

모바일 앱의 성능향상을 위한 커스터마이제이션 방안

조은숙¹, 김철진^{2*}

¹서일대학교 컴퓨터소프트웨어과, ²인하공업전문대학교 컴퓨터시스템과

A Customization Method for Mobile App.'s Performance Improvement

Eun-Sook Cho¹, Chul-Jin Kim^{2*}

¹Dept. of Computer Software, Seoil University

²Dept. of Computer System, Inha Technical College

요약 4차 산업혁명을 맞이해서 여러 분야에서 커스터마이제이션(Customization, 맞춤제작)이 주된 화두가 되고 있다. 4차 산업혁명인 인더스트리 4.0은 제조업에 사이버 물리 시스템(CPS: Cyber Physical Systems), 사물인터넷(IoT: Internet of Things), 클라우드 컴퓨팅(Cloud Computing)을 적용하고 있다. 그리고 이 인더스트리 4.0의 핵심 키워드 중 하나가 대량 맞춤제작이다. 이 맞춤제작을 통해 고객의 요구에 맞는 최적화 된 제품이나 서비스를 개발하여 제공하게 된다. 따라서 제품의 경쟁력은 높아지고 고객만족도는 향상되게 된다. 특히 사물 인터넷 기술의 확산으로 인해서 다양한 디바이스나 사물들 간의 원활한 서비스 연동을 위해서는 이러한 맞춤제작은 필수적인 부분이다. 모바일 환경에서도 모바일 앱(Mobile App.)들이 여러 다양한 모바일 디바이스에 맞춰서 고객에게 필요한 서비스들이 맞춤형으로 조립되어 작동되어야 한다. 따라서, 본 논문에서는 현재 많은 안드로이드나 애플의 ios 플랫폼 등을 기반으로 한 여러 다양한 모바일 디바이스들에서 이루어지고 있는 맞춤제작 구조의 성능을 향상시키기 위한 방안으로 클라우드 서버 기반의 모바일 맞춤제작 구조와 프로세스, 그리고 성능 측정 메트릭을 제시한다. 제시된 맞춤제작 구조와 프로세스, 그리고 성능 측정 메트릭을 여러 디바이스들에 적용해 본 결과 기존에 비해 맞춤제작에 걸리는 총 시간이 기존에 비해 반으로 줄어드는 효과를 가져올 수 있었다.

Abstract In the fourth industrial revolution, customization is becoming a conversation topic in various domains. Industry 4.0 applies cyber-physical systems (CPS), the Internet of Things (IoT), and cloud computing to manufacturing businesses. One of the main phrases in Industry 4.0 is mass customization. Optimized products or services are developed and provided through customization. Therefore, the competitiveness of a product can be enhanced, and satisfaction is improved. In particular, as IoT technology spreads, customization is an essential aspect of smooth service connections between various devices or things. Customized services in mobile applications are assembled and operate in various mobile devices in the mobile environment. Therefore, this paper proposes a method for improving customized cloud server-based mobile architectures, processes, and metrics, and for measuring the performance improvement of the customized architectures operating in various mobile devices based on the Android or IOS platforms. We reduce the total time required for customization in half as a result of applying the proposed customized architectures, processes, and metrics in various devices.

Keywords : Customization, Cloud Server, Customization Architecture, Forth Industrial Revolution, Mobile Application

1. 서론

산업혁명의 흐름은 증기기관에서 시작한 1.0, 전기 시

대인 2.0, 컴퓨터 시대인 3.0을 지나 지금은 새로운 4차 산업혁명인 4.0을 맞이하고 있다. 이러한 4차 산업혁명의 선두 두자인 독일이 4차 산업혁명을 인더스트리 4.0

본 논문은 2016년도 서일대학교 학술연구비에 의해 연구되었음.

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

Tel : +82-32-870-2338 email : cjkim@inhac.ac.kr

Received October 19, 2016

Accepted November 10, 2016

Revised (1st October 27, 2016, 2nd November 1, 2016)

Published November 30, 2016

으로 정의하고 있는데, 이 정의의 중심에는 사물 인터넷 (IoT)과 이와 관련된 다양한 서비스들을 제시하고 있다[1-6].

인더스트리 4.0의 핵심 키워드에는 대량 맞춤제작 (Customization)으로의 이전을 다루고 있으며, 이를 통하여 달성되고 있는 목표는 사용자들이 맞춤제작을 통한 양질의 경험을 가지게 되고, 제품 자체가 플랫폼으로 전환하고 있다는 부분에 있다.



Fig. 1. Industry 4.0

Fig. 1에 나타난 것처럼 사용자들은 제품으로부터 자신들이 원하는 것을 선택할 수 있게 된다.

이처럼 이제 여러 분야에서 맞춤제작이 주된 화두가 되고 있다[7]. 본 논문에서는 현재 안드로이드[8]나 애플

의 ios[9]플랫폼 등을 기반으로 한 많은 모바일 디바이스에서 이루어지고 있는 맞춤제작 구조의 성능을 향상시키기 위한 방안으로 클라우드 서버기반의 모바일 맞춤제작 구조와 프로세스를 제시하고, 이를 검증하기 위한 도구로 성능 측정 메트릭을 제시하고자 한다[10-11].

본 논문의 구성으로 2장에서는 관련연구로 기존 맞춤제작 구조, 맞춤제작 프로세스 및 성능 측정 메트릭에 대해 설명하며, 3장에서는 본 논문에서 제안하는 클라우드 서버 기반의 모바일 맞춤제작 구조, 프로세스 및 성능 측정 메트릭을 제안한다. 4장에서는 3장에서 제안한 기법을 실제 사례에 적용하여 측정된 결과를 제시하고, 이를 통해 본 논문에서 제시한 기법이 가져온 성능향상을 객관적으로 검증하고자 한다. 마지막으로 5장에서 결론 및 향후 연구 과제를 제시한다.

2. 관련 연구

2.1 기존 맞춤제작 구조

기존 맞춤제작 구조는 Fig. 2와 같은 구조로 되어 있다. Fig. 2에서처럼, 각 모바일 디바이스에서 서비스에 대한 맞춤제작이 필요할 경우, 서비스 맞춤제작을 담당

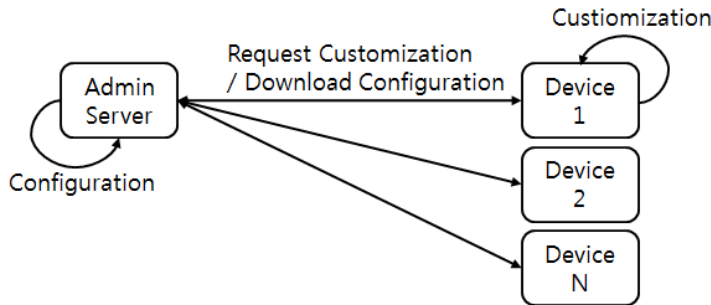


Fig. 2. Existing Customization Architecture in Mobile Service

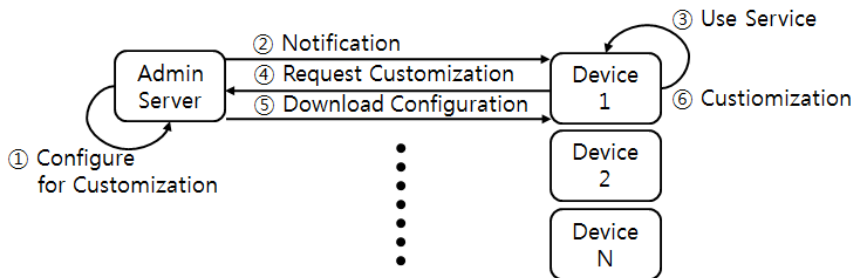


Fig. 3. Existing Customization Process in Mobile Service

하는 관리서버(Admin Server)에 맞춤제작 요청을 보내면, 해당 서버가 맞춤제작 설정을 변경한다. 그런 후, 요청한 모바일 디바이스가 설정 변경한 사항을 다운로드 받아서, 자신의 디바이스에 맞춤제작을 수행하게 된다.

이와 같은 구조에서는 관리 서버에서 일괄적으로 설정정보를 모바일 디바이스에서 전송하므로 설정 서비스의 성능이 저하되는 문제점을 갖고 있다.

2.2 기존 맞춤제작 프로세스

Fig. 2의 구조를 기반으로 하는 맞춤제작 프로세스는 Fig. 3과 같은 순서로 진행된다.

Fig. 3에 제시된 것처럼, 1단계로는 관리서버에서 맞춤제작을 위한 환경 설정을 하게 된다. 환경 설정이 끝나면, 2단계인 Notification 단계로 해당 디바이스에 맞춤제작 서비스를 알려준다. 그러면 3단계로, 맞춤제작 서비스가 알려진 해당 모바일 디바이스에서 해당 서비스를 사용할 것인지 여부를 판단한다. 만일 맞춤제작 서비스를 사용할 경우, 4단계로 모바일 디바이스에서 관리 서버로 맞춤제작 요청 메시지를 보내게 된다. 그런 후 5단계로 관리 서버로부터 맞춤제작을 위한 환경설정 내역을 다운로드 받게 되고, 마지막 6단계에서 해당 디바이스에서 맞춤제작이 이루어지게 된다.

2.3 기존 맞춤제작 구조의 성능측정 메트릭

Fig. 2와 같은 맞춤제작 구조의 성능을 측정[7-9]하기 위해서는 7개의 측정 요소가 필요했다.

첫 번째 요소인 설정시간은 여러 다양한 디바이스 별로 맞춤제작 해야 할 정보들이 다르기 때문에 각 디바이스 별로 필요한 설정 정보들을 셋팅하는 시간이다.

두 번째 요소인 알림 시간은 서버에서 해당 모바일 디바이스로 설정을 맞춤제작 서비스 요청에 대한 설정 완료를 알려주는 시간을 말한다.

세 번째 요소인 알림 시간은 서버에서 각각의 디바이스에서 서버로 맞춤제작을 요청하는데 걸리는 시간을 말한다.

네 번째 요소인 다운로드 시간은 각각의 디바이스로 맞춤제작 관련 설정 정보가 다운로드 되는 시간을 말한다.

다섯 번째 맞춤제작 시간은 각각의 디바이스로 맞춤제작 하는데 소요되는 시간을 말한다.

여섯째 요소인 디바이스 개수는 맞춤제작이 이루어지는 디바이스들의 총 수를 말한다.

- ① 설정시간: CFT(Configuration Time)
- ② 알림시간: NT(Notification Time)
- ③ 요청시간: RT(Request Time)
- ④ 다운로드시간: DT(Download Time)
- ⑤ 맞춤제작시간: CT(Customization Time)
- ⑥ 디바이스 개수: n

이러한 요소들을 기반으로 정의된 성능 측정 메트릭은 다음과 같다.

$$PF_{EC} = CFT + \sum_{i=1}^n NT_i + \left(\sum_{i=1}^n (RT_i + DT_i + CT_i) \right) \times 0.2$$

3. 모바일 맞춤제작 성능 향상 방안

이 장에서는 기존에 이뤄졌던 모바일 맞춤제작 구조, 프로세스, 그리고 성능 측정 메트릭을 토대로 성능 향상을 위해 클라우드 서버를 중개로 이용한 클라우드 서버 기반의 맞춤제작 구조, 프로세스 및 성능 측정 메트릭을 제안하고자 한다.

3.1 성능향상을 위한 모바일 맞춤제작 구조

기존의 관리 서버 기반의 맞춤제작 구조는 서비스의 독립성을 통해 서비스 흐름을 제어할 수 있는 장점에도 불구하고, 다중 디바이스에 서비스를 주입하기에는 서비스를 수동으로 변경하는 단점으로 인해 개개의 디바이스를 조작하는 시간허비와, 서비스가 변경된 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 배포함으로써 기존 구조에서 발생하는 위험성을 해소할 수 있다.

그리고 무엇보다 기존의 모바일 서비스 맞춤제작 구조가 관리 서버에서 일괄적으로 설정정보를 모바일 디바이스에서 전송하므로 설정 서비스의 성능이 저하되므로 본 연구에서 제안하는 클라우드 기반의 맞춤제작 구조는 관리 서버에서 클라우드 서버에 설정 정보를 공유함으로써 설정 서비스의 성능 저하 문제를 해결할 수 있다[15-16].

Fig. 4에서와 같이 관리 서버는 클라우드 서버에 설정 정보를 공유하면 클라우드 서버는 설정정보를 해당 모바일 디바이스에 푸시[10]하여 맞춤제작을 수행한다.

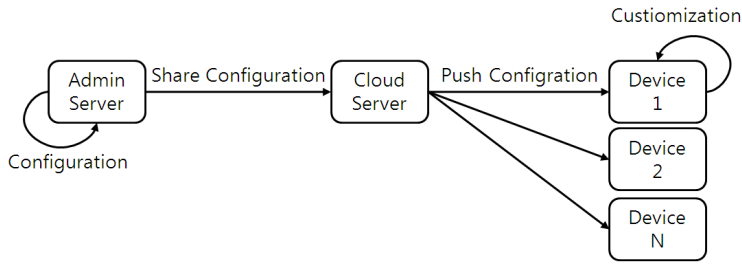


Fig. 4. Customization Architecture based on Cloud Server

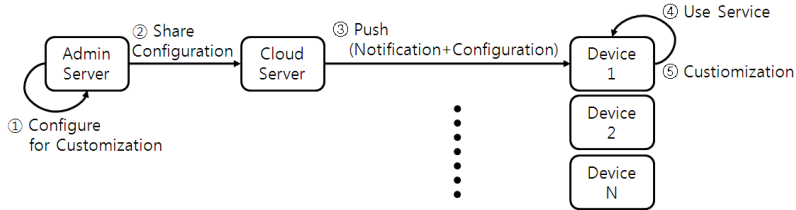


Fig. 5. Customization Process based on Cloud Server

3.2 성능 향상을 위한 모바일 맞춤제작 프로세스

성능향상을 위한 클라우드 서버 기반의 환경 하에서 서비스되는 프로세스는 다음과 같다.

- 1) 맞춤제작 설정
- 2) 설정 정보 공유
- 3) 맞춤제작 알림
 - 푸시서비스 이용
 - 설정 정보 포함
- 4) 맞춤제작

클라우드 서버 기반의 맞춤제작 시점은 디바이스에서 실제 서비스가 사용되는 시점에 맞춤제작 기능이 실행된다. 따라서 기존 구조와 프로세스가 가졌던 성능저하의 문제가 해결되게 된다. 물론 이 때, 맞춤제작은 전체 디바이스 중에서 20%가 동시에 맞춤제작 된다는 가정 하에서 이루어지며, 기존 맞춤제작 구조도 동일하게 20%로 적용한 것이다.

3.3 성능 향상을 위한 모바일 성능측정 메트릭

본 논문에서 제안하는 클라우드 서버 기반의 맞춤제작 구조와 프로세스를 기반으로 기존의 모바일 성능측정 메트릭을 새롭게 재정의하였다. 기존의 성능측정 메트릭보다 측정에 들어가는 항목들이 줄어들었고, 항목의 내

용도 공유시간, 푸쉬 시간 등과 같은 요소가 새롭게 대체되었다. 새롭게 추가된 공유시간은 관리 서버에서 맞춤제작 설정이 완료되고 난 이후, 클라이드 서버로 설정정보를 공유하는데 걸리는 시간을 말한다. 그리고 푸쉬 시간은 클라우드 서버에서 맞춤제작 할 해당 디바이스에 알려주는 시간인데, 이는 기존의 메트릭에서는 알람시간과 설정시간이 달랐기 때문에 분리되었지만, 이제는 실제 서비스가 실행될 때 알려지기 때문에 푸쉬 시간 하나로 통합되어 이루어지기 때문이다.

- ① 설정시간: CFI(Configuration Time)
- ② 공유시간: ST(Sharing Time)
- ③ 푸쉬시간: PT(Push Time)
- ④ 맞춤제작 시간: CT(Customization Time)
- ⑤ 디바이스 개수: n

$$PF_{EC} = CFT + ST + \sum_{i=1}^n PT_i + \left(\sum_{i=1}^n CT_i \right) \times 0.2$$

4. 평가

이 장에서는 기존 맞춤제작 구조에 대한 성능 측정 메트릭과 본 논문에서 제안한 성능측정 메트릭을 가지고 모바일 서비스 맞춤제작을 측정한 결과를 제시하여, 본 논문에서 제안한 기법의 객관적 타당성을 제시하고자 한다.

ExistingCustomization							
Configuration Time	Notification Time	Request Time	Download Time	Customization Time	n (# Device)		Performance
10	10	10	10	20	1		28
10	10	13	10	20	10		196
10	10	16.9	10	20	100		1,948
10	10	21.97	10	20	1,000		20,404
10	10	28.561	10	20	10,000		217,132
10	10	37.1293	10	20	100,000		2,342,596
10	10	48.26809	10	20	1,000,000		25,653,628

Fig. 6. Experiment Results in Existing Customization Architecture

ThisCustomization							
Configuration Time	Sharing Time	Push Time		Customization Time	n (# Device)		Performance
10	10	10		20	1		34
10	10	10		20	10		160
10	10	10		20	100		1,420
10	10	10		20	1,000		14,020
10	10	10		20	10,000		140,020
10	10	10		20	100,000		1,400,020
10	10	10		20	1,000,000		14,000,020

Fig. 7. Experiment Results in This Customization Architecture

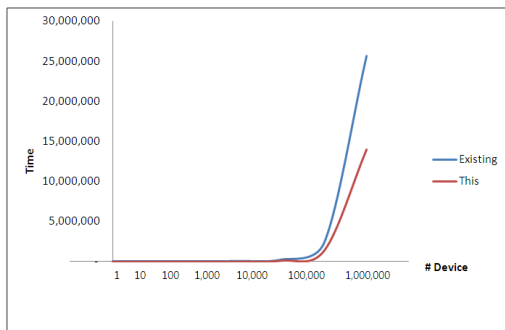


Fig. 8. Compare Experiment Results

실험결과 Fig. 6, Fig. 7, Fig. 8에서와 같이 설정 성능은 10만대이하의 디바이스에서는 별 차이는 보이지 않지만 10만대가 넘어가면 기존 구조의 설정보다 본 구조의 설정에서 성능 향상이 됨을 파악할 수 있다.

5. 결론 및 향후 연구과제

본 연구를 통해 기존의 맞춤제작 구조와 프로세스에서 야기되었던 성능 저하의 문제점을 클라우드 서버를 기반으로 한 맞춤제작 구조와 프로세스로 변경함으로써

다음과 같은 기대효과들을 얻을 수 있다. 첫째, 여러 다양한 디바이스들에 맞춤제작 되더라도 실제 서비스가 실행되는 시점에 맞춤제작 되기 때문에 성능이 저하되지 않는다. 둘째, 새로운 유형의 모바일 디바이스가 추가되어 연동되더라도 클라우드 서버에 설정 정보만 반영하면 되기 때문에 모바일 앱의 맞춤제작 성능에 영향을 미치지 않는다. 본 연구를 통해 도출된 연구 결과는 안드로이드 기반의 모바일 애플리케이션을 대상으로 적용하였지만, ios와 같은 플랫폼에도 적용할 수 있기 때문에 여러 다양한 모바일 플랫폼에 반영할 수 있는 개방형 맞춤제작 방안이라 할 수 있다.

향후 연구로는 제안된 성능 측정 메트릭을 좀 더 연구하여 자동으로 측정할 수 있는 자동화 도구로 개발하여 모바일 앱 개발 또는 품질 관리시에 반영하는 것이다.

References

[1] Jung Tae Kim, "Analyses of Requirement of Security based on Gateway Architecture for Secure Internet of Things," Asia-pacific Journal of Multimedia Services Convergent with Art, Humanities, and Sociology, Vol.6, No.3, pp. 461-470, Mar. 2016.
DOI: <http://dx.doi.org/10.14257/AJMAHS.2016.03.02>

- [2] Namje Park, "Smart Centralized Remote Security Service Provisioning Framework for Open ICT Environment," Asia-pacific Journal of Multimedia Services Convergent with Art, Humanities, and Sociology, Vol.6, No.2, pp. 81-88, Feb. 2016.
DOI: <http://dx.doi.org/10.14257/AJMAHS.2016.46>
- [3] Kee-Hyun Park, Jong-Hwi Lee, Min-Woo Woo, Joon-Suu Park, "Development of an IoT System Based on the oneM2M Communication Protocol," Asia-pacific Journal of Multimedia Services Convergent with Art, Humanities, and Sociology, Vol.6, No.3, pp. 41-49, Mar. 2016.
DOI: <http://dx.doi.org/10.14257/AJMAHS.2016.03.14>
- [4] "Gi-Hong Ki, "What is the appropriate Policy Direction to develop Internet of Things?," Asia-pacific Journal of Multimedia Services Convergent with Art, Humanities, and Sociology, Vol.6, No.2, pp. 89-100, Feb. 2016.
DOI: <http://dx.doi.org/10.14257/AJMAHS.2016.48>
- [5] Woo-Chul jung, Sueng-Soo Lee, Joon-Hoon Park, "Design of optimal snowmelting system with snowfall image processing based on the IoT technology," Asia-pacific Journal of Multimedia Services Convergent with Art, Humanities, and Sociology, Vol.5, No.6, pp.521-530, Dec. 2015.
DOI: <http://dx.doi.org/10.14257/AJMAHS.2015.12.38>
- [6] O.G. Min, H. Y. Kim, G. H. Nam, Trends in Technology of Cloud Computing, KINX, vol. 24, no. 4, pp. 1-13, Aug. 2009.
- [7] Chul Jin Kim, Eun Sook Cho, "A Variability Design and Customization Technique for Improving Generality", Journal of Korea Multimedia Society, vol. 9, no. 8, pp. 1076-1085, Aug. 2006.
- [8] Google Android [Online]. <http://www.android.com>
- [9] Apple ios-7, <http://www.apple.com>
- [10] Lee G. C., "Mobile Cloud concepts and technology trends", Telecommunications Technology Association, pp. 54-56, 2012.
- [11] Woo J. J., "Mobile Cloud Trends and Prospects", Korea information processing society review vol. 17, no. 3, pp. 107-115, 2010.
- [12] Jeffrey S. Poulin, "Measuring Software Reusability," Proceeding of the Third International Conference on Software Reuse, Rio de Janeiro, Brazil, vol. 1, no. 4, pp. 1-5, Nov. 1994.
- [13] ISO/IEC, *FCD 9126-1.2 Information Technology Software product quality-Part 1:Quality model*, 1998.
- [14] ISO/IEC JTC1/SC7 N2419 "DTR 9126-2: Software Engineering - Product Quality Part 2 - External Metrics", 2001.
- [15] J. B. Jung, "Mobile Cloud Market Trends and Preview", Korea information processing society review, vol. 18, no. 5, pp. 4-9, 2011.
- [16] Yun S. H., Kim G. T., "Cloud Service Type and Mobile Cloud Service Case", Korea information processing society review, vol. 18, no. 5, pp. 88-94, 2011.

조 은 숙(Eun-Sook Cho)

[정회원]



- 1993년 2월 : 동의대학교 전산통계학과 졸업(이학사)
- 1996년 2월 : 숭실대학교 대학원 컴퓨터학과(공학석사)
- 2000년 2월 : 숭실대학교 대학원 컴퓨터학과(공학박사)
- 2000년 9월 ~ 2005년 2월 : 동덕여자대학교 정보학부 강의전임교수
- 2005년 3월 ~ 현재 : 서일대학교 컴퓨터 소프트웨어과 부교수

<관심분야>

Software Engineering, Embedded Software, Mobile Computing

김 철 진(Chul-Jin Kim)

[정회원]



- 1996년 2월 : 경기대학교 전자계산학과(학사)
- 1998년 2월 : 숭실대학교 컴퓨터공학부(공학석사)
- 2004년 2월 : 숭실대학교 컴퓨터공학부(공학박사)
- 2004년 3월 ~ 2009년 2월 : 삼성전자 책임연구원
- 2009년 3월 ~ 현재 : 인하공전 컴퓨터시스템과 부교수

<관심분야>

Software Engineering, Embedded Software, Mobile Computing