

멀티 디바이스 상호작용을 위한 차세대 모바일 프레임워크 설계

**Designing Advanced Mobile Frameworks for Multi-device
Interaction**

오상은
소프트웨어학과
아주대학교
(16499) 경기도 수원시 영통구
월드컵로 206, 대한민국
sangeunoh@ajou.ac.kr

요약

최근 모바일 및 IoT 기술의 발전으로, 우리는 다양한 형태의 스마트 기기들로 이루어진 다중 기기 환경 속에서 살아가고 있다. 이러한 추세에 따라 여러 스마트 기기들을 어떻게 협력적으로 함께 사용할지에 대한 관심이 점점 커져가고 있으며, 멀티 디바이스 상호작용이 사용자에게 새롭고 유용한 경험을 제공할 수 있음을 보여주는 사례가 등장하고 있다. 본 논문에서는 멀티 디바이스 상호작용을 위해 여러 기기의 앱들을 마치 하나의 단일 기기에서 동작하는 것처럼 만들어주는 새로운 시스템 수준의 추상화 기법을 소개한다. 이를 통해, 기존 단일 기기를 위해 만들어진 애플리케이션들을 멀티 디바이스 환경에서 어떻게 쉽게 이용할 수 있는지를 보여준다.

키워드 : Multi-device Computing, Mobile Frameworks, User Experience, Single-device Abstraction

1. 서론

최근 모바일 및 IoT 기술의 발전으로, 우리는 다양한 형태의 스마트 기기들로 이루어진 다중 기기 환경 속에서 살아가고 있다. 이러한 추세에 따라 여러 스마트 기기들을 어떻게 협력적으로 함께 사용할지에 대한 관심이 점점 커져가고 있으며, 멀티 디바이스 상호작용이 사용자에게 새롭고 유용한 경험을 제공할 수 있음을 보여주는 여러 사례가 등장하고 있다.

하지만 현재 모바일 프레임워크는 애초에 이러한 멀티 디바이스 상호작용을 지원하도록 설계되지 않았기 때문에, 멀티 디바이스 환경을 위한 적절한 추상화 (abstraction) 기법이 부족한 실정이다. 이 때문에, 현재 대부분의 멀티 디바이스 상호작용은 앱-레벨 접근 방식으로 이루어진다 [1-3]. 즉, 멀티 디바이스 환경을 위해 작성된 커스텀 앱들이 각 디바이스에 미리 설치되어야 하며, 커스텀 앱들이 서로 통신함으로써 멀티 디바이스 상호작용이 이루어질 수 있다. 이러한 방식은 개발자에게 멀티 디바이스 앱 개발을 위한 높은 난이도의 개발 수고가 요구되며, 단일 기기를 위해 작성된 기존 상용화 앱들은 멀티 디바이스 환경에 맞게 사용할 수 없기 때문에 멀티 디바이스 활용도가 극히 제한되는 문제들이 존재한다.

본 논문에서는 이러한 문제들을 근본적으로 해결할 수 있는 시스템-레벨 접근 방식을 소개한다. 즉, 여러 기기에 존재하는 앱들이 마치 하나의 단일 기기에서 동작하는 것처럼 만들어주는 단일 기기 추상화 기법을 시스템 수준에서 제공함으로써, 현재 존재하는 멀티 디바이스 환경과 모바일 프레임워크 사이의 괴리를 좁힐 수 있다. 즉, 이러한 단일 기기 추상화 기법을 통해 개발자는 단일 기기 앱 프로그래밍 모델을 이용하여 멀티 디바이스 앱들을 쉽게 제작할 수 있으며, 사용자는 기존에 출시된 단일 기기 앱들을 별도의 수정없이 멀티 디바이스 환경에 맞게 그대로 이용할 수 있는 장점을 가질 수 있다.

2. 단일 기기 추상화 기법 소개

그림 1과 같이, 멀티 디바이스 모바일 프레임워크가 제공할 수 있는 단일 기기 추상화 기법에는

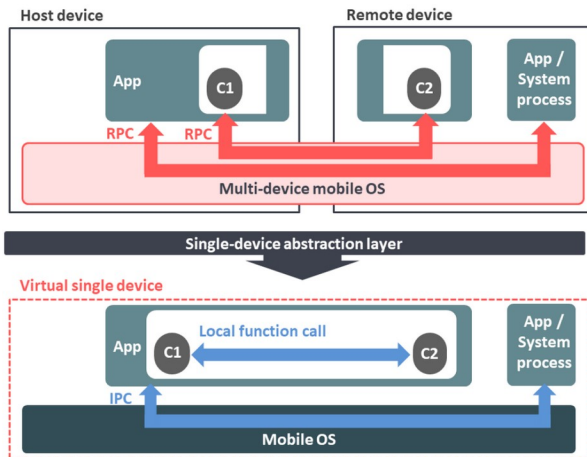


그림 1. 단일 기기 추상화 기법의 개요

두 가지 종류가 존재하며, 이를 위한 새로운 멀티 디바이스 모바일 프레임워크들이 연구된 바 있다.

첫 번째는 Mobile Plus [4]라는 모바일 프레임워크로, IPC (inter-process communication) 메커니즘을 통해 발생하는 앱들 사이의 상호작용 (inter-app interaction)을 RPC (remote procedure call) 형태로 멀티 디바이스 환경에 맞게 확장하는 추상화 기법을 제시한다. 이를 통해 Mobile Plus는 여러 기기의 앱들이 각자의 기능을 공유할 수 있도록 만들어 줄 수 있다. 기본적으로 단일 기기에서는 여러 앱들이 서로의 기능을 공유하기 위해 IPC를 통해 상호작용을 한다. 제안한 프레임워크에서 제공하는 추상화는 여러 기기의 앱들이 마치 동일한 단일 기기에서 존재하는 것처럼 상호작용을 지원하기 때문에, 기존 상용화 앱들도 별도의 수정 없이 다중 기기 환경에서의 기능 공유가 가능해지는 것이다. 예컨대, 사용자는 그림 2와 같이 스마트 TV에서 센서 기반의 레이싱 게임을 할 때, Mobile Plus를 통해 스마트폰의 센서 기능들을 원격으로 사용함으로써, 이전보다 더욱 편한 게임 컨트롤을 즐길 수 있다. 실제로, Mobile Plus는 27개의 유명 상용화 안드로이드 기반 앱들을 통해 22개의 유용한 시나리오들이 가능함을 보였다.

두 번째는 FLUID [5]라는 모바일 프레임워크로, 로컬 함수 호출을 통해 앱 내부의 컴포넌트들

사이에서 발생하는 상호작용 (intra-app interaction)을 멀티 디바이스 환경에 확장하는 추상화 기법을

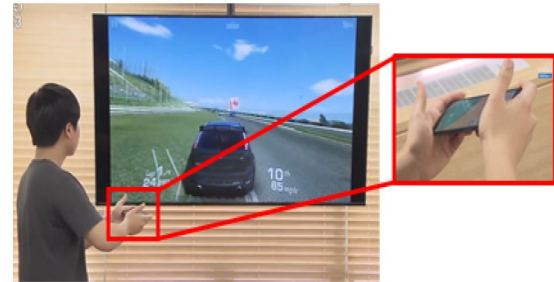


그림 2. Mobile Plus를 통한 remote sensor 시나리오

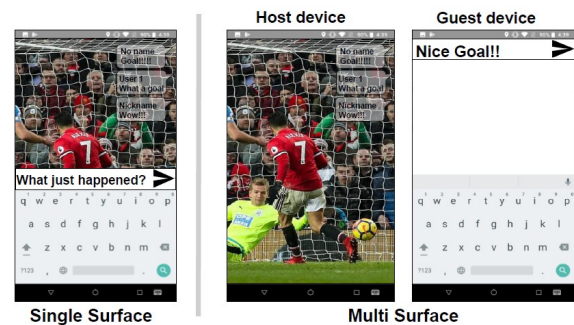


그림 3. FLUID를 통한 라이브 스트리밍 앱에서의 UI 분산 시나리오

제시한다. 이를 통해 FLUID는 하나의 앱이 여러 기기의 화면들을 유연한 방식으로 사용할 수 있도록 지원해 줄 수 있다. 즉, FLUID는 하나의 앱에 존재하는 여러 UI (user interface) 요소들을 여러 기기에 분산시킴으로써, 사용자가 원하는 대로 여러 기기의 화면에 보일 UI 요소들을 유연하게 구성할 수 있다. 여기서 제공하는 추상화 기법은 여러 기기에 분산된 UI 요소들과 앱 로직 간의 상호작용이 마치 동일한 단일 기기/단일 프로세스 내에서 이루어지는 것처럼 만들어 주기 때문에, 기존 상용화 앱들이 별도의 수정 없이도 여러 기기의 화면을 함께 사용할 수 있도록 만들어준다. 예컨대, 그림 3과 같이 사용자는 하나의 앱에 있는 여러 UI들을 다른 기기기로 분산시킴으로써, 라이브 스트리밍 앱이나 멀티-뷰 앱 등 여러 멀티미디어

기반 앱들을 이전보다 더 편하게 이용할 수 있다. FLUID는 실제 20개의 유명 상용화 앱들을 이용하여 10개의 유용한 멀티 디바이스 시나리오들이 가능함을 성공적으로 보였다.

3. 현재 멀티 디바이스 프레임워크의 한계 및 향후 연구 방향

앞서 언급한 것처럼, 단일 기기 추상화 기법은 기존 단일 기기 앱을 수정하지 않고도 멀티 디바이스 환경에서 transparent하게 동작할 수 있도록 하는 기능을 제공하기 때문에, 기존에 상용화된 많은 앱들을 멀티 디바이스 환경으로 끌어들이 수 있는 큰 장점을 지닌다.

하지만 단일 기기 추상화 기법을 위해 연구된 Mobile Plus와 FLUID는 필연적으로 Android와 같은 기존 모바일 프레임워크의 내부를 대폭 수정해야 하는 단점을 지니고 있다. 이러한 부분은 시스템을 실제로 사용자의 디바이스에 배치하는 것을 어렵게 만든다 (low deployability). 즉, 구글, 애플, 삼성과 같은 스마트폰 벤더들이 이러한 방식을 채택하지 않는 이상 사용자들의 디바이스에서는 본 논문에서 제시한 멀티 디바이스 상호작용을 수행하기가 현실적으로 어려운 것이다.

이러한 문제를 해결하기 위해, 기존 싱글 디바이스 앱을 멀티 디바이스 앱으로 자동 변환해 주는 프레임워크 기술에 대한 연구가 필요할 것으로 보인다. 즉, 개별 모바일 앱이 다른 디바이스의 자원 및 서비스를 transparent하게 사용할 수 있도록 앱 기능을 확장해줌으로써, 모바일 프레임워크에 대한 수정 없이도 멀티 디바이스 상호작용을 지원할 수 있다. 이는 코드 삽입 (code instrumentation) 기법을 통해 개발이 가능할 것으로 보이며, 이 과정에서 코드 삽입 오버헤드 완화 문제, 애플리케이션 보안 유지 문제 등의 이슈들을 함께 고려할 필요가 있을 것으로 예상된다.

4. 결론 및 향후 연구

본 논문에서는 현재 멀티 디바이스 상호작용 방식의 문제점을 논하고, 이를 근본적으로 해결할 수 있는 멀티 디바이스 모바일 프레임워크의 단일 기기 추상화 기법을 소개하였다. 특히, Mobile Plus와 FLUID라는 새로운 멀티 디바이스 프레임워크들이 제공하는 앱 간 상호작용 추상화와 앱 내부 상호작용 추상화에 대해 살펴보았으며, 이 단일 기기 추상화 기법들이 기존 상용화 앱들을 성공적으로 멀티 디바이스 환경으로 끌어올 수 있음을 보였다.

Acknowledgement

"본 연구는 과학기술정보통신부 및 정보통신기획평가원의 대학 ICT 연구센터육성지원 사업의 연구결과로 수행되었음" (IITP-2023-2018-0-01431)

참고 문헌

- [1] P. Khlebovich. IP Webcam.
<https://goo.gl/FQgQst>
- [2] W. Morrison. WiFi Speaker.
<https://goo.gl/N128Ar>
- [3] T. Richardson, Q. Staord-Fraser, K. R. Wood, and A. Hopper, "Virtual network computing", IEEE Internet Computing, 1998.
- [4] Sangeun Oh, Huck Yoo, Dae R. Jeong, Duc Hoang Bui, and Insik Shin, "Mobile Plus: Multidevice Mobile Platform for Cross-device Functionality Sharing", in ACM MobiSys'17
- [5] Sangeun Oh, Ahyeon Kim, Sunjae Lee, Kilho Lee, Dae R. Jeong, Steven Y. Ko, and Insik Shin, "FLUID: Flexible User Interface Distribution for Ubiquitous Multi-device Interaction", in ACM MobiCom'19