

RESTful 기반의 Android - PC간 동영상 이어보기 구현

김정길* 종신회원

Implementation of a Continuous Playing Schemes on Android - PC Environment Based On RESTful

Cheong Ghil Kim* *Lifetime Members*

요 약

최근 모바일 디바이스 보급의 확산으로 다양한 기종의 디바이스를 다수 보유중인 사용자의 수가 지속적으로 증가하고 있는 추세이다. 이에 따라 하나의 미디어 콘텐츠를 사용자가 보유한 다양한 기기에서 서로 공유해야 할 필요성이 증가하고 있으며, 이는 연결성, 이동성, 통합성 기반의 N-screen 서비스의 확대를 가지고 왔다. N-screen 서비스는 하나의 미디어 콘텐츠를 공유하여 사용자가 보유한 다양한 디바이스에서 서로 이어보기가 가능하게끔 서비스를 제공한다. 그러나 N-screen 서비스들은 서비스 제공 사업자의 전용 플랫폼에서 시행되는 문제점을 가지고 있다. 따라서 본 논문은 개방형 서비스 플랫폼에서의 이어보기 기능의 설계를 위하여 시험적으로 이어보기 기능을 제공하는 RESTful 기반의 이어보기 기능 제공이 가능한 프로토타입 프로그램을 개발하였으며 동작검증은 Android - PC간 시행하였다.

Key Words : N-screen, Continuous playing, Media server, Android media player, Window media player

ABSTRACT

In recent days, the number of users having multiple devices according to time and places continue to increases with the help of wide spreading plurality of mobile devices. Consequently, the need for a user to share a media content on his/her multiple devices; furthermore, this movement brings the expansion of N-screen service focusing on connectivity, mobility, and integrity. N-screen service is platform that mediates the use of content or services on multiple devices with the continuous playing scheme. However, N-screen services have the problem of being provided exclusively by a service provider. This paper aims to implement a continuous playing scheme based on RESTfull and on an open service platform; the prototype was successfully implemented on the Android - PC environment.

I. 서 론

현재 PC 중심의 컴퓨팅 환경이 스마트폰 또는 태블릿 PC 등의 모바일 기반으로 급격히 이동하면서 지난 10여년의 변화를 넘어선 새로운 패러다임 변화를 맞이하고 있으며, 클라우드 컴퓨팅[1]과 N-screen[2,3] 등은 인터넷 연결성, 소프트웨어, 하드웨어, 콘텐츠와 서비스가 모두 하나로 집결되는 총체적인 IT 환경을 잘 대변하기 때문이다. 특히 N-screen 서비스는 인터넷이 연결되는 TV가 증가하고 스마트폰을 통한 무선 인터넷이 증가하면서 사용자들은 미디어 콘텐츠를 끊임 없이 장소에 구애받지 않고 이용하고자하는 희망이 증대하여 그 중요성이 크다. 이는 스크린 확장 개념의 서비스로

사용자가 보유한 콘텐츠를 인터넷상의 서버에 저장하여 모바일, PC 등의 다양한 매체에서 마치 하나의 공유 스크린처럼 활용할 수 있다는 특징을 지닌다.

그러나 이러한 N-screen 서비스들은 서비스 제공 사업자의 전용 플랫폼에서 시행되는 폐쇄적인 구조를 가지고 있어 일련의 솔루션 외의 플레이어에서는 서비스를 제공받을 수 없으며, N-screen 서버의 리소스를 참조하고자 할 때에도 별도의 전용 애플리케이션이 필요한 실정이다.

본 논문에서는 N-screen의 기본 개념을 바탕으로, OpenAPI 사용하여 RESTful[4,5] 방식의 웹서비스 디자인 표준을 적용한 멀티 디바이스 및 멀티 플레이어 환경에서 효율적인 이어보기 기능을 제공하기 위한 서비스의 제안 및 구

* 본 논문은 2013년도 남서울대학교 교내연구비 지원에 의하여 연구되었음.

*남서울대학교 컴퓨터학과 (cgkim@nsu.ac.kr)

접수일자 : 2013년 11월 8일, 수정완료일자: 2013년 11월 14일, 최종게재확정일자: 2013년 11월 15일

현을 목적으로 한다.

본 논문의 구성은 2장에서 관련 연구와 3장에서 목표 구현 시스템의 구조를 설명한다. 4장에서 구현 방법을 소개하며, 5장에서는 구현 결과의 성능 검증을 위하여 이어보기 동영상 크기에 따른 지연시간을 측정하였다. 최종적으로 6장에서 결론을 제시한다.

II. 관련 연구

2.1 N-screen 서비스

멀티스크린 서비스의 일환으로 N-screen 서비스는 사용자가 자신의 목적에 맞는 다양한 디바이스(IPTV, DCATV, 스마트TV, 스마트폰, 스마트패드)를 자유롭게 이용할 수 있는 서비스를 말한다. 방송과 통신 간 융합서비스가 등장한 이후, 민간·공공분야에서는 다양한 매체를 통한 서비스를 사용자들에게 제공해왔다. 하지만 이러한 서비스들은 주로 단일 기기나 서비스에 종속된 서비스들로, 기기와 서비스 간의 연결이 플랫폼이나 OS로 원활하게, 사용자들에게 끊임없이(seamless) 제공되는 경우는 많지 않았었다.

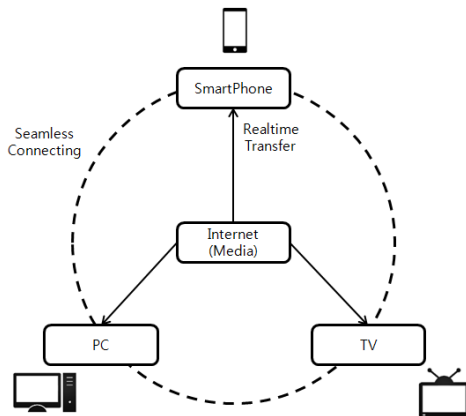


그림 1. 3-Screen의 개념도

그러나 N-screen 서비스는 궁극적으로 TV, PC, 휴대폰 등 다양한 스크린에서 끊임없이 동일한 콘텐츠를 즐기는 것이 가능한 서비스이다. 이는 AT&T사의 3-screen[6] 개념 소개로 시작되었으며 그림 1에서와 같이 TV, 컴퓨터, 휴대폰 등 3가지 유형의 스크린에 콘텐츠를 제공하는 서비스로 네트워크 통합을 통해 콘텐츠의 자유로운 이동과 함께 3-screen을 통해 끊임없이 이용할 수 있는 환경을 말한다.

그림 2는 애플사의 iCloud 플랫폼 중심의 N-screen 서비스의 예를 보여준다[7]. 애플은 단말 별로 콘텐츠를 저장 후 이용하던 하드웨어 종속 모델에서 클라우드 서비스를 통해, 모든 단말을 연결하는 플랫폼 중심 모델로 옮겨 가면서 사용자가 한번 구매한 콘텐츠를 어떤 단말에서도 재생 가능한 소프트웨어 종속 모델로 발전하였다.

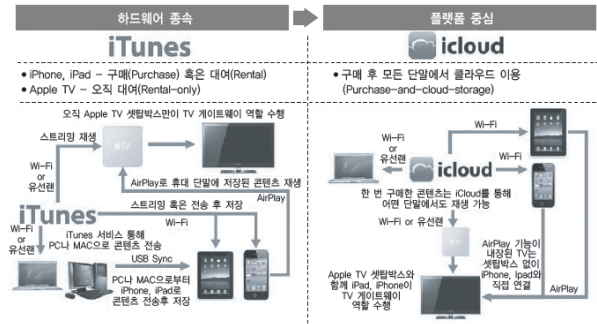


그림 2. N-screen 개념도[7]

2.2 RESTful 웹서비스

REST(Representational State Transfer)는 웹의 장점을 최대한 활용할 수 있는 네트워크 기반의 아키텍처를 제안했는데 이것이 REST(Representational State Transfer)이다. REST는 부수적인 레이어나 세션 관리의 추가 없이 HTTP 프로토콜로 데이터를 전달하는 프레임워크로서 월드 와이드 웹과 같은 분산 하이퍼미디어 시스템을 위한 소프트웨어 아키텍처의 한 형식이다.

또한 클라이언트/서버 간의 구성요소를 엄격히 분리하여 구현은 단순화시키고 확장성과 성능은 높일 수 있는 아키텍처로서 웹에 개방된 리소스들을 원격에서 또는 지역적으로 쉽게 이용하려는 웹 응용에 정착하게 되었고, REST 아키텍처 스타일에 따라 정의되고 이용되는 서비스나 응용을 RESTful 웹서비스라 한다. 여기서 리소스(resource)란 REST 아키텍처의 핵심 요소로서 웹사이트, 블로그, 이미지, 음악, 이용자, 지도, 검색 결과 등 웹에서 다른 이들과 공유하고자 개방된 모든 자원을 의미한다. REST 구조에서의 리소스는 그들의 고유한 URI를 가지며, HTTP의 기본 메소드인 GET/PUT/POST/DELETE만으로 접근할 수 있다[4].

이 RESTful 방식은 별도로 세션을 처리 할 필요 없이 그림 3과 같이 모든 리소스를 URI의 형태로 접근 가능하다는 장점이 존재하며 오픈형 서비스 제공에 적합하다[8,9].



그림 3. RESTful 방식의 Request 및 Response

III. 이어보기 시스템 구성

일반적으로 멀티스크린 서비스를 가능하게 하는 미디어 동기화 방법은 이어보기를 위한 프레임 정보 또는 시간 정보를 공유함으로써 가능하다[10,11]. 해당 정보를 공유하는 방법으로는 네트워크를 통한 전송방법과 클라우드 기반으로 설

계가능하다. 본 논문에서는 하나의 멀티미디어 기기에서 재생 중인 콘텐츠를 또 다른 기기에서도 이어 보기가 가능하도록 하는 동기화 기술을 제안한다. 기존의 시스템과 비교하여 제안하고자 하는 기술은 처음에 재생중인 기기를 포함한 새로운 기기에서도 같은 콘텐츠의 동시 재생이 가능하며, 각 기기에서 다른 플레이어를 사용하여서도 이어보기가 가능하도록 설계하였다[12].

그림 3은 제안 시스템의 기본 이어보기 환경을 요약하여 보여주고 있다. 기본적으로 PC환경과 모바일 환경에서 스마트폰과 태블릿PC에서 이어보기 기능이 다른 기기와 플레이어에서 가능하도록 하였다. 이를 위해서 사용자의 동영상 재생 정보를 저장할 DB 서버와 동영상을 업로드 할 파일서버, 그리고 서버상의 두 정보를 활용하여 이어보기를 지원할 에이전트 애플리케이션 및 플레이어를 구현하였다.

따라서 본 논문에서는 MSSQL을 이용하여 사용자의 동영상 재생 정보를 저장할 DB 서버를 구축하고, Java 기반의 Spring 및 iBatis 프레임워크[13]를 통해 RESTful 환경을 구축하여 손쉽게 커넥션 관리 및 데이터 쿼리를 할 수 있도록 하였다. 마지막으로 윈도우 및 안드로이드 기기를 위한 테스트용 클라이언트 애플리케이션을 구현하여, 이어보기의 성능을 검증할 수 있도록 설계하였다.

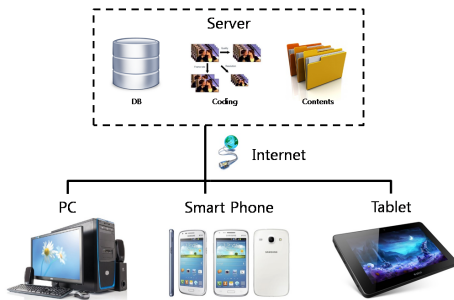


그림 4. 전체 시스템 구성도

그림 5는 본 논문에서 설계된 이어보기 시스템의 소프트웨어 아키텍처를 보인다. RESTful 기반으로 설계하였기 때문에 시스템 외적으로는 Query 문이 존재하지 않는 대신, 인증을 제외한 모든 리소스에 대해 URI로 접근이 가능하다.

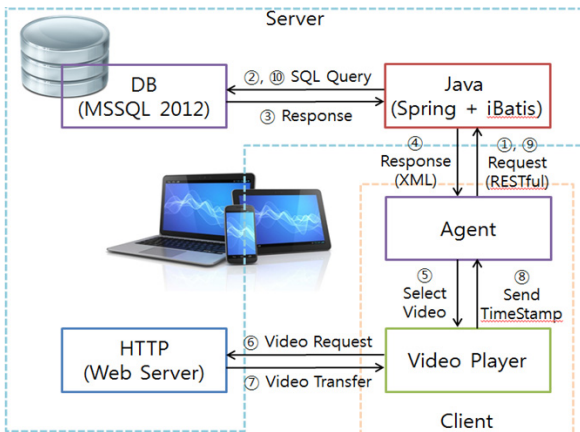


그림 5. 소프트웨어 아키텍처

IV. 시스템 구현

본 논문에서는 사용자의 동영상 재생 정보를 저장할 DB 서버와 동영상을 업로드 할 외부 파일서버로 분리하여 구성하였으나, DB서버와 함께 파일서버를 단일 서버로 구축할 수도 있다. 사용자의 동영상 재생 정보를 저장할 DB에는 사용자의 인증 정보를 저장할 테이블과 동영상의 정보, 각 동영상의 시간 정보를 저장할 테이블로 나누어 설계하였다.

사용자의 정보는 일반적으로 널리 사용되는 회원가입 및 로그인과 유사한 개념으로, 각각의 사용자를 구분하기 위한 아이디와 패스워드, 그리고 해당 아이디의 고유 시퀀스 값을 지닌다. 동영상의 정보를 저장할 테이블은 해당 동영상의 경로(URL)와 동영상 재생 시간 등을 비롯하여 다양한 동영상의 정보와 동영상 소유자 아이디의 고유 시퀀스 값을 저장한다. 이 때, 상기 테이블을 이용하여 이어보하고자 하는 동영상의 리스트를 해당 사용자에게 제공하며, 동영상 선택 시, 이전에 플레이 한 기록이 남아있으면 해당 시점부터 이어볼 수 있도록 시간 정보를 함께 전송한다.

마지막으로 수월한 DB 커넥션의 관리 및 모바일 단말에서의 DB 활용, 그리고 개방형 시스템의 구축을 위해 Java 기반의 Spring, iBatis 프레임워크를 활용하여 RESTful 환경을 구현하였다. 테스트용 클라이언트는 윈도우와 안드로이드 환경에서 각각 구현하였다. 윈도우 클라이언트의 구성은 다음 팟 플레이어, 혹은 자체 플레이어를 활용하여 이어보기가 가능하게 구현하였으며, 발급받은 아이디로 로그인을 하게 되면 서버의 사용자 정보와 대조하여 로그인 성공 여부를 확인한다.

로그인에 성공한 경우, 클라이언트는 서버에 RESTful 방식으로 사용자 소유의 동영상 리스트에 대한 데이터를 요청하며, 서버는 이를 내부 DB에 쿼리 후, 결과를 XML 형식으로 클라이언트에 반환한다. 클라이언트는 상기 XML을 파싱하여 리스트 형태로 사용자에게 표시하고, 사용자는 목록의 동영상 중 하나를 선택하여 재생할 수 있다.

동영상 선택 시 클라이언트는 사용자의 선택에 따라 다음 팟 플레이어 혹은 자체 플레이어를 통해 동영상을 감상할 수 있으며, 사용자가 선택한 동영상의 경로와 이어보기 정보를 서버로부터 전달받아 해당 동영상의 이어보기 정보가 존재할 경우 해당 위치로부터의 이어보기 여부를 사용자에게 질의하여 해당 위치로부터의 동영상 이어보기를 지원한다. 동영상 정지 및 에이전트 종료 시에는 반대로 해당 영상의 재생 정보를 차후 이어보기를 위한 정보로써 서버에 기록한다.

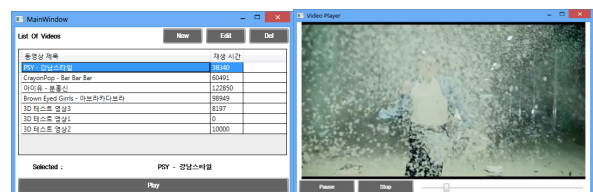


그림 6. 윈도우 기반 클라이언트 구현

안드로이드 기반 클라이언트는 윈도우 기반 클라이언트와는 달리, 다른 애플리케이션의 환경설정 정보에 접근하여 동영상 재생 정보를 수정하기 위해서는 루트 권한이 필요하기 때문에 일반적인 방법으로는 디바이스에 설치된 다른 플레이어에서의 이어보기 기능 구현이 불가능하여, 본 논문에서는 안드로이드 기반 이어보기를 위한 전용 플레이어를 별도로 설계하여 클라이언트에 포함시킴으로 안드로이드에서 기본 제공하는 비디오 뷰를 통해 이어보기를 지원하는 자체 플레이어를 구현하게 되었다. 따라서 본 플레이어를 포함한 안드로이드 클라이언트의 기본적인 동작은 윈도우 기반 클라이언트와 동일하나, 디바이스의 종류 또는 지원 코덱 등에 의해 일정한 사양의 동영상 서비스가 제한될 수 있다.

그림 6과 7은 이어보기 기능이 구동된 재생 실행화면으로 각각 윈도우와 안드로이드 기반으로 실행되었다.

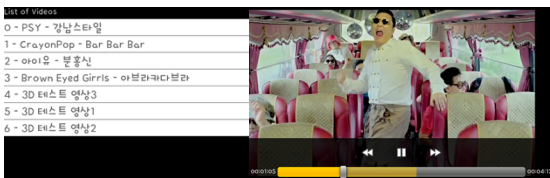


그림 7. 안드로이드 기반 클라이언트 구현

V. 동작검증

본 논문에서의 동작 검증을 위하여 i7-2670 CPU 및 4GB의 RAM이 장착된 컴퓨터와 안드로이드 디바이스의 경우 갤럭시S3를 사용하였다. 그리고 정상 동작 검증을 위하여 동영상 재생까지의 소요 시간을 최대 10초로 설정하고, 각기 다른 용량을 지닌 세 동영상 파일을 대상으로 동영상 선택 후 동영상 이어보기까지 걸리는 소요시간을 동일한 환경에서 각 클라이언트별로 산정하여 측정하였다.

저용량(3.7Mb) 영상의 경우 그림 8과 같이 안드로이드 클라이언트의 경우 평균 1.7초, 윈도우 클라이언트의 경우 0.5초의 시간이 소요되었으며, 중용량(55.55Mb) 영상의 경우 그림 9와 같이 안드로이드 클라이언트의 경우 평균 2.3초, 윈도우 클라이언트의 경우 0.9초의 시간이 소요되었다. 마지막으로(109Mb) 고용량 영상의 경우 그림 10과 같이 안드로이드 클라이언트의 경우 평균 2.7초, 윈도우 클라이언트의 경우 1.2초의 시간이 소요되었다.

실험 결과 유선으로 연결되어 인터넷 연결 상태가 비교적 안정적인 윈도우 클라이언트의 경우 용량에 비례하여 소요되는 시간 또한 소폭 증가하는 것을 확인할 수 있었으나, 무선 와이파이에 연결되어 비교적 인터넷 연결 상태가 불안정한 안드로이드 클라이언트의 경우 AP와 디바이스 간 거리나 인터넷 연결 상태에 따라 시간이 크게 좌우됨을 확인할 수 있었다. 단, 이동통신사가 제공하는 4G망을 이용할 경우에는 대체적으로 윈도우 클라이언트와 유사한 결과를 관측

할 수 있었다.

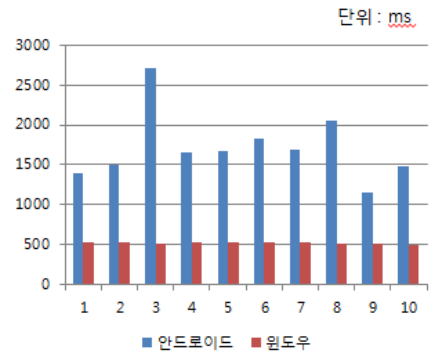


그림 8. 저용량 영상의 재생 소요시간 비교

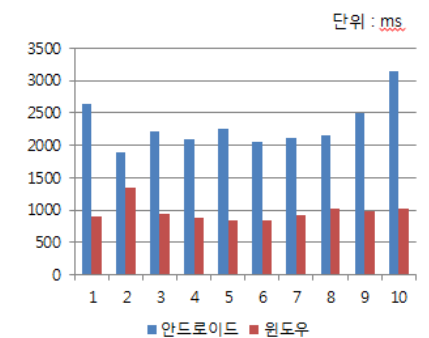


그림 9. 중용량 영상의 재생 소요시간 비교

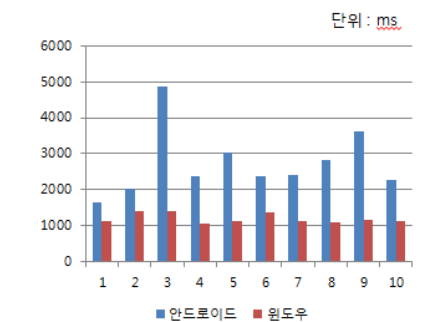


그림 10. 고용량 영상의 재생 소요시간 비교

VI. 결론

본 논문에서는 PC와 안드로이드 환경 간의 이어보기 솔루션을 비교적 단순화하여 간단하게 구성하였으며, 수월한 DB 커넥션의 관리 및 모바일 단말에서의 DB 활용, 그리고 개방형 플랫폼이라는 사전 목표의 달성을 위해 Java 기반의 Spring, iBatis 프레임워크를 활용하여 RESTful 환경을 구성하였다.

이어보기를 위한 서버로 자신의 PC를 활용하였으므로 대규모의 서비스에는 적합하지 않다고 할 수 있으나, 본 논문에서 제시하였던 N-screen 기술을 바탕으로 RESTful 기반의 멀티 디바이스 및 멀티 플레이어 환경에서의 효율적인 이

어보기 기능을 제공하기 위한 서비스의 제안 및 구현이라는 취지를 달성하기에는 큰 무리가 없었다.

또한, 구현 결과의 성능 검증을 위하여 이어보기 지연 시간을 측정하여 실행에 문제가 없음을 보였다. 향후 완성도 높은 플레이어의 구현을 위해 FFmpeg 기반의 플레이어 설계를 목표로 하고 있다.

참 고 문 헌

- [1] Qiang Duan, Yuhong Yan, Vasilakos, A.V, "A Survey on Service-Oriented Network Virtualization Toward Convergence of Networking and Cloud Computing", IEEE Transactions on Network and Service Management, vol. 9 (4), pp. 373-392, 2012.
- [2] 강현승, 김정길, "개인화 N-screen 서비스를 위한 요소 기술", 2011 한국디지털콘텐츠학회 학술발표대회 논문집, Vol.12, No.1, pp. 260-262, 2011. 12. 3.
- [3] 강상욱, 최용수, 김형중, N-screen 요소 기술 및 서비스 동향 분석, 전자공학회지 제38권 제6호, pp. 442-448, 2011. 6.
- [4] 박유미, 문예경, 유현경, 정유철, 김상기, "SOAP 기반 웹서비스와 RESTful 웹서비스 기술 비교", 전자통신동향분석, 제25권 제2호, pp. 112-120, 2010. 4.
- [5] R. Fielding, "Architectural Styles and the design of Network-based Architectures", PhD Thesis, University of California at Irvine, 2000.
- [6] N. Narasimhan, Doo Myungcheol, J. Wodka, V. Vasudevan, "CollecTV intelligence: A 3-screen 'social search' system for TV and video queries", Collaborative Computing: Networking, Applications and Worksharing, 2010 6th International Conference on, pp. 1-6, 2010.
- [7] 유은재, "N스크린(N-Screen)", NET TERM, 한국인터넷진흥원, 2012.
- [8] 한창환, 길준민, 한연희, "REST 웹 서비스를 이용한 데이터베이스 정보 획득", 한국정보기술학회논문지, 제6권, 제4호, pp.196-206, 2008.8.
- [9] 김상일, 김화성, "REST 프로토콜 기반의 API 선별 기법 및 Open API 자동 합성 방안", 한국통신학회논문지, Vol.38C, No.07, 2013.
- [10] 강미란, 김세영, 김영일, 김대진, "다중 기기에서 이어 보기를 위한 동기화 기법", 한국콘텐츠학회논문지 제11권 제12호, pp. 543-551, 2011.12.
- [11] 김성원, 정문열, "VOD 서비스에서 이종매체간 연결재생을 위한 세션 동기화 시스템의 설계 및 서비스 시뮬레이션", 한국방송공학회지, 제14권, 제1호, pp.15-27, 2009.
- [12] 윤형덕, 백봉진, 김정길, 김영섭, 박제호, 윤경로, "이종 멀티미디어 기기 기반 동영상 이어 보기 연구", 2013년도 정보통신설비 학술대회, pp.66-68, 2013.08.
- [13] 이일민, 토비의 스프링 3, 에이콘, 2011.

저자

김 정 길(Cheong Ghil Kim)

중신회원



- 1987년 8월 : Univ. of Redlands, USA 컴퓨터과학과 학사졸업
- 2003년 8월 : 연세대학교 컴퓨터과학과 공학석사 졸업
- 2006년 8월 : 연세대학교 컴퓨터과학과 공학박사 졸업

- 2006년~2007년 : 연세대학교 컴퓨터과학과 박사후 연구원
 - 2007년~2008년 : 연세대학교 컴퓨터과학과 연구교수
 - 2008년~현재 : 남서울대학교 컴퓨터학과교수
- <관심분야> : 멀티미디어 임베디드 시스템, 이종 컴퓨팅, 모바일 AR, 3D Contents