condition</i>
불변조건 :	<i>Constraint_Condition</i>
기본 인터페이스 :	"IP"<Interface name <sub>i</sub> >()
범 위 :	<i>Closed</i>
가변성 유형 :	<i>Logic</i>
매개변수 :	<i>int a</i>
설 명 :	.....

그림 3. Required 인터페이스 명세서

### 5. 결론

본 논문에서는 컴포넌트 인터페이스 중 가변성을 위한 *Required* 인터페이스의 설계 프로세스를 제시하였다. 제시된 기법은 컴포넌트 공통성 및 가변성이 분석되었음을 전제로 기술하였다. 분석 산출물을 통하여 가변성 오퍼레이션을 BNF 형식의 정규언어로 정의하여 이해를 도왔으며, 가변성 오퍼레이션으로 구성된 *Required* 인터페이스 설계 및 명세를 위한 지침을 제공하였다. 컴포넌트 자체를 수정하지 않고 컴포넌트 배치 시 사용자의 요구에 맞게 컴포넌트를 특화하는 *Required* 인터페이스 설계 및 명세함에 따라 향후 특정 어플리케이션 등에 적용될 수 있는 *Required* 인터페이스 설계 등에 활용될 것으로 기대된다.

### 참고문헌

- [1] D'souza, *Objects, Components and Frameworks with UML*, Addison-Wesley, 1999
- [2] Whitehead, K., *Component-based Development: Principles and Planning for Business Systems*, Addison Wesley, 2002.
- [3] 소동섭, 신석규, 김수동, “컴포넌트 가변성 유형 및 Scope에 대한 정형적 모델”, 한국정보과학회논문지 소프트웨어 및 응용, Vol. 30, No. 05, pp. 414-429, 2003년 6월.