

데이터 가변성 기반 모바일 자가적응 커스터마이제이션 기법

Mobile Self-Adaptable Customization Technique based on Data Variability

김철진

인하공업전문대학 컴퓨터시스템과

Chul-Jin Kim(cjkim@inhac.ac.kr)

요약

기존 객체지향 개발 방식과 컴포넌트 개발 방식에서의 재사용성 문제는 모바일 개발에서도 해결해야 하는 문제이다. 모바일 어플리케이션에서 기능을 변경하는 것은 모바일 어플리케이션 전체를 업그레이드 할 필요가 있는데, 이는 모바일 어플리케이션이 디바이스 내의 다른 어플리케이션들과 연계된 서비스와의 문제를 발생시킨다. 따라서 모바일 어플리케이션 내에 변경 가능성이 있는 기능들에 대해 재사용성을 고려하여 개발되어야 한다. 본 논문에서는 모바일 어플리케이션이 다양한 요구사항에 대해 재사용성을 높일 수 있는 커스터마이제이션 기법을 제안한다. 가변성 중에 데이터 기반의 가변성을 적용하여 변경을 요구하는 기능에 대해 자동으로 변경될 수 있는 자가적응 커스터마이제이션 기법을 제시한다.

■ 중심어 : | 커스터마이제이션 | 자가적응 | 데이터가변성 | 모바일 서비스 |

Abstract

The reusability problems in the existing object-oriented development method and component development method is the problem that should be also resolved in the mobile development. Changing the function in mobile applications need to upgrade the entire mobile application, which can cause problems in mobile application in conjunction with other applications within the device. Thus, the changable functions in mobile application should be developed by considering their reusability. In this paper, we propose the customization technique that can be improved reusability for a variety of requirements of mobile application. We propose the self-adaptable customization technique that can change dynamically for the function to require changes by applying the data variability among variabilities.

■ keyword : | Customization | Self-Adaptable | Data Variability | Mobile Service |

I. 서론

최근 모바일 어플리케이션의 사용이 증가함에 따라 모바일 어플리케이션 개발 방식 및 아키텍처에 대한 연구가 활발하게 이루어지고 있다. 기존 객체지향 및 컴포넌트 개발 방식에서 이슈화되었던 재사용성[1] 문제

들이 모바일 개발에서도 문제가 됨에 따라 이에 적합한 해결 방안이 요구되고 있다[2]. 모바일 어플리케이션의 기능에 대한 변경은 어플리케이션 전체를 업그레이드 해야 하는 상황을 자주 발생시키는데, 이는 모바일 어플리케이션이 디바이스 내의 다른 어플리케이션들과 연계된 서비스와의 문제를 발생시킨다. 따라서 모바일

어플리케이션 내에 변경 가능성이 있는 기능들에 대해 재사용성을 고려하여 개발되어야 하며, 기존 다른 어플리케이션과 연계된 기능에 대해 변경 가능성을 보장해 주어야 한다.

본 논문에서는 모바일 어플리케이션이 다양한 요구 사항에 대해 재사용성을 높일 수 있는 커스터마이제이션 기법을 제안한다. 다양한 가변성 중에 데이터 기반의 가변성[3]을 적용하여 변경을 요구하는 기능에 대해 자동으로 변경될 수 있는 자가적응 커스터마이제이션 기법을 제시한다.

본 논문은 다음과 같이 구성된다. 2장에서는 관련연구로 재사용 관련기술, 안드로이드 어플리케이션 프레임워크에 대해 연구하며, 3장에서는 데이터 가변성 기반 모바일 서비스 커스터마이제이션 기법을 제안한다. 4장에서는 본 논문에서 제안한 기법을 사례에 적용하여 타당성을 검증하며, 5장에서 결론을 맺는다.

II. 관련 연구

본 논문은 모바일 기반의 가변성을 커스터마이제이션하기 위한 기법을 제안하므로 기존 연구의 재사용 기술 및 데이터 가변성 연구에 대해 알아본다. 또한 모바일 플랫폼 중 안드로이드 플랫폼의 가변성 기술에 대해 알아본다.

1. 일반성 향상을 위한 가변성 설계 기법 및 커스터마이제이션 기법

연구 [4]는 행위 가변성에 대한 설계 및 커스터마이제이션 기법을 제안하고 있다. 행위를 커스터마이제이션하기 위해 기능 추가나 기능 변경을 위한 설계 방안을 제시하며, 기존 클래스를 교체하여 행위를 변경할 수 있다. 이러한 기법들을 설계 기법 측면에서 제안하고 있으므로 동적인 커스터마이제이션 기법도 고려될 수 있다. 연구[4]은 다른 방법론(Catalysis, Componentware, 등)들과 비교했을 때, 가변성 추출 및 가변성 설계 기법을 제시하고 있으며, 인터페이스 추가나 인터페이스 변경, 클래스 교체를 통해 행위를 커스

터마이제이션 할 수 있음을 차별성으로 제시하고 있다. 그러나, 연구[4]은 데이터 가변성 및 모바일 측면에서의 커스터마이제이션에 대한 연구가 고려되고 있지 않다.

2. 객체지향 기반 개발에서 재사용성 컴포넌트 설계 방법

연구[5]는 재사용 가능한 컴포넌트를 설계하기 위한 기법을 제안하고 있다. 분석 단계와 설계 단계로 구분하여 재사용 컴포넌트를 설계하기 위한 기법들을 정의하고 있다. 특히 설계 단계의 커스터마이징 포인트 식별 단계에서는 잦은 변경이 발생할 가능성이 있는 부분을 추출하여 컴포넌트 인터페이스를 정의한다. 이렇게 정의된 인터페이스를 통해 컴포넌트 내의 객체들 간의 제어 흐름을 정의한다. 다음 단계로 컴포넌트 내의 객체들의 상태 변화를 설계하기 위해 상태머신 다이어그램을 설계하며 상호작용에 대해 순차 다이어그램을 이용하여 설계한다. 연구[5]는 가변성 추출 단계를 강조하여 재사용 컴포넌트 설계가 가능함을 제안하고 있다. 그러나, 구체적인 가이드를 제시하지 못하고 있으며, 설계 기법 측면에서 제안하고 있으므로 실현 가능성에 대한 검증이 요구된다.

3. 온톨로지 기반 데이터 가변성 처리 기법

연구[6]은 DTT 컴포넌트 모델(Data Type-Tolerant Component Model)을 통해 서비스 컴포넌트와 비즈니스 데이터 간에 직접적인 결합을 없애므로써 서비스 컴포넌트들이 수정되지 않고도 새로운 비즈니스 데이터들로 변경할 수 있는 방안을 정의하고 있다. 서비스 컴포넌트는 SCDT(Self-Contained Data Type)와 비즈니스 엔티티에 대한 메타데이터 온톨로지를 포함하며, 데이터 타입 컨버터(Data Type Converter)를 이용하여 비즈니스 엔티티를 접근하기 위한 코드를 생성한다. 따라서 서비스 컴포넌트는 SCDT, 메타데이터 온톨로지, 그리고 데이터 타입 컨버터에 의해 다양한 데이터 인터페이스를 생성할 수 있다. 이렇게 생성된 인터페이스를 통해 실제 비즈니스 데이터로 연결될 수 있다. 연구[6]은 서비스 컴포넌트와 데이터 간의 결합력을 줄여 서비

스 컴포넌트가 다양한 데이터로 변경을 하기 위한 데이터 가변성 중심의 연구를 수행하였다. 이에 반해 본 논문에서는 데이터 가변성을 기반으로 다양한 서비스 컴포넌트로 변경될 수 있는 방안을 모바일 환경에서 제시한다.

4. 안드로이드 기반의 웹 서비스 프로토콜 커스터마이제이션 기법

연구[7]은 모바일 어플리케이션이 웹 서비스를 사용하기 위해 다양한 웹 서비스 프로토콜을 커스터마이제이션하기 위한 기법을 제안한다. 기존 웹 서비스 프로토콜을 사용하는 모바일 어플리케이션을 변경할 경우 어플리케이션 전체를 재설치하는 것이 아니라, 기존 서비스는 유지하면서 새로운 프로토콜 서비스를 플러그인하여 서비스를 커스터마이제이션 할 수 있다. 그러나 연구[7]은 가변성 식별자를 변경하여 명시적으로 서비스를 커스터마이제이션 해야 한다.

5. 모바일 환경에서 콘텐츠 추천 시스템 설계 및 구현

연구[8]은 모바일 대량 콘텐츠 중에서 요구되는 콘텐츠를 추천하기 위한 시스템 설계 및 구현 방안을 제안한다. 상황정보를 이용하여 콘텐츠를 검색하고 가중치를 적용하여 콘텐츠를 추천한다. 연구[8]은 콘텐츠를 상황정보에 의해 추천하는 것이며, 본 논문의 데이터 가변성 정보를 통해 적합한 모바일 서비스를 커스터마이제이션하기 위한 연구는 미흡하다.

6. 모바일 환경을 위한 지능형 추천 에이전트에 관한 연구

연구[9]는 모바일 환경에서 에이전트를 이용하여 사용자에게 능동적으로 정보를 제공할 수 있는 시스템을 제안한다. 프로파일 모듈, 규칙생성 모듈, 필터링 모듈, 서비스 모듈로 구성된다. 추천 에이전트를 이용하여 사용자의 요구 사항에 대해 파악 가능하도록 구성한다. 연구[9]는 에이전트에 의한 상황인지 기반 서비스를 제공하는 것으로, 모바일 서비스의 커스터마이제이션에

대한 연구는 제한적이다.

7. 재사용 기술 비교

[표 1]에 나타난 바와 같이 재사용 기술에는 포함, 상속, 매개화, 동적 클래스 로딩 등과 같은 다양한 기술들이 존재하며, 다양한 기술들이 Kobra[10], FAST[11], FODA[12], UML Component[13], Catalysis[14], Componentware[15], Fusion[16]과 같은 기존 컴포넌트 방법론에서는 어떻게 재사용을 위해 제공하고 있는지 비교한다. [표 1]에 나타난 바와 같이 기존 방법론들에서는 필요성에 대한 언급이나 정의는 하고 있지만 구체적인 기법이나 실행 가이드를 제시해 주지 못하고 있다. 일부 방법론들은 객체지향 기술(포함, 상속, 매개화, 등)에 대해 설계 패턴형태로 제시하고 있지만, 구체적인 실행 가이드를 제시하지 못하고 있다.

표 1. 재사용 기술 비교

Methodology	Kobra	FAST	FODA	UML Comp.	Catalysis	Comp.ware	Fusion
Reusability Technology							
Aggregation (Delegation)			S		S		
Inheritance		S	S		M	S	
Parameterization		S	S				
Dynamic Class Loading							
Plug-In							S
Templates					M		S
Overloading (Polymorphism)					M		
Static Library							
DLL (Dynamic Link Library)		S					
Conditional Compilation							
Reflection							
AOP (Aspect-Oriented Prog.)							
Connector					M		
Adaptor						S	
Wrapper						S	
Mediator							
Proxy							
Re-implementation							

S: Supported, M: Mentioned Only, Blank: Not Supported

본 논문에서는 이러한 기법들에 비해 실행 가능한 형태로 사용될 수 있는 구체적인 가이드를 제시하고 있으며, 모바일 어플리케이션의 잦은 변경에 대한 동적인 커스터마이제이션이 가능할 수 있는 기법을 제시하고 있다.

8. 안드로이드 어플리케이션 프레임워크

안드로이드 프레임워크[17]에서 인텐트 필터(Intent Filter)는 안드로이드 모바일 어플리케이션의 기능을 가

변적으로 처리할 수 있도록 제공되는 메타정보이다. 가변적으로 처리되어야 할 기능에 대해 인텐트 필터로 정의하면 안드로이드 프레임워크 내의 패키지관리자(Package Manager)가 인텐트 필터를 인지하여 해당하는 기능을 가변적으로 처리해 준다. 기능에 흐름을 변경해야 하는 경우 코드를 변경하는 것이 아니라 인텐트 필터를 수정한다. 모바일 어플리케이션 내의 액티비티(Activity)간에 데이터의 전달은 인텐트(Intent)라는 객체 내에 포함되어 전달되는데, 이러한 인텐트에 대한 처리는 액티비티 관리자(Activity Manager)가 담당한다. 본 논문에서는 데이터 가변성을 적용하기 위해 인텐트 필터와 인텐트를 기반으로 한다.

9. 가변성

가변성(Variability)이란 같은 종류의 컴포넌트들 간에 서로 다른 처리를 하는 기능을 가변성이라고 한다 [18]. 컴포넌트가 사용될 때 다양한 요구사항이 발생하기 때문에 그럴 때 마다 컴포넌트를 수정해야 한다면 효율성이 저하될 것이다. 따라서 다양한 요구사항을 만족시킬 수 있도록 가변성을 고려하여 컴포넌트를 개발해야 한다. 가변성은 다양한 측면에서 고려될 수 있는데, 예를 들면, 기능에 대한 가변성[19]이나 기능 흐름에 대한 가변성[20]을 고려할 수 있으며 데이터 타입이나 데이터 유무에 따라 처리를 다르게 할 수 있는 데이터 가변성을 고려할 수 있다. 본 논문에서는 모바일 환경에서 데이터의 특성에 따라서 기능이 가변적으로 처리될 수 있는 데이터 가변성을 기반으로 한다.

III. 데이터 가변성 기반 커스터마이제이션 기법

이 장에서는 본 논문에서 제안하는 모바일 커스터마이제이션을 위한 모바일 데이터 가변성과 자가 적응 커스터마이제이션 기법을 제안한다.

1. 모바일 데이터 가변성

데이터 가변성은 데이터의 존재 유무에 따라 기능이 변경되거나, 데이터의 타입에 따라 기능이 변경되는 경

우로 분류할 수 있다.

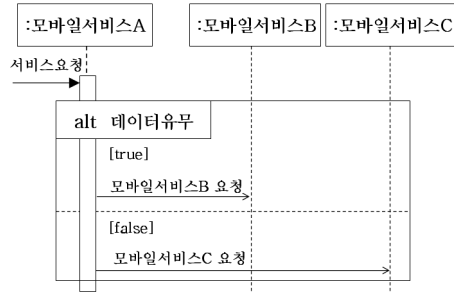


그림 1. 데이터 유무에 따른 커스터마이제이션

데이터 존재 유무에 따라 기능이 커스터마이제이션 되는 경우는 [그림 1]에서와 같이 ‘모바일서비스A’를 제공하기 위해 데이터 유무에 따라 ‘모바일서비스B’ 나 ‘모바일서비스C’로 변경될 수 있다. ‘모바일서비스B’는 존재하는 데이터와 관련된 기능을 제공하며 ‘모바일서비스C’는 데이터가 존재하지 않은 상태의 서비스를 제공한다.

데이터 타입에 따라 모바일 서비스가 커스터마이제이션되는 경우는 [그림 2]에서와 같이 ‘모바일서비스A’는 데이터 타입이 ‘String’ 또는 ‘URI’에 따라 ‘모바일서비스B’나 ‘모바일서비스C’로 변경될 수 있다. ‘모바일서비스B’는 ‘String’ 데이터 타입과 관련된 서비스를 제공하며, ‘모바일서비스C’는 ‘URI’ 데이터 타입과 관련된 서비스를 제공한다.

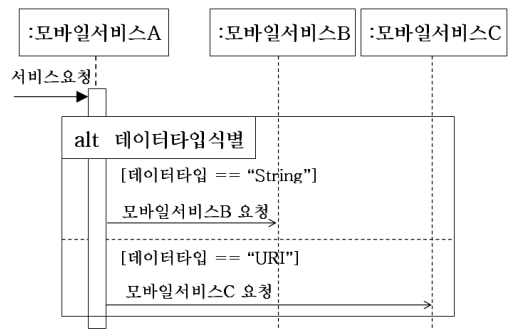


그림 2. 데이터 타입에 따른 커스터마이제이션

데이터 타입은 [그림 3]에서와 같이 하나의 데이터

타입에 의해 서비스가 변경될 뿐만 아니라, 데이터 타입의 조합에 의해서도 서비스를 변경할 수 있다.

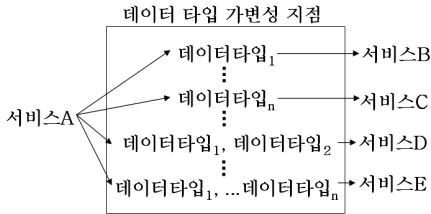


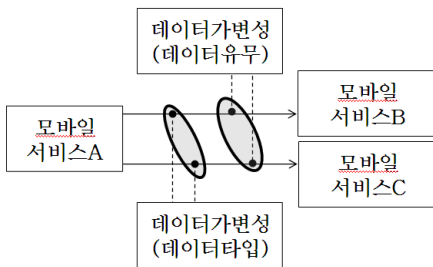
그림 3. 데이터 타입의 조합에 따른 커스터마이제이션

데이터 가변성은 데이터 존재 유무와 데이터 타입에 따라 모바일 서비스를 커스터마이제이션 할 수 있다. 행위 가변성 및 워크플로우 가변성이 가변성 식별자에 의해 명시적으로 모바일 서비스를 커스터마이제이션 할 수 있는 방식인 것에 반해, 데이터 가변성은 모바일 서비스들 간에 서비스 호출을 자동적으로 커스터마이제이션 할 수 있는 기반을 제공한다.

2. 자가 적응 커스터마이제이션 기법

데이터 가변성 기반의 자가적응 커스터마이제이션을 하기 위한 기본 구조는 [그림 4]와 같다. '모바일 서비스 A'에서 데이터 가변성에 의해 '모바일 서비스B'나 '모바일 서비스C'로 자동으로 변경될 수 있다.

데이터 가변성에 의한 서비스 자동 변경은 3.1절에서 정의한 것처럼 데이터 유무나 데이터 타입에 따라 가변성이 적용될 수 있다.



○: 데이터 가변성 지점

그림 4. 자가적응 커스터마이제이션 구조

[그림 4]의 자가적응 커스터마이제이션 구조에 대해 모바일 기반 환경에서의 클래스 다이어그램[21]을 [그림 5]와 같이 정의할 수 있다.

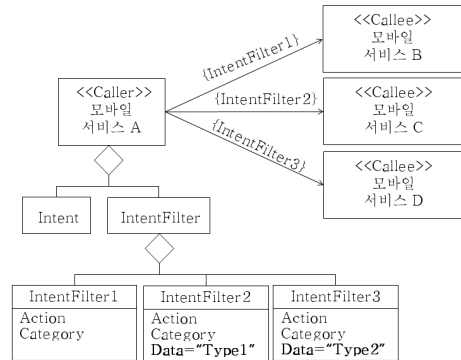


그림 5. 모바일 자가적응 커스터마이제이션 기법 클래스 다이어그램(안드로이드 기반)

'모바일 서비스A'는 인텐트와 인텐트 필터로 구성된 다. 인텐트는 전달되는 데이터를 포함하고 있으며, 인텐트 필터는 데이터 가변성을 제공하기 위한 설정정보를 포함하고 있다. '인텐트 필터1'은 데이터를 정의하지 않으며 기본 데이터 타입(문자열)을 허용한다. '인텐트 필터2'와 '인텐트 필터3'는 데이터 타입을 각각 정의하여, 'Data' 속성값이 'Type1'이나 'Type2'에 따라 '모바일 서비스C'나 '모바일 서비스D'로 자가적응 될 수 있다.

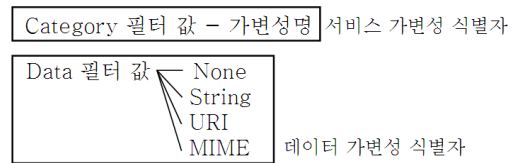


그림 6. 가변성을 위한 인텐트 필터

자가적응 커스터마이제이션을 위한 가변성 정의는 [그림 6]과 같이 인텐트 필터를 통해 정의할 수 있다. 전달되는 데이터는 인텐트에 포함하며 이렇게 전달되는 데이터의 타입은 인텐트 필터에 정의한다. 인텐트 필터는 우선적으로 서비스 가변성('가변성명')을 통해 일치하는 모바일 서비스로 변경을 할 수 있으며, 추가적으로 데이터 가변성(데이터 유무 또는 데이터 타입)을 통

해 자동적으로 서비스를 변경할 수 있다. 데이터 타입 중에 'String', 'URL', 'MIME' 타입을 지정하여 일치하는 모바일 서비스로 변경할 수 있다. 데이터 유무에 대한 가변성은 타입을 지정하지 않으면 되며, 호출하는 모바일 서비스에서 데이터의 유무에 따라 가변적으로 서비스를 호출한다.

```
public class 모바일서비스A extends Activity {
    public void onCreate(Bundle savedInstanceState) {
        ...
        myButton.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                Intent i = new Intent();

                i.setAction(Intent.ACTION_MAIN);

                i.addCategory("가변성명");
                i.putExtra("Data_1", "testData2"); // String 타입
                i.putExtra("Data_2", "testData2"); // String 타입

                startActivity(i);
            }
        });
    }
}
```

그림 7. 데이터 유무에 따른 가변적 서비스 호출 코드

데이터 유무에 따른 서비스 호출은 [그림 7]의 코드와 같이 구현할 수 있다. 서비스 가변성('가변성명')에 대해 정의하고, 데이터는 기본 데이터 타입(문자열)을 전달한다. 서비스 가변성에 의해 여러 서비스가 선택될 수 있으며, 문자열 데이터를 전달 받을 수 있는 서비스가 자동으로 선택된다.

```
<activity android:name="모바일서비스B" >
<intent-filter>
    ...
    <action android:name="android.intent.action.MAIN" />
    <category android:name="android.intent.category.DEFAULT" />
    <category android:name="가변성명" />
    <!-- 데이터타입을 명시하지 않으면 String 데이터 타입으로 filtering -->
</intent-filter>
</activity>
<activity android:name="모바일서비스C" >
<intent-filter>
    ...
    <category android:name="가변성명" />
    <data android:scheme="http" /> <!-- 다른 형태의 데이터 타입 정의 -->
</intent-filter>
</activity>
```

그림 8. 데이터 유무에 따른 가변적 서비스 호출 설정

데이터 유무에 따른 가변적 서비스 호출을 위한 설정 코드는 [그림 8]과 같다. 2개의 모바일 서비스 중에 데이터 타입을 정의하지 않은 '모바일서비스B'가 [그림 7]의 코드에 대해 자동으로 선택될 수 있는 서비스가 될 것이다. '모바일서비스C'는 동일한 서비스 가변성을 정

의하고 있지만, 다른 형태의 데이터 타입을 정의했으므로 데이터 가변성에 의해 호출되지 않는다.

```
public class 모바일서비스A extends Activity {
    public void onCreate(Bundle savedInstanceState) {
        ...
        myButton.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                Intent i = new Intent();

                i.setAction(Intent.ACTION_MAIN);

                i.addCategory("가변성명");
                Uri uri = Uri.parse("http://www.google.com");
                // URI 데이터 타입
                i.setData(uri);

                startActivity(i);
            }
        });
    }
}
```

그림 9. 데이터 타입에 따른 가변적 서비스 호출 코드

데이터 타입에 따른 서비스호출은 [그림 9]의 코드와 같이 구현할 수 있다. 서비스 가변성('가변성명')에 대해 정의하고, 데이터는 'URI'를 전달한다. 서비스 가변성에 의해 여러 서비스가 선택될 수 있으며, 'URI' 데이터를 전달 받을 수 있는 서비스가 자동으로 선택된다.

```
<activity android:name="모바일서비스B" >
<intent-filter>
    ...
    <category android:name="가변성명" />
    <data android:scheme="http" /> <!-- 다른 형태의 데이터 타입 정의 -->
</intent-filter>
</activity>
<activity android:name="모바일서비스C" >
<intent-filter>
    ...
    <category android:name="가변성명" />
    <data android:scheme="http" /> <!-- URI 타입의 데이터 filtering -->
</intent-filter>
</activity>
<activity android:name="모바일서비스D" >
<intent-filter>
    ...
    <category android:name="가변성명" />
    <data android:scheme="..." /> <!-- 다른 형태의 데이터 타입 정의 -->
</intent-filter>
</activity>
```

그림 10. 데이터 타입에 따른 가변적 서비스 호출 설정

데이터 타입에 따른 가변적 서비스 호출을 위한 설정 코드는 [그림 10]과 같다. 3개의 모바일 서비스 중에 데이터 타입을 'http'로 정의한 '모바일서비스C'가 [그림 9]의 코드에 대해 자동으로 선택될 수 있는 서비스가 된다. 데이터를 정의하지 않은 '모바일서비스B'와 다른 형태의 데이터 타입을 정의한 '모바일서비스D'는 동일한 서비스 가변성을 정의하고 있지만 데이터 가변성에 의해 호출되지 않는다.

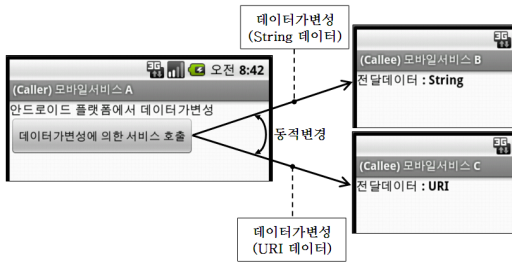


그림 11. 데이터 가변성에 의한 자가적용 사례

[그림 11]은 [그림 9]와 [그림 10]의 데이터 가변성을 위한 코드와 설정정보를 통해 구현된 자가적용 커스터마이제이션 기법의 사례이다. 전달되는 데이터의 타입이 문자열인 경우 ‘모바일서비스B’를 호출하며, 전달되는 데이터 타입이 URI인 경우 ‘모바일서비스C’가 호출될 수 있다. 이와 같이 자가적용 커스터마이제이션 기법은 서비스 실행 중에 선택 과정 없이 자동적으로 변경된다.

IV. 사례 연구

본 논문에서 제안한 데이터 가변성 기반의 자가적용 커스터마이제이션 기법을 상품주문 확인 사례에 적용하여 본 연구의 적합성을 검증한다.

시나리오는 상품주문 확인 발송 시 데이터의 유무와 데이터의 타입에 따라 이메일 서비스나 문자 서비스로 자동적으로 변경되어 공지 될 수 있도록 한다.

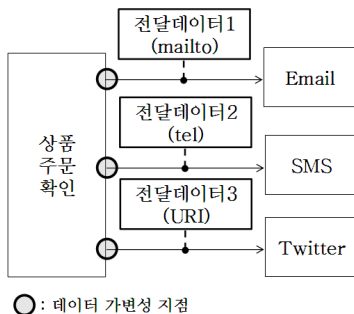


그림 12. 데이터가변성에 의한 상품주문 확인발송 자동 변경 구조

[그림 12]에서와 같이 상품주문 확인을 통해 전달되는 데이터에 따라 Email 서비스나 SMS 서비스, 그리고 Twitter 서비스로 변경될 수 있다.

전달되는 데이터는 ‘상품명’, ‘상품가격’, ‘상품명세’, ‘주문고객’으로 정의한다. 이 데이터 중에 ‘주문고객’의 타입에 따라 가변적으로 서비스 호출할 수 있도록 다양한 데이터 타입을 정의한다.

```

<activity android:name="OrderConfirm_Email" >
  <intent-filter>
    ...
    <category android:name="ORDER_CONFIRM" />
    <data android:scheme="mailto" />
  </intent-filter>
</activity>
<activity android:name="OrderConfirm_SMS" >
  <intent-filter>
    ...
    <category android:name="ORDER_CONFIRM" />
    <data android:scheme="tel" />
  </intent-filter>
</activity>
<activity android:name="OrderConfirm_Twitter" >
  <intent-filter>
    ...
    <category android:name="ORDER_CONFIRM" />
    <data android:scheme="twitter"
          android:host="twitlecture.com" />
  </intent-filter>
</activity>
    
```

그림 13. 상품주문 확인을 가변적으로 처리하기 위한 설정

전달되는 ‘주문고객’의 데이터 타입에 따라 서비스가 변경될 수 있도록 [그림 13]에서와 같이 정의한다. 데이터 타입이 ‘mailto’일 경우 메일 서비스를 호출하며, 데이터 타입이 ‘tel’이나 ‘twitter’ 인 경우 문자 서비스나 트위터 서비스로 변경될 수 있도록 설정을 정의한다.



그림 14. 데이터 가변성에 의한 상품주문 확인 메일발송

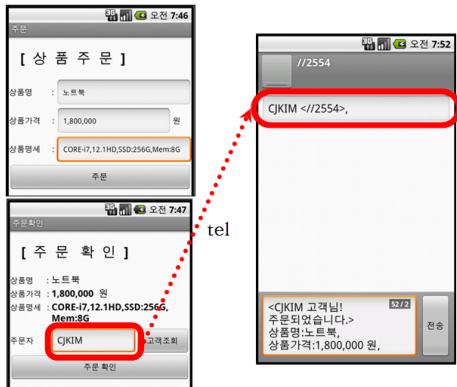


그림 15. 데이터가변성에 의한 상품주문 확인 문자발송

[그림 14]와 [그림 15]는 데이터 가변성에 의해 제공되는 결과 화면으로, [그림 14]는 데이터 타입이 'mailto' 타입으로 전달될 경우 메일 서비스로 변경된다. [그림 15]는 데이터 타입이 'tel' 타입으로 전달될 경우 문자 서비스로 자동 변경된다.

본 사례 적용을 통해 본 논문에서 제안한 데이터 가변성을 기반으로 하는 모바일 서비스의 자가적응 커스터마이제이션 기법이 타당함을 검증하였다.

V. 결론 및 향후 연구

지금까지 다양한 가변성 중에 데이터 가변성을 통해 안드로이드 플랫폼 기반의 모바일 서비스 가변성 기법을 제안하였다. 데이터 가변성 중에 데이터 유무와 데이터 타입에 따라 모바일 서비스 코드와 설정 코드가 어떻게 정의되는지를 구체적으로 제시하였으며, 사례에 적용하여 적합성을 검증하였다. 향후 모바일 데이터 가변성을 제공할 수 있는 추가 데이터 타입 및 분류를 분석하여 모바일 서비스를 커스터마이제이션할 수 있는 모바일 데이터 온톨로지(Ontology)에 대한 연구를 진행한다.

참고 문헌

[1] G. T. Heineman and W. T. Councill, *Component -*

Based Software Engineering, Addison-Wesley, 2001.

[2] B. König-Ries and F. Jena, "Challenges in Mobile Application Development," *Information Technology*, Vol.52, No.2, pp.69-71, 2009.

[3] C. J. Kim, E. S. Cho, and C. Y. Song, "A Design Technique of Configurable Framework for Home Network Systems," *Journal of the Korea Academia-Industrial Cooperation Society*, Vol.12, No.4, pp.1844-866, 2011(4).

[4] 김철진, 조은숙, "일반성 향상을 위한 가변성 설계 기법 및 커스터마이제이션 기법", *한국멀티미디어 학회 논문지*, 제9권, 제8호, pp.1076-1085, 2006.

[5] 안희수, 박만근, "객체지향 개발에서 재사용성 컴포넌트 설계 방법", *한국멀티미디어 학회 2001년도 춘계학술발표논문집*, pp.558-563, 2001.

[6] 임운선, 김명, "온톨로지 기반 데이터 가변성 처리 기법", *정보과학회논문지 : 소프트웨어 및 응용*, 제37권, 제4호, pp.239-251, 2010.

[7] 김철진, "안드로이드 기반의 웹 서비스 프로토콜 커스터마이제이션 기법", *한국콘텐츠학회논문지*, 제12권, 제6호, pp.92-99, 2012.

[8] 이락규, 피준일, 박준호, 복경수, 유재수, "모바일 환경에서 콘텐츠 추천 시스템 설계 및 구현", *한국콘텐츠학회논문지*, 제11권, 제12호, pp.40-51, 2011.

[9] 김만선, 주복규, "모바일 환경을 위한 지능형 추천 에이전트에 관한 연구", *한국콘텐츠학회논문지*, 제6권, 제4호, pp.55-62, 2006.

[10] C. Atkinson, J. Bayer, C. Bunse, E. Kamstices, O. Laitenberger, R. Laqua, D. Muthig, B. Paech, J. Wust, and J. Zettel, *Component-based Product Line Engineering with UML*, Addison-Wesley, 2001.

[11] J. Coplien, D. Hoffman, and D. Weiss, "Commonality and Variability in Software Engineering," *IEEE Software*, pp.37-45, November 1998.

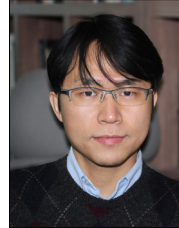
[12] K. C. Kang, S. G. Cohen, W. E. Novak, and A.

- S. Peterson, "Feature-oriented Domain Analysis (FODA) Feasibility Study," Technical Report CMU/SEI-90-TR-21, Software Engineering Institute (SEI), 1990(11).
- [13] J. Cheesman and J. Daniels, *UML Components : A Simple Process for Specifying Component-Based Software*, Addison Wesley, 2001.
- [14] D. F. D'souza and A. C. Wills, *Objects, Components, and Frameworks with UML*, Addison-Wesley, 1998.
- [15] A. Rausch, "Software Evolution in COMPONENTWARE Using Requirements/ Assurances Contracts," Proceedings of the 22th International Conference on Software Engineering, 2000(1).
- [16] C. Jose, "Next-Generation Object-Oriented Software Analysis and Design Methodology," at URL: http://www.hpl.hp.com/fusion/ma_961007.html, 1996.
- [17] Android Developers, <http://developer.android.com/index.html>.
- [18] J. Coplien, D. Hoffman and D. Weiss, "Commonality and Variability in Software Engineering," IEEE Software, pp.37-45, 1998(11).
- [19] C. J. Kim and S. D. Kim, "A Component Workflow Customization Technique," Korea Information Science Society, Vol.27, No.5, 2000.
- [20] C. J. Kim, S. H. Lee and E. S. Cho, "A Framework for Improving Reusability of Home Network System," Vol.1, No.2, ITIRC, 2008(9).
- [21] Rational Software Corp., *Unified Modeling Language(UML) Summary*, 2005.

저 자 소 개

김 철 진(Chul-Jin Kim)

정회원



- 2004년 : 숭실대학교 컴퓨터정보공학부(공학박사)
- 2004년 ~ 2009년 : 삼성전자 책임연구원
- 2009년 ~ 현재 : 인하공전 교수

<관심분야> : 클라우드 컴퓨팅, 웹 서비스, 모바일 서비스, 제품공학, 커스터마이제이션