

모바일 클라우드 서비스 기반 다중 커스터마이제이션 기법

예준호¹, 김철진^{1*}

¹인하공업전문대학 컴퓨터시스템과

A Multiple Customization Technique based on Mobile Cloud Service

Jun-Ho Ye¹ and Chul-Jin Kim^{1*}

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

요 약 기존 모바일 커스터마이제이션 연구는 단일 디바이스의 서비스를 변경하려는 단일 커스터마이제이션 기법으로 수행되어왔다. 그러나, 변경하려는 모바일 서비스가 다중 디바이스에 일괄적으로 적용되기 위해서는 기존의 단일 커스터마이제이션 기법으로는 미흡하다. 본 논문에서는 모바일 서비스의 흐름제어 데이터를 공유하여 다중 디바이스를 커스터마이제이션하기 위한 기법을 제안 한다. 제안하는 다중 커스터마이제이션 기법은 푸시 서비스와 클라우드 서비스를 기반으로 한다.

Abstract The existing mobile customization researches have been performed by the single customization technique to change service of single device. But, the existing single customization techniques are insufficient to change mobile service to multiple devices in batches. In this paper, we propose the customization technique for customizing multiple devices sharing flow control data of mobile service. A multiple customization technique is based on the push service and the cloud service.

Key Words : Cloud Service, Multiple Customization, Push Service

1. 서론

모바일 어플리케이션에서 적용되어 사용되는 서비스는 일반적으로 고정 된 흐름을 지닌다. 기기에 설치된 운영체제에 어플리케이션이 종속되어 있기 때문이다. 서비스의 변경은 이러한 종속성에서 결코 자유롭지 못하다. 그렇기에 서비스나 데이터의 변경을 위해서는 필연적으로 재설치를 통한 업데이트의 형태로 이루어지게 된다. 서비스 종속의 대안으로 서비스를 단편화 시키고, 관리자가 원하는 서비스로 변경하는 기법이 바로 단일 커스터마이제이션이다. 단일 커스터마이제이션의 서비스의 독립성을 통해 서비스 흐름을 제어할 수 있는 장점에도 불구하고, 다중 디바이스에 서비스를 주입하기에는 서비스

를 수동으로 변경하는 단점으로 인해 개개의 디바이스를 조작하는 시간허비와, 서비스가 변경 된 이후에도 디바이스의 서비스가 비동기화 되었을 위험성, 그리고 의도하지 않은 서비스로 동기화 될 위험성이 있다. 이런 위험성을 줄이고자 클라우드에 서비스를 제어할 수 있는 데이터를 저장하고, 다중 디바이스가 클라우드와 서비스 동기화하거나, 일괄적으로 서비스를 배포하는 것으로 단일 커스터마이제이션에서 발생하는 위험성을 해소할 수 있다. 단일 커스터마이제이션의 위험성 해소를 위해 본 논문에서는 단일 커스터마이제이션 기법과 클라우드 서비스를 결합한 다중 커스터마이제이션 기법을 제안한다.

본 논문은 다음과 같이 구성된다. 2장에서는 관련 연구로서 모바일 클라우드 서비스와 푸시 서비스에 대해

*Corresponding Author : Chul-Jin Kim(Inha Technical College)

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

Received August 14, 2013

Revised September 5, 2013

Accepted September 6, 2013

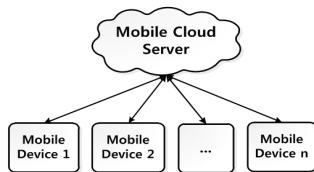
알아보고, 기존 커스터마이제이션 연구를 분석하며, 3장에서는 두 가지 다중 커스터마이제이션 연구를 제안한다. 4장에서는 팀원 간의 일정 공유를 위한 어플리케이션 사례를 통해 본 논문에서 제안한 기법의 타당성을 검증하고, 5장에서 결론과 향후 연구의 방향성을 제시한다.

2. 관련 연구

2.1 모바일 클라우드 서비스

모바일 클라우드(Mobile Cloud)란 기존 클라우드 환경에서 디바이스가 데이터나 서비스를 얻어 올 시, OS 혹은 디바이스 내부의 데이터베이스처럼 고정된 환경에 영향을 받는 것과 달리, 유동적인 상황과 다른 OS환경에서도 데이터를 쉽게 얻어오는 기술을 의미한다. 기존 클라우드와 모바일 클라우드의 차이점은 모바일 클라우드가 서버에 대한 접근이 간단하고, 서비스의 높은 확장성을 가지며, 자유롭게 디바이스를 이동할 수 있다는 것이다 [1].

모바일 클라우드 서비스를 사용함은 디바이스 내의 데이터베이스를 이용하는 것이 아닌, 모바일 클라우드와의 연결을 통해서 데이터나 서비스를 받거나 저장하는 것을 의미한다[2].



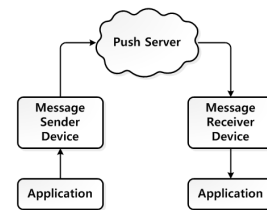
[Fig. 1] Mobile Cloud Service Structure

즉, Fig. 1과 같이 모바일 클라우드가 디바이스에게 직접 서비스나 데이터를 제공한다. 모바일 클라우드는 그에 대한 관심과 활발한 연구로 많은 오픈 API가 존재한다. 그렇기에 모바일 클라우드의 오픈 API를 사용하는 것으로 구축비용과 유지 보수비용을 줄일 수 있으며, 라이선스 수수료에 대한 부분에서도 비용이 절감 된다는 장점이 있다[3]. 이외에도 HTML5의 확산과 하이브리드 웹 앱(Hybrid Web App)의 등장으로 디바이스 OS 의존적인 구축성에서 벗어나, 활용 영역이 넓어졌다는 점을 의미한다[4].

2.2 푸시 서비스

푸시 서비스(Push Service)란 서버에서 디바이스로 보

내는 메시지로서 푸시 서비스를 받는 클라이언트의 입장에서 클라이언트의 의지와 상관없이 메시지를 받아들이게 된다. 그러한 점에서 푸시 서비스는 클라이언트가 서버로 요청이나 전달을 목적으로 보내는 메시지 서비스인 풀 서비스(Pull Service)와는 정반대되는 개념이다. 푸시 메시지를 활용하기 위해서는 디바이스 정보와 디바이스에 설치된 어플리케이션의 정보를 취합하여 만든 푸시 키 데이터를 서버에서 관리되어야 한다. 또한 푸시 서비스 메시지를 받기 위해서는 클라이언트의 디바이스는 반드시 켜져 있어야하고, 디바이스에 받을 어플리케이션이 설치되어야만 서버로부터 푸시를 받을 수 있다.

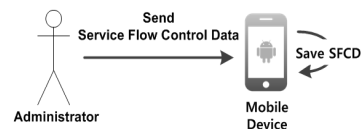


[Fig. 2] Push Service Structure

Fig. 2는 현재 상용화 되어 사용되는 푸시서비스 구조이다. 푸시서비스의 사례로는 SMS(Short Message Service)와 메일(SMTP) 서비스가 대표적인 푸시서비스의 예다. 애플에서는 APNs(Apple Push Notification service)를 제공하고 있으며, 구글에서는 GCM(Google Cloud Messaging)을 통해 푸시 서비스를 제공하고 있다. 두 서비스는 각각 이름과 명칭은 다르지만 Fig. 2를 통해 볼 수 있는 푸시 서비스의 구조와 유사한 형태의 서비스 제공 방식을 취하고 있다[5].

2.3 단일 커스터마이제이션 구조

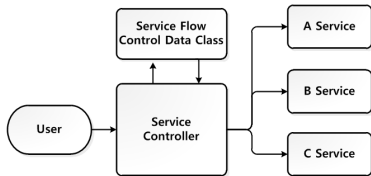
단일 커스터마이제이션이란 변경하려는 서비스 흐름을 단일 디바이스에 적용하는 것을 의미한다.



[Fig. 3] Service Flow of Single Customization

관리자가 서비스를 변경하기 위해서는 서비스에 대한 설정 값이 들어있는 서비스 흐름 제어 데이터(Service Flow Control Data)를 사용자의 디바이스로 전달함으로써

써 사용자의 서비스를 제어한다. 여기서 서비스란 디바이스를 통해 사용자가 보게 되는 화면-액티비티(Activity)를 의미한다. 관리자가 서비스 제어를 위해 보낸 서비스 흐름제어 데이터는 Fig. 3과 같은 과정을 거쳐 사용자의 모바일 디바이스로 전송 후 저장 된다.



[Fig. 4] Service Flow Structure of Single Customization

단일 커스터마이제이션의 서비스 구조는 Fig. 4와 같이 사용자가 서비스를 사용하려 할 때, 모바일 디바이스에 서비스를 요청하고, 제공받는 역할을 하는 클래스인 ServiceController에 서비스를 요청한다. ServiceController는 관리자가 전달한 서비스 흐름제어 데이터를 분석하여, 관련 서비스로 변경한 뒤, 관리자가 의도한 서비스로 디바이스의 서비스를 전환한다.

2.4 모바일 클라우드 기반 다중 커스터마이제이션 연구

2.4.1 iCloud와 Google Cloud Storage

애플사에서 제공하는 클라우드 컴퓨팅 서비스(Cloud Computing Service)로서 모바일 디바이스와 개인 컴퓨터, 태블릿 등을 연결하는 매개체 역할을 한다[6]. 개인의 데이터 관리가 주 목적인 클라우드 서비스이기에 커스터마이제이션을 위한 데이터 공유나 서비스 제어에 사용키에는 부적절하다.

구글에서 제공하는 구글 클라우드 스토리지는 RESTful형식의 서비스로서 저장과 데이터 접근을 위한 클라우드이다. iCloud와 마찬가지로 데이터베이스적인 측면이 강하고, 서비스에 대한 지원은 iCloud 보다 더 적다[7,8]. 유럽과 미국을 중심으로 서비스를 제공하기 때문에 접근이 불안정하고, Google Console로 API를 제어할 수 있지만, 원하는 대로 구조를 변경할 수가 없어 서비스 제어를 통한 커스터마이제이션에는 적합하지 못하다. 따라서 상용화되거나 기업에서 제공하는 클라우드에 대한 서비스를 사용하기보단 직접 모바일 클라우드 환경을 만드는 것이 더욱 효과적일 것이다.

2.4.2 푸시서비스 기반 T-DMBS사례

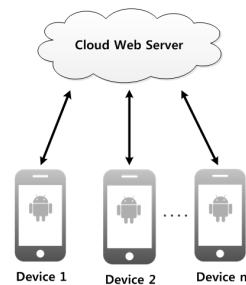
현재는 GCM으로 업데이트 된 구글의 C2DM 서비스

를 활용한 본 사례[9]는 재난 발령서버에서 재난이 예상되거나 발생한 지역에서 지상파 디지털 멀티미디어 방송(T-DMB: Terrestrial Digital Multimedia Broadcasting)을 통해 재난방송을 할 수 있도록, T-DMB를 탑재된 안드로이드 모바일 디바이스에 푸시 메시지를 보내어 자동인식을 통하여 재난을 조기에 알리는 시스템이다. 서비스가 디바이스에 전달되고, 어플리케이션의 서비스 변경을 유도한다는 점에서 커스터마이제이션과 유사함을 알 수 있다. 하지만, 사용자가 서비스 변경을 알아야 한다는 점과 다중의 디바이스에 변경한 서비스를 배포하는 과정을 보았을 때, 커스터마이제이션으로 보기엔 미흡한 연구이다.

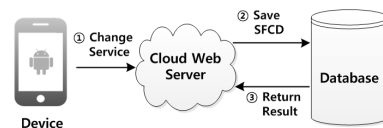
3. 클라우드 서비스 기반 다중 커스터마이제이션

3.1 다중 커스터마이제이션 구조

서비스변경의 모든 권한을 부여받은 관리자가 서비스 흐름제어 데이터를 직접 디바이스에 전달하여, 서비스 흐름을 변환하는 구조가 단일 커스터마이제이션이라면, 다중 커스터마이제이션은 Fig. 5의 구조처럼 서비스 흐름제어 데이터를 모바일 클라우드를 통해 다중 디바이스가 공유 받음으로써, 서비스의 대한 흐름을 일괄적으로 전환할 수 있는 구조이다.



[Fig. 5] Multiple Customization Structure



[Fig. 6] Service Flow of Customization by Cloud Server

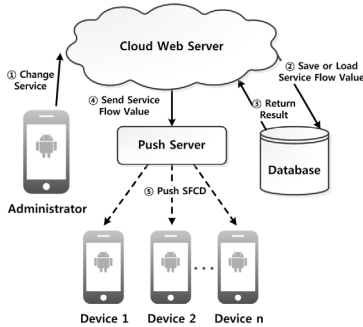
Fig. 6에서와 같이 사용자가 변경할 서비스 흐름제어 데이터(SFCD)를 모바일 클라우드에 전달하는 것으로 서비스 변경을 요청한다. 모바일 클라우드에서는 데이터베

이에 서비스 흐름제어 데이터를 저장한다. 이로써, 다중 디바이스의 서비스를 변경할 수 있는 준비가 끝나게 된다.

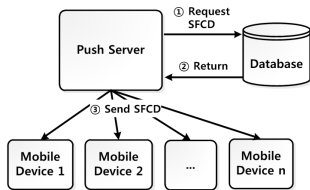
변경된 서비스 정보를 다중의 디바이스에게 전달하는 과정은 푸시서비스를 이용하는 방식과 모바일 클라우드 서비스를 이용하는 방식으로 구분된다.

3.2 푸시 서비스 기반 다중 커스터마이제이션 기법

푸시 서비스를 통한 모바일 클라우드 기반 다중 커스터마이제이션은 모바일 클라우드의 사용자가 변경한 서비스 흐름제어 데이터를 저장함으로써 다른 디바이스의 서비스 변경을 유도한다. 푸시 서버는 변경된 정보를 수신하고, 모바일 클라우드에 가입된 다중 디바이스에 서비스 흐름제어 데이터를 전송하는 기법이다. 사용자는 변경된 서비스에 대한 흐름을 인식하지 못하는 상황에서 디바이스 내부에 서비스 흐름제어 데이터를 저장한 후, 저장된 데이터를 바탕으로 서비스를 이용한다. 푸시서비스 기반 다중 커스터마이제이션 구조는 다음 Fig. 7과 같다.



[Fig. 7] Multiple Customization Structure by Push Service

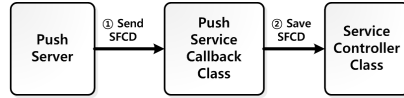


[Fig. 8] Sending Message by Push Server

푸시 서버 활용은 모바일 클라우드의 담당이다. 각 디바이스에 데이터 전송부분은 Fig. 8에서와 같이 푸시 서버에 위임하여 모바일 클라우드의 부담을 최소화한다.

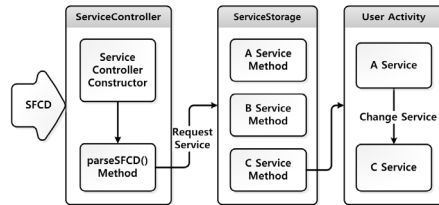
푸시서버는 모바일 클라우드가 보낸 서비스 요청을 바탕으로 데이터베이스에서 정보를 받아오고, 데이터베이스

스에서 받아온 서비스 흐름제어 데이터(SFCD)를 확인하여 다중 디바이스에 전송한다. 만일에 전송이 실패할 경우를 대비해 동일한 메시지를 재전송하게 된다.



[Fig. 9] Callback Process of Mobile Device

푸시 서버가 보낸 설정 정보는 Fig. 9의 과정과 같이 Push-ServiceCallback 클래스를 통해 전달 받는다. 수신한 설정 정보는 ServiceController 클래스를 생성한 뒤, 서비스 흐름제어 데이터를 전달한다.



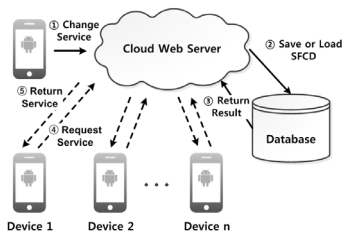
[Fig. 10] Analysis Process of SFCD

ServiceController 클래스는 Fig. 10처럼 생성자를 통해 서비스 흐름제어 데이터를 전달 받는다. 생성자는 parseSFCD() 메소드를 호출한다. parseSFCD() 메소드는 서비스 흐름제어 데이터를 분석하고, 최종으로 반환할 서비스를 판단한다. 분석한 데이터를 바탕으로 ServiceStorage 클래스의 메소드를 호출하며 사용자 화면으로 사용자에게 반환한다.

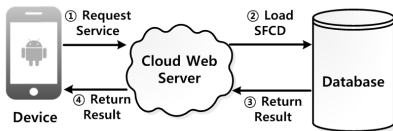
3.3 클라우드 서비스 기반 다중 커스터마이제이션

클라우드 서비스 기반 다중 커스터마이제이션 기법은 서비스 흐름제어 데이터를 서비스 흐름제어 데이터 객체에 담아 모바일 클라우드로 전송하는 과정은 푸시 서버 기반 다중 커스터마이제이션 기법과 동일하다.

그렇지만, 푸시 서버 기반 다중 커스터마이제이션이 푸시 메시지의 형태로 서비스 흐름제어 데이터를 다중 디바이스에 전달하여 일괄적으로 서비스를 변경하는 기법이라면, Fig. 11을 통해 볼 수 있는 모바일 클라우드 서비스 기반 다중 커스터마이제이션은 디바이스가 변경 서비스를 클라우드 서버와 동기화하여 서비스를 변경하는 기법이다.

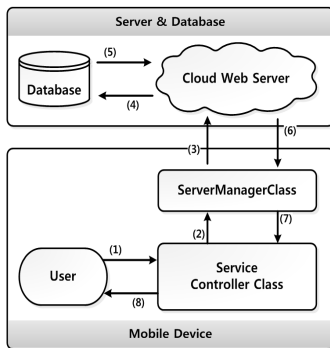


[Fig. 11] Multiple Customization Structure based on Mobile Cloud Service



[Fig. 12] Request SFCD to Cloud Web Server

Fig. 12은 디바이스에 서비스 데이터가 저장 되었지만, 실행을 위해서 반드시 동기화 과정을 거치는 구조를 나타낸다. 모바일 클라우드는 요청 내용을 토대로 서비스를 데이터베이스로부터 얻어와 모바일 디바이스로 전송한다.



[Fig. 13] Synchronization Process between Device and Mobile Cloud

모바일 디바이스와 클라우드 서비스와의 동기화는 Fig. 13과 같은 과정을 거친다.

- (1) 사용자가 ServiceControllerClass에 서비스 요청
- (2) ServerManagerClass에 SFCD를 요청
- (3) CloudWebServer에 SFCD를 요청
- (4) 데이터베이스에 SFCD를 로드
- (5) 데이터베이스에서 그에 상응하는 값을 Cloud WebServer에 반환
- (6) CloudWebServer는 전달 받은 값을 SFCD 객체에 담아 SFCD를 요청한 사용자 모바일 디바이스로 객체를 전달

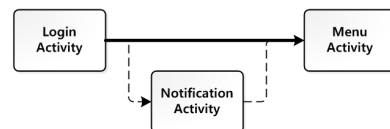
- (7) ServerManagerClass에서 SFCD 객체를 받으면, 객체를 ServiceControllerClass로 반환
- (8) SFCD를 분석하여 사용자에게 View를 리턴

사용자가 서비스를 요청하고, 받아오는 행위는 모두 ServiceController에서 처리한다. 푸시 서비스 기반 커스터마이제이션 기법의 ServiceController 클래스와 달리 ServerManager 클래스의 도움을 받아 데이터를 받아온다. ServerManager 클래스는 받아온 SFCD 값을 관련 객체에 담아 저장한 후, ServiceController 클래스를 다시 호출하여, 모바일 클라우드로 통신한 결과를 표시한다. 모바일 클라우드로 통신에서 실패 결과를 받는 경우, 반드시 사용자에게 위험 요소를 알린 후, 이전 서비스 흐름을 사용한다.

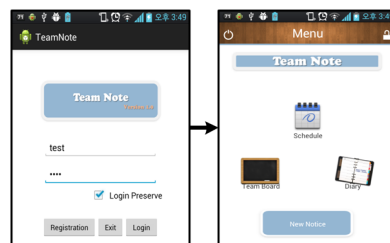
4. 실험 및 평가

본 연구에서는 모바일 클라우드를 이용하여 다중의 디바이스 서비스를 변경하는 다중 커스터마이제이션 기법을 제안하였다. 이에 대한 적합성과 실현 가능성을 검증하기 위해 사례연구로 팀원 간 일정을 공유하는 어플리케이션에 적용한다. 본 어플리케이션에서는 공지사항 서비스를 커스터마이제이션하는 사례를 구현한다.

본 실험은 서비스 흐름을 전환할 데이터를 모바일 클라우드로 저장하는 과정과 서비스를 변경하는 과정으로 구분하여 수행한다. 후자의 과정은 푸시서비스를 이용한 방법과 동기화를 이용한 방법으로 다시 나누어 수행한다.



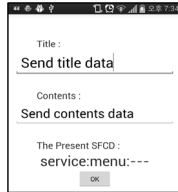
[Fig. 14] Concept Service Flow between Login and Menu Activity



[Fig. 15] Actual Service Flow between Login and Menu Activity

모바일 클라우드에 저장 되어 있는 기본 서비스 흐름은 Fig. 14처럼 로그인 액티비티에서 메뉴 액티비티로 이어지는 흐름을 지닌다. Fig. 15은 실제 액티비티 서비스에 대한 결과를 나타낸다.

Fig. 16은 공지사항을 추가하여 서비스를 커스터마이제이션하기 위한 액티비티다. 공지사항 제목, 공지 내용을 기입하고, 서비스 흐름제어 데이터는 디바이스 내부에서 만들어져 모바일 클라우드로 전송한다.

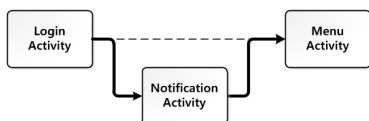


[Fig. 16] Input Customization Data in New Notification Activity

Fig. 16처럼 모바일 클라우드에 변경을 요청하면, 현 서비스를 제공 받는 다중 디바이스는 메뉴 액티비티가 아닌, 공지 액티비티로 전환되게 된다. 공지사항 데이터를 모바일 클라우드로 전송함과 동시에 서비스에 대한 변경 요청이 자동 수행된다.

다음은 서비스의 변경이 클라우드로 설정된 후, 변경 서비스를 디바이스에 전달하는 과정으로, 실험은 푸시 서비스와 동기화 방법으로 나누어진다. 두 실험에 적용되는 서비스 요청 값은 Fig. 16의 입력 값과 같다. Fig. 17의 The Present SFCD는 현재 서비스가 적용되고 있는 서비스 흐름제어 데이터를 의미한다. 본질적으로 사용자는 서비스 흐름제어 데이터를 확인 할 수 없지만, 변화되는 서비스 흐름제어 데이터를 알기위하여 임의 삽입되었다.

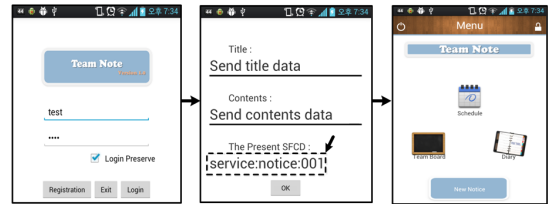
푸시서비스 기반 커스터마이제이션 기법에서 모바일 클라우드는 공지사항에 대한 정보를 전송 받게 되는 즉시, 전달받은 서비스 흐름제어 데이터와 공지사항의 데이터 값은 데이터베이스에 저장한 후에 공지사항 정보의 ID 값과 서비스 흐름 제어 데이터를 푸시서버의 인자 값으로 전달된다. 푸시서버는 수신 한 데이터를 인지하여 다중 디바이스에 데이터를 전송한다. 다중 디바이스는 푸시서비스에 대한 콜백 메소드를 통해 데이터를 받고, 그 즉시 저장한다. 어플리케이션을 재실행 시 서비스 흐름은 Fig. 17과 같이 변경되어 동작한다.



[Fig. 17] Changing Activity Flow

전환을 통해, 로그인 액티비티에서 이어지던 액티비티는 메뉴 액티비티가 아닌, 공지 액티비티로 변경되어 실행된다. 전자와 같이 전환 된 현 서비스흐름 상태는 서비스 변경 전까지 지속된다.

클라우드 동기화를 통한 커스터마이제이션 기법은 Fig. 16에 입력 값과 같다. 서버에서는 전달 받은 공지사항 데이터와 서비스 흐름제어 데이터를 저장한다. 서버에서는 서비스 동기화를 위해 아무런 행위가 없는 대신, 디바이스가 능동적으로 서비스 데이터를 클라우드로 요청하여 받아온다.



[Fig. 18] Changed service activity from Menu Activity to Notification Activity

서비스 흐름변경은 푸시 서비스 기반 커스터마이제이션 기법과 동일하며, 결과 또한 같다. Fig. 18는 서비스가 변경 된 두 실험의 결과 화면으로, The Present SFCD의 서비스 흐름제어 데이터가 Fig. 16의 The Present SFCD의 서비스 흐름제어 데이터와 다르다. 이는 서비스 흐름제어 데이터로 서비스가 변경되었음을 의미한다.

안드로이드 플랫폼을 기반으로 본문에서 제시한 다중 커스터마이제이션과 모바일 동기화 기법에 대한 검증은 위 실험을 통해 실제 적용 가능성을 알 수 있다. 또한 그 타당성을 입증하였다.

5. 결론 및 향후 연구과제

본 논문에서는 모바일 클라우드의 서비스 공유를 기반으로 디바이스와 동기화를 하는 다중 커스터마이제이션을 제안하였다. 다중 커스터마이제이션은 서비스 흐름제어 데이터를 모바일 클라우드로 저장 관리함으로써 가능하다. 즉, 서비스를 공유한다는 것은 모바일 클라우드로 서비스 흐름제어 데이터를 공유한다는 것이다. 이러한 서비스 제어는 디바이스가 모바일 클라우드로 요청하거나, 모바일 클라우드가 푸시 서버를 통해 다중 디바이스에 주입할 때 가능하다.

향후 다중 커스터마이제이션의 활용성을 보장하기 위해 디바이스와 모바일 클라우드 간에 서비스 동기화 과

정에서 발생하는 시간적 소모와 보안에 대해 연구하고, 사용자들의 서비스 변경의 혼선으로 인해 문제가 될 수 있는 서비스의 병목 현상에 대한 방안을 연구한다.

References

- [1] Lee G. C., "Mobile Cloud concepts and technology trends", Telecommunications Technology Association, p54-56, 2012
- [2] I. H. Bea., "Context Awareness Mobile Multimedia Service in Cloud Computing", Review of Korean society for internet information / v.11 no.4, pp.9-15, 2010
- [3] Gwon J. Y., "Samsung SDI Mobile Cloud Computing Propulsion Strategy and Construction Case", Korea information processing society review v.18 no.5 , pp.75 - 80 , 2011
- [4] Woo J. J., "Mobile Cloud Trends and Prospects", Korea information processing society review v.17 no.3, pp.107 - 115, 2010
- [5] "Smartphone and Push Alarm Service", A Magazine of Microsoftware, 2011
- [6] Yun S. H., Kim G. T., "Cloud Service Type and Mobile Cloud Service Case", Korea information processing society review v.18 no.5, pp.88 - 94, 2011
- [7] "A homepage for Google Cloud Storage Introduce", <https://developers.google.com/storage/>
- [8] J. B. Jung, "Mobile Cloud Market Trends and Preview", Korea information processing society review, v.18 no.5, pp.4-9, 2011
- [9] Jeon I. C., Choi S. J., Lee Y. T., "An implementation of wake-up method for T-DMB by android C2DM", The journal of the Institute of Webcasting, Internet and Telecommunication v.12 no.5, pp.283 - 290, 2012
DOI: <http://dx.doi.org/10.7236/JIWIT.2012.12.5.283>

예 준 호(Jun-Ho Ye)

[준회원]



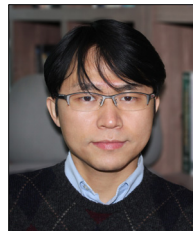
- 2009년 3월 ~ 현재 : 인하공전 컴퓨터시스템과 재학

<관심분야>

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

김 철 진(Chul-Jin Kim)

[종신회원]



- 2004년 2월 : 송실대학교 대학원 컴퓨터학과 (공학박사)
- 2004년 3월 ~ 2009년 2월 : 삼성전자 책임연구원
- 2009년 3월 ~ 현재 : 인하공전 컴퓨터시스템과 조교수

<관심분야>

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