

RESTful 기반의 모바일 서비스 매쉬업 커스터마이제이션 기법

김철진^{1*}

¹인하공업전문대학 컴퓨터시스템과

A Mashup Customization Technique of Mobile Service based on RESTful

Chul-Jin Kim^{1*}

¹Dept. of Computer Systems and Engineering, Inha Technical College

요 약 기업 업무의 스마트워크 요구로 모바일 서비스의 사용이 증가하고 있으며, 기업 업무의 다양한 요구사항을 대응할 수 있는 모바일 서비스 구조에 대한 연구가 요구된다. 모바일 서비스의 다양성을 제공하기 위해 커스터마이제이션에 대한 연구들이 모바일 플랫폼에 적용되도록 연구되어야 한다. 모바일 서비스가 다른 서비스들과의 연동 및 디바이스 연속성을 제공하기 위해 공개 인터페이스(Open API)를 이용하여 서비스들 간의 통합을 가능하게 해야 한다. 본 논문에서 이러한 서비스들 간의 통합을 통해 다양한 요구사항을 만족시킬 수 있는 매쉬업 커스터마이제이션 기법을 제안한다.

Abstract Business needs of companies to require smart work have been increased use of mobile services and the research for the mobile service architecture to respond to the diverse needs of business operations is required. For providing variety of mobile services, customization research for the mobile platform should be performed. Mobile service to provide integration services and device continuity should be able to integrated(Mashup) among services using open interface(Open API).

This paper provides the mashup customization technique that can meet a variety of requirements by integration among such services.

Key Words : Mashup, Customization, Mobile Service, RESTful

1. 서론

현재 기업 업무의 스마트워크 요구로 모바일 서비스의 사용이 증가하고 있으나, 또한 기업 업무의 다양한 요구사항을 대응할 수 있는 모바일 서비스 구조에 대한 연구가 미흡한 상황이다. 기존의 CBD(Component Based Development)분야에서는 다양성을 제공하기 위해 커스터마이제이션에 대한 연구들이 진행되었으며, 이러한 재사용성에 대한 연구들이 모바일 플랫폼에 적합하도록 적용되어야 할 것이다. 또한 모바일 서비스가 기업 내부의 레거시 시스템(Legacy System)과 연동을 통해 연속적인 서비스를 제공해야 한다. 현재 많은 기업들이 기업 내부

의 시스템 연동 및 디바이스 연속성을 제공하기 위해 공개 인터페이스(Open API)를 이용하여 서비스들 간의 통합(매쉬업(Mashup))을 가능하게 하고 있다. 본 논문에서 이러한 서비스들 간의 통합을 통해 다양한 요구사항을 제공할 수 있는 매쉬업 커스터마이제이션 기법을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구로서 커스터마이제이션 기법, 가변성, RESTful 서비스, 그리고 안드로이드 프레임워크에 대해 알아본다. 3장에서는 본 논문의 핵심연구로서 매쉬업 커스터마이제이션 기법을 제안한다. 4장에서는 बैं킹 웹 서비스와 지도 오픈 서비스를 매쉬업하여 본 연구의 타당성을 검증한다.

*Corresponding Author : Chul-Jin Kim

Tel: +82-10-6398-7471 email: cjkim@inhac.ac.kr

접수일 12년 06월 12일

수정일 12년 06월 25일

게재확정일 12년 07월 12일

2. 관련 연구

2.1 커스터마이제이션 기법

Catalysis[1]는 상호 연동할 수 있는 컴포넌트들을 가지고 어플리케이션을 개발하는 데 초점을 두고 있는데, 분석에서부터 구현까지 이르는 컴포넌트 모델링 기법들을 정의하며 컴포넌트들을 명세화하는 기법 등을 제시하고 있다. 그러나 각각의 단계에 정의된 절차들을 보면 다양한 도메인에 적용될 수 있도록 가변성을 추출하여 컴포넌트를 설계하는 세부 지침을 정의하고 있지 않다.

Componentware에서는 컴포넌트 개발의 여러 단계들을 조합하여 다양한 형태의 프로세스를 구성할 수 있도록 프로세스 패턴들을 제공한다[2]. 컴포넌트는 재사용성이 높고 동적 바인딩이 용이하도록 설계 기법이나 가변성 설계 지침이 구체적으로 제공되어야 하는데, 이 방법론에서는 단지 인터페이스 정의 수준에 제공하고 있다.

CoPAM(Component-Oriented Platform Architecting Method)[3]은 제품 공학(Product Engineering)을 위한 개발 프로세스로서, 플랫폼 공학(Platform Engineering)과 제품 공학 2개의 서브 프로세스로 구성되어 있다. 플랫폼 공학에서는 여러 개의 재사용 컴포넌트들로 구성된 플랫폼을 개발하기 위한 프로세스를 정의하고 있으며, 제품 공학 프로세스에서는 이러한 플랫폼을 이용한 제품을 개발하기 위한 업무들을 정의하고 있다. 이 방법론은 플랫폼 엔지니어링 프로세스에서 여러 도메인에 공통된 부분과 가변적인 부분들을 컴포넌트로 추출하기 위한 활동에 초점을 두고 있다. 따라서, 컴포넌트 내부 설계 기법이나 가변성 설계와 관련된 구체적인 지침은 제시하고 있지 않다.

2.2 가변성

가변성(Variability)은 동일 기능을 제공하는 제품군(Product Family)들 간의 차이를 말한다[4,5]. 컴포넌트에서 가변성은 기능(함수)의 차이를 의미한다. 동일 컴포넌트군(Component Family)들 간에 동일 기능을 제공하는 함수가 서로 다른 알고리즘으로 기능을 제공하는 경우를 컴포넌트의 기능 가변성[6]이라고 한다. 예를 들면, A 은행의 बैं킹 컴포넌트와 B 은행의 बैं킹 컴포넌트에서 적금이자 계산 기능이 A 은행의 컴포넌트에서는 복리계산을 통해 이자를 계산하고, B 은행의 컴포넌트에서는 단리계산을 통해 이자를 계산할 경우, 두 컴포넌트를 통해 बैं킹 컴포넌트 군은 이자 계산 기능을 가변성으로 정의할 수 있다.

2.3 RESTful 서비스

RESTful(Representational State Transfer)[7] 서비스는 URI(Uniform Resource Identifier)를 통해 웹 서비스를 제공하기 위한 방식이다. 클라이언트와 서버의 형식에 상관없이 HTTP 프로토콜만으로 접근이 가능하다. 기본적으로 HTTP 프로토콜을 사용하여 전달되는 자원의 상태를 관리하지 않으나, 기존 웹 서비스인 SOAP(Simple Object Access Protocol)[8] 방식보다 사용이 간단하고 개방성이 뛰어나므로 사용성이 확대되고 있다. 이러한 개방성을 통해 서비스들 간의 통합을 제공하는 매쉬업(Mashup)[9] 서비스가 가능하게 할 수 있다. RESTful 서비스에서 제공되는 기능은 GET, POST, PUT, DELETE, HEAD, OPTION 이 있으며, 이러한 기능을 통해 자원을 관리할 수 있다.

2.4 안드로이드 어플리케이션 프레임워크

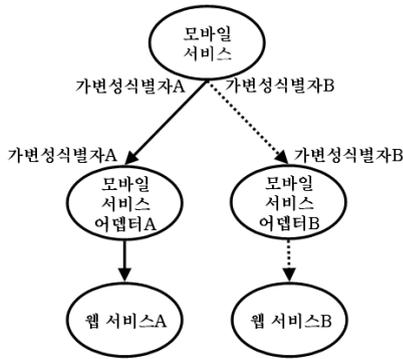
안드로이드 어플리케이션 프레임워크는 본 논문에서 제안한 모바일 서비스 커스터마이제이션을 구현하기 위한 기본 메커니즘을 제공할 수 있다. 서비스를 가변적으로 변경하기 위한 방법은 설정 정보를 통해서 가능할 수 있는데, 안드로이드 어플리케이션 프레임워크는 설정정보(AndroidManifest.xml) 내의 인텐트 필터(Intent Filter)에 가변적으로 서비스를 변경할 수 있도록 식별자를 정의한다. 안드로이드 어플리케이션 프레임워크 내의 패키지 관리자(Package Manager)는 인텐트 필터에 정의된 서비스를 필터링하여 서비스를 변경할 수 있다[10].

3. 모바일 매쉬업 커스터마이제이션 기법

모바일 매쉬업 커스터마이제이션 기법은 안드로이드 어플리케이션 프레임워크 기반의 구조를 이용하여 매쉬업 커스터마이제이션을 제안한다.

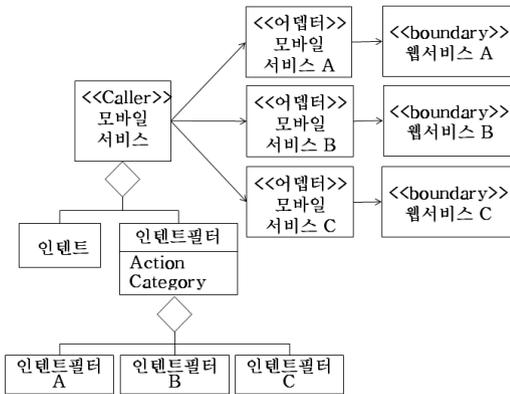
3.1 모바일 커스터마이제이션

모바일 커스터마이제이션은 가변성을 제공하는 설정 정보를 통해 커스터마이제이션이 가능하다. 그림 1에서와 같이 가변성 식별자에 따라 '모바일 서비스 어댑터 A' 나 '모바일 서비스 어댑터 B' 로 변경될 수 있다. 이때 모바일 서비스 어댑터는 웹 서비스를 호출하기 위한 어댑터 역할을 수행한다.



[그림 1] 모바일 커스터마이제이션 구조
[Fig. 1] Mobile Customization Architecture

그림 1의 모바일 커스터마이제이션 구조에 대한 클래스 다이어그램은 그림 2와 같다. 안드로이드 기반의 커스터마이제이션 구조는 인텐트 필터라는 설정정보를 통해 모바일 서비스를 가변적으로 변경할 수 있다. 인텐트 필터에 따라 모바일 서비스 어댑터를 변경할 수 있으며 이에 따라 웹서비스를 변경할 수 있다.



[그림 2] 모바일 커스터마이제이션 구조에 대한 클래스 다이어그램
[Fig. 2] Class Diagram of Mobile Customization Architecture

모바일 서비스를 가변적으로 변경하기 위한 안드로이드의 인텐트 필터에 대한 설정 코드는 그림 3에서와 같이 정의할 수 있다. ‘<intent-filter>’ 태그 내의 ‘<category>’ 태그에 가변성 식별자를 명시하여 변경하고자하는 서비스를 지정하게 할 수 있다. 해당 가변성 식별자에 대한 정의는 호출되는 모바일 서비스에서도 동일한 식별자명으로 지정해야 한다.

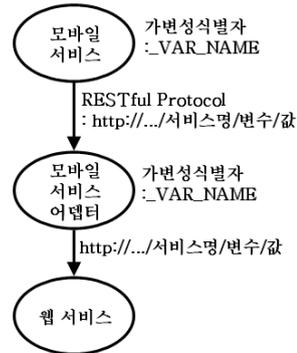
```

<activity android:name="모바일서비스A" >
  <intent-filter>
    <action android:name="android.intent.action.MAIN" />
    <category android:name="android.intent.category.DEFAULT" />
    <category android:name="모바일서비스A_가변성식별자" />
  </intent-filter>
</activity>
<activity android:name="모바일서비스B" >
  <intent-filter>
    ...
    <category android:name="모바일서비스B_가변성식별자" />
  </intent-filter>
</activity>
<activity android:name="모바일서비스C" >
  <intent-filter>
    ...
    <category android:name="모바일서비스C_가변성식별자" />
  </intent-filter>
</activity>
  
```

[그림 3] 모바일 커스터마이제이션 설정 코드
[Fig. 3] Configuration Code of Mobile Customization

3.2 매쉬업 커스터마이제이션

RESTful 기반의 모바일 서비스는 그림 4에서와 같이 웹 서비스의 정보를 모바일 서비스 어댑터를 통해 호출할 수 있다. 웹 서비스의 호출은 RESTful 형태로 호출된다.



[그림 4] RESTful 기반 모바일 서비스 기본 구조
[Fig. 4] Mobile Service Basic Structure based on RESTful

모바일 서비스 어댑터를 통해 RESTful 기반으로 웹 서비스를 호출하는 코드는 그림 5와 같다. 호출되는 RESTful 서비스는 URL에 해당하는 IP와 포트(Port), 서비스명, 그리고 전달되는 데이터 변수와 데이터 값으로 구성된다. 이와 같이 모바일 서비스를 통해 RESTful 방식으로 웹 서비스를 호출하는 방식은 GET 방식이나 POST 방식으로 호출할 수 있다.

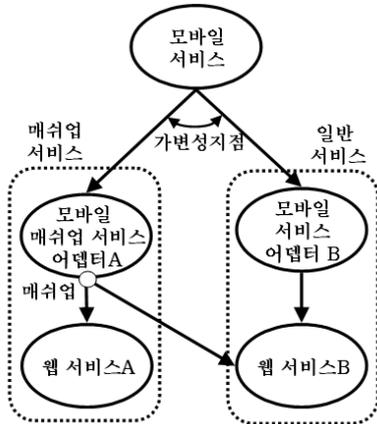
```

HttpGet mGetMethod =
new HttpGet("http://192.168.0.12:8080"
+ "/서비스명"
+ "/변수1/값1"
+ "/변수2/값2");

HttpResponse mHttpRes =
httpClient.execute(mGetMethod);
    
```

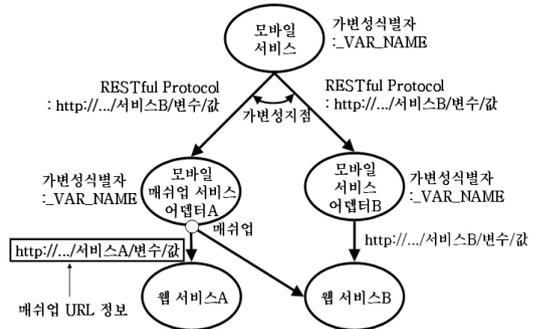
[그림 5] RESTful 기반 모바일 서비스 코드
 [Fig. 5] Mobile Service Code based on RESTful

그림 4와 그림 5의 모바일 웹 서비스 구조와 코드를 기반으로 그림 6과 같이 매쉬업 커스터마이제이션을 제공할 수 있다. 모바일 서비스는 모바일 서비스 어댑터를 통해 일반 웹 서비스 뿐만 아니라, 웹 서비스들 간에 매쉬업을 제공하는 통합된 웹 서비스를 호출 할 수 있다.



[그림 6] 매쉬업 커스터마이제이션 구조(1)
 [Fig. 6] Mashup Customization Structure(1)

매쉬업 커스터마이제이션을 위해 전달되는 데이터 형태는 그림 7과 같다. 모바일 서비스로부터 가변성 지점의 가변성 설정 정보를 통해 모바일 매쉬업 서비스 어댑터 A나 모바일 서비스 어댑터 B로 변경될 수 있다. 모바일 서비스 어댑터 B는 일반 웹 서비스를 호출하지만, 모바일 매쉬업 서비스 어댑터 A는 일반 웹서비스와 다른 웹 서비스를 매쉬업하여 서비스를 제공한다. 이때 전달되는 서비스 호출 정보는 그림 5에서와 같이 URL, 서비스명, 전달 데이터로 구성된다. 이때 모바일 매쉬업 서비스 어댑터A는 다른 웹 서비스('http://.../서비스A/...')를 매쉬업하여 서비스를 제공한다.



[그림 7] 매쉬업 커스터마이제이션 구조(2)
 [Fig. 7] Mashup Customization Structure(2)

매쉬업하기 위해 RESTful 형태로 전달되는 데이터는 그림 8에서와 같이 해당 웹 서비스 API와 전달 데이터로 구성된다. 모바일 매쉬업 어댑터에서는 2개 이상의 웹 서비스를 매쉬업하기 그림 8에서 정의한 웹 서비스A의 전달 데이터 '값A1'을 웹 서비스B의 '변수B1'으로 전달하여 매쉬업을 가능하게 할 수 있다.

웹 서비스 A의 RESTful

항목	전달 데이터
서비스API	http://.../서비스A
전달데이터 {1..n}	변수A1/값A1/ ... /변수An/값An

웹 서비스 B의 RESTful

항목	전달 데이터
서비스API	http://.../서비스B
전달데이터 {1..n}	변수B1/값A1/ ... /변수Bn/값Bn

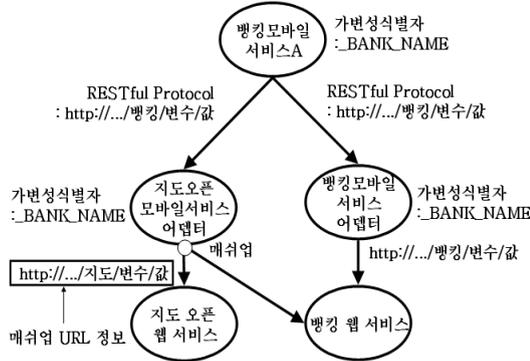
[그림 8] RESTful 기반의 매쉬업 전달 데이터
 [Fig. 8] Mashup Sending Data based on RESTful

지금까지 모바일 기반의 커스터마이제이션 구조에 모바일 서비스 어댑터의 매쉬업 기능을 정의하여 커스터마이제이션할 수 있는 기법을 제안하였다.

4. 실험 및 평가

본 논문에서 제안한 모바일 매쉬업 커스터마이제이션 기법을 온라인 뱅킹(Online Banking) 사례에 적용한다. 그림 9에서와 같이 뱅킹 모바일 서비스 어댑터를 이용하여 뱅킹 웹 서비스를 호출하는 일반 서비스와 지도 오픈 모바일 서비스 어댑터를 호출하여 지도 오픈 웹 서비스

와 बैं킹 웹 서비스를 매쉬업하여 서비스를 제공할 수 있다. 2가지 서비스를 가변적으로 커스터마이제이션 시킬 수 있다.



[그림 9] 온라인 बैं킹 매쉬업 커스터마이제이션 구조
[Fig. 9] Mashup Customization Structure of Online Banking

```

i.setAction(Intent.ACTION_MAIN);
i.addCategory("_VARIABILITY_NAME");
i.addCategory(Intent.CATEGORY_ALTERNATIVE);

Intent intent = getIntent();

userId = intent.getStringExtra("userId");
money = intent.getStringExtra("money");
serviceAPI = intent.getStringExtra("serviceAPI");

HttpGet mGetMethod =
    new HttpGet(serviceAPI+"/userId/"+userId+"/"
                +"/money/"+money+"/");
HttpResponse mHttpRes = httpClient.execute(mGetMethod);

Header[] headers = mHttpRes.getHeaders("balance");
balance = headers[0].getValue().toString();
    
```

[그림 10] 온라인 बैं킹 매쉬업 코드
[Fig. 10] Mashup Code of Online Banking

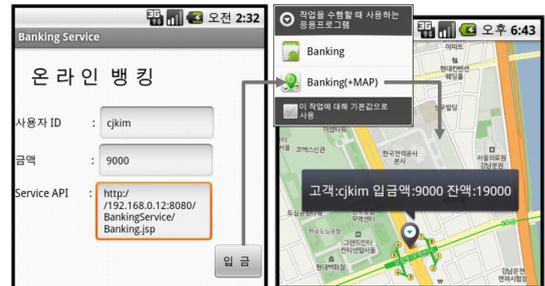
모바일 서비스 어댑터에서 बैं킹 웹 서비스를 매쉬업 하는 코드를 그림 10에서와 같이 정의한다. 'serviceAPI' 를 통해 전달된 웹 서비스 API는 지도 서비스와 매쉬업 할 수 있다. 이때 전달된 사용자정보('userId')와 입금액 ('money')은 지도 서비스의 데이터로 전달되어서 매쉬업을 가능하게 한다.

그림 11은 온라인 बैं킹 서비스 어댑터를 통해 बैं킹 웹 서비스를 연동하는 결과화면이다. 이 경우는 매쉬업 없이 기본 서비스를 제공받을 수 있다. 사용자는 공개된 서비스 API를 활용하여 बैं킹 웹 서비스를 사용한다.



[그림 11] 온라인 बैं킹 기본 서비스 결과
[Fig. 11] Basic Service's Result of Online Banking

그림 12는 지도 모바일 서비스 어댑터를 이용하여 오픈 지도 서비스[11]와 बैं킹 웹 서비스를 매쉬업하여 서비스를 커스터마이제이션 시킬 수 있음을 나타내는 결과화면이다.



[그림 12] 온라인 बैं킹 매쉬업 커스터마이제이션 결과
[Fig. 12] Mashup Customization's Result of Online Banking

온라인 बैं킹 서비스에서 지도 모바일 서비스 어댑터로 전달되는 RESTful 정보는 서비스명과 전달 데이터(사용자 ID, 금액)이다. 전달된 데이터를 기반으로 बैं킹 웹 서비스와 지도 서비스를 매쉬업 한다. 지도 위에 사용자 ID, 입금액, 그리고 बैं킹 웹 서비스로부터 얻은 잔액을 표시한다.

본 사례연구를 통해 본 논문에서 제시한 모바일 서비스 매쉬업 커스터마이제이션 기법이 타당함을 검증하였다.

5. 결론 및 향후 연구과제

본 논문은 모바일 서비스와 웹 서비스들 간의 다양한 요구사항을 만족시킬 수 있는 매쉬업 커스터마이제이션 기법을 제안하였다. RESTful 방식의 웹 서비스 호출을 모바일 서비스 어댑터를 통해 가변적으로 호출하거나 매쉬업할 수 있는 커스터마이제이션 기법을 제안하였다. 또한 बैं킹 웹 서비스와 오픈 지도 서비스를 통합하여 매쉬

업 커스터마이제이션이 적용될 수 있음을 사례연구를 통해 증명하였다. 향후에는 클라우드 서비스 환경에서 개발 서비스를 RESTful 기반으로 공개 인터페이스를 제공하여 개발자에게 PaaS(Platform as a Service) 기반 커스터마이제이션 기법을 제공할 수 있는 연구를 진행한다.

References

- [1] D'souza D. F. and Wills A. C., Objects, Components, and Components with UML, Addison-Wesley, 1998.
- [2] Rausch A. "Software Evolution in COMPONENTWARE Using Requirements/Assurances Contracts", Proceedings of the 22th International Conference on Software Engineering, 06/2000.
- [3] P. America, H. Obbink, R. V. Ommering, F. V. D. Linden, "CoPAM: A Component-Oriented Platform Architecting Method Family for Product Family Engineering", The First Software Product Line Conference(SPLC), Kluwer International Series in Software Engineering and Computer Science, Denver, Colorado, USA, p.15, 2000.
- [4] Heineman, G. T. and Councill, W. T., "Component - Based Software Engineering", Addison-Wesley, 2001.
- [5] Atkinson, C., Bayer, J., Bunse, C., Kamsties, E., Laitenberger, O., Laqua, R., Muthig, D., Paech, B., Wust, J., Zettel, J., Component-based Product Line Engineering with UML, Pearson Education Ltd, 2002.
- [6] Kim C. J. and Kim S. D., "A Component Workflow Customization Technique", Vol.27, No.5, Korea Information Science Society, 2000.
- [7] SOAP Version 1.2 Recommendation documents, www.w3.org/TR/soap/, World Wide Web Consortium (W3C), 2007.
- [8] RESTful Web services: The basics, <https://www.ibm.com/developerworks/webservices/library/ws-restful/>, IBM, 2008.
- [9] D. Merrill, Mashups: The new breed of Web app: An introduction to mashups, <http://www.ibm.com/developerworks/library/x-mashups.html>.
- [10] <http://developer.android.com/>
- [11] NAVER OpenAPI, <http://dev.naver.com/openapi/>

김철진(Chul-Jin Kim)

[종신회원]



- 2004년 2월 : 숭실대학교 대학원 컴퓨터학과 (공학박사)
- 2004년 4월 : 가톨릭대학교 컴퓨터 정보공학부 강의전담교수
- 2004년 3월 ~ 2009년 2월 :삼성전자 책임연구원
- 2009년 3월 ~ 현재 : 인하공전 컴퓨터시스템과 조교수

<관심분야>

컴포넌트 기반 개발 방법론, 컴포넌트 커스터마이제이션, 모바일 서비스, 클라우드 컴퓨팅