텔레매틱스 서비스를 위한 단말기 및 서버 기술동향 - GST 시스템을 중심으로

임베디드포코리아 장승운
서강대학교 정성원*

1. 서 론

텔레매틱스는 이미 여러 매체를 통하여 알려진바 데이터를 다루는 IT기술과 정보를 전달하는 통신기술의 통합으로 우선 적용될 환경으로 자동차를 꼽았다. 이러한 텔레매틱스를 구현할 수 있는 많은 기술이 논의되었고 또 적용되고 있으나 실제 서비스를 구현할 수 있는 기술 논의가 국내에서 활발하지 못하였다. 여기에서는 텔레매틱스 서비스가 구현되기 위한 핵심 영역인 자동차 내의 서비스 구현하는 단말기와 서버기술에 대해 알아본다.

텔레매틱스 서비스 모델은 단독형인 1세대, 개별기업의 임의의 고 정형 서비스인 2세대, 다중 단말기 벤더와 다중 서비스 제공업체가 개방형 환경으로 운영되는 3세대로 발전하고 있다.

여기에서는 3세대 서비스 모델로 사실상 표준(de facto)인 3GT(The Third Generation Telematics) 참조모델에 따라 단말기 아키텍쳐, 차량 내 네트워크의 표준인 AMI-C(Automotive Multimedia Interface Collaboration), 텔레매틱스 서비스 제공(TSP: Telematics Service Provider) 아키텍쳐 그리고 단말기와 TSP 간을 연결하는 텔레매틱스 프로토콜(GTP: Global Telematics Protocol)을 중심으로 접근한다.

텔레매틱스 서비스를 제공하기위한 구성은 텔레매틱스 서비스의 인터랙티브 환경을 사용자에게 제공하는 단말기, 차량 내 정보를 연계하는 차량 네트워크, 단말기와 센터를 연결하는 이동통신망, 그리고 자동차용 정보수집 및 배포를 담당하는 TSP로 구성된다.

이와 같이 텔레매틱스 서비스의 가치사슬에 있어 많은 이해관계자로 구성되므로 실제 구현의 어려움이 극복하고 시장개척을 위한 개방형 시스템(OS: Open System) 기반의 구체적인 비즈니스 모델인 GST(Global System for Telematics)를 유럽에서 제안하였다.

2. GST 서비스 시스템

GST는 유럽에서 단일 시스템, 단일 약정, 단일 인보이스와 같은 혁신적 텔레매틱스 서비스를 위한 개방형 환경을 제공한다. 미래 자동차는 다양한 통신장비를 구비할 것 이고, 이러한 통신 시스템은 공통 아키텍쳐와 표준 인터페이스 기반으로 상호운용 환경을 제공할 것 이다. 운전자 내지는 소유주는 온보드 형태의 통합 텔레매틱스 시스템에 의존하고, 유럽 내 어디라도 온라인상으로 안전(Safety), 효율성, 편리한 확장서비스에 접속할 목적으로 도입하여 진행 중 이다.

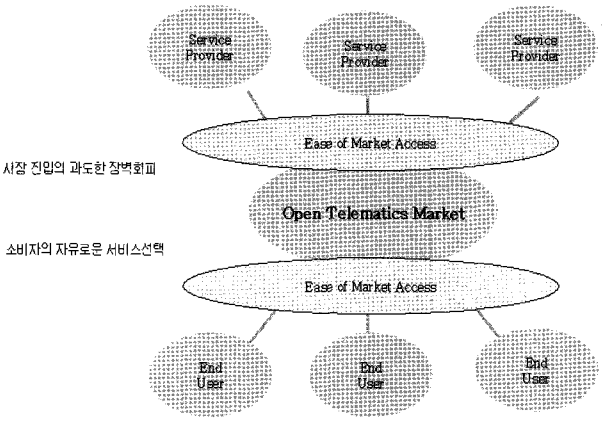


그림 1 텔레매틱스 시스템과 개방형 마켓

GST 프레임워크의 구성과 설계는 GST 구조뿐만 아니라 표준통신 프로토콜 기반 하에 엔드-투-엔드 텔레매틱스 솔루션에 공통적이며 특정 요소에 영향을 끼치지 않는 구조이다. GST의 개방형 시스템은 현재 및 미래 텔레매틱스 솔루션 구축을 위하여 프로토콜 정의, 서비스 배치 및 프로비저닝, 클라이언트 시스템 실행환경, 실행환경상의 보안과 지불연동절차, 그리고 텔레매틱스 가치사슬에서 디바이스 통합에 대하여 기술하고 정의한다.

* 종신회원

3. 텔레매틱스 단말기

3.1 텔레매틱스 단말기 요구

텔레매틱스 단말 플랫폼에서도 범용 임베디드 시스템에서 요구되는 상호 연동성, 호환성, 유연성, 확장성 등의 특성이 요구된다. 그러나, 텔레매틱스에서는 일반적인 임베디드 시스템과의 차별성이 요구되는 독특한 특성이 있다. 이는 차량과 IT 기술 사이의 생명주기 차이, 제공 가능한 서비스의 다양성 및 발전성, 안전이 극히 요구되는 차량의 특성 등에서 기인된다고 볼 수 있다. 일부 서비스 이미 제공되거나 혹은 가까운 미래에 실현될 것으로 보이는 다양한 고급 정보와 멀티미디어 기반 엔터테인먼트, 편리하면서도 안전한 운전 등의 텔레매틱스 서비스는 그 구체적 종류와 기술이 아주 다양하다. 또, 새로운 기술을 바탕으로 하는 새 서비스가 생길 수 있을 것이다. 텔레매틱스 단말은 새로운 서비스를 적은 비용으로 쉽게 추가 혹은 제거할 수 있는 구조이어야 한다.

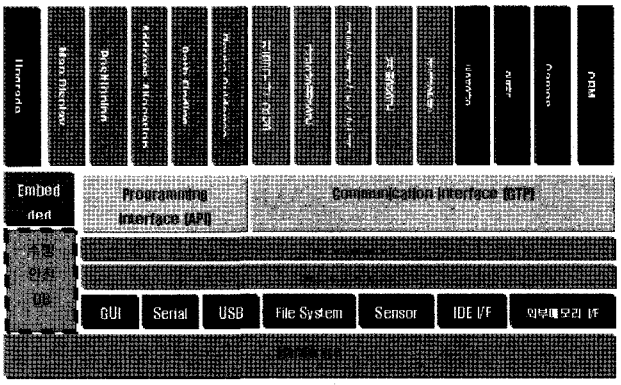


그림 2 모듈화된 서비스구조(출처: ㈜The Sign)

결과적으로 텔레매틱스 단말은 모듈화 한 구조를 가져야 할 것이다. 여기서의 모듈은 소프트웨어에 국한되지 않는다. 스케일러블, 유연성, 신뢰성 그리고 가장 중요한 적은 메모리 사용도 포함된다. 폭 넓은 어플리케이션 배치하기를 원하는 기업은 선택의 어려움에 직면하나, 통일된 서비스 플랫폼은 어플리케이션의 새로운 번영을 낳게 될 것으로 기대한다.

3.2 텔레매틱스 단말기 소프트웨어

3.2.1 .NET 컴팩트 프레임워크

.NET 컴팩트 프레임워크는 디바이스의 네이티브(특정 컴퓨터로만 사용하도록 고안된) 운영체제를 사용한 어플리케이션 접속 제공하고 서버단과의 통합은 네이티브 운영체제 서비스를 제공하고 선택적으로 네이티브 API 호출을 제공한다.

관리와 네이티브 어플리케이션을 동시에 수행 가능하고, 네이티브 어플리케이션 자체 도메인 호스트는 관리된 코드를 실행하기 위한 공통 언어 실행시간의 인스턴스를 시작한다.

OSGi와 기본개념은 동일하고 차이점으로 구현언어는 C# 기반으로 마이크로소프트사의 운영체제 상에서만 실행환경을 구축할 수 있다.

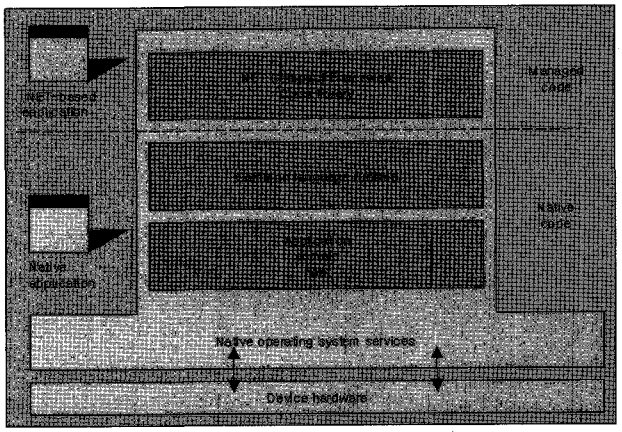


그림 3 .NET 컴팩트 프레임워크의 시스템 구성도
(출처: MS의 .NET Compact Framework)

3.2.2 OSGi 서비스 플랫폼

OSGi(Open Service Gateway initiative) 프레임워크는 어플리케이션 개발 동안 일관된 프로그래밍 모델의 제공한다, 이는 개발과 구현으로부터 서비스의 스펙을 분리에 의한 서비스 사용을 제공하므로 개발자는 동일한 서비스 인터페이스에 다중 구현을 지원 가능하고, 서비스를 사용하는 개발자는 구현과 무관하게 서비스의 인터페이스를 작성 가능하도록 한다.

프레임워크가 다양한 디바이스에서 실행되도록 설계했기 때문에 디바이스의 다른 하드웨어 특성은 서비스 구현에 많은 영향을 미치지만, 안정적인 서비스 인터페이스는 디바이스 상에서 실행되는 전체 소프트웨어 시스템의 안정성을 보장한다, 예를 들어, 고성능 디바이스에서, 로그인 서비스는 하드드라이버에서 로그 메시지를 저장할 수 있지만, 디스크 없는 디바이스에서, 로그 엔트리는 원격저장기능 지원하여. 두 가지 로그인 서비스 구현 개발자는 동일한 인터페이스를 보유한다. 로그인 서비스를 사용하는 서비스의 개발자는 사용될 서비스의 구현 고민 없이 로그인 서비스 인터페이스에 대한 코드 작성을 제공한다.

어플리케이션 개발자 지원으로는 어플리케이션을 적은 자체-인스톨 가능한 컴포넌트로 분할하여 수명-주기 관리 기능을 지원하고 이들 컴포넌트를 번들이라 부른다. 번들은 요구에 따라 다운로드할 수 있고 필요 없으면 제거, 갱신 등의 기능이 제공된다.

서비스 플랫폼으로써 특징은 OSGi는 생산업체, 서

비스 제공업체, 개발자를 위한 표준 소프트웨어 컴포넌트 프레임워크로 OSGi 스펙이 개방형 표준의 요소는 모든 참여자의 공정한 역할뿐만 아니라 어플리케이션의 생명주기를 제어하는 유연한 배치 API 어플리케이션은 표준화 배치 형식에 인스톨되고 JVM의 재시작 없이 시작, 종료, 갱신, 그리고 언 인스톨 기능을 제공한다.

또한 제한된 공간에서 어플리케이션을 실행하는 안전한 환경으로 어플리케이션이 환경에 이상 없고, 다른 상주 어플리케이션에 간섭이나 영향을 끼치지 않고 신뢰성 적은 어플리케이션이 전체 시스템 통합 절충안 없이 JVM 내의 실행을 허용하므로 전체 시스템의 안정성을 확보할 수 있다.

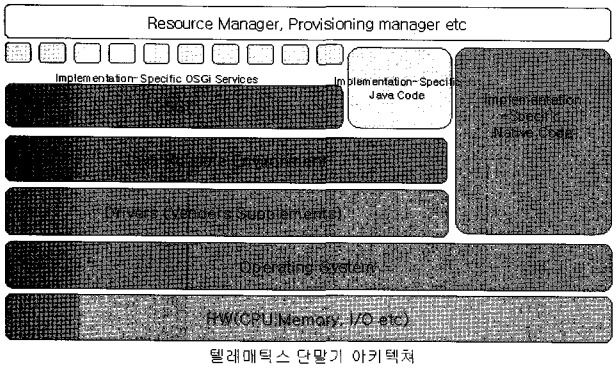


그림 4 OSGi 기반 텔레매틱스 단말기 구조

OSGi 서비스 플랫폼구조는 시스템 운영자나 이를 관리하는 기업의 관점에서 단일 관리 영역으로부터 수백만의 서비스 플랫폼의 관리를 허용하는 유연한 원격 관리 구조로 OSGi 서비스 운영자에게 섬세한 제어를 제공한다.

3.2.3 OSGi 서비스 플랫폼의 운영방법

OSGi는 네트워크 서비스를 위한 표준화, 컴포넌트 지향, 컴퓨팅 환경을 정의하므로, OSGi 서비스 플랫폼을 네트워크 디바이스에 추가하면, 네트워크 어떤 곳으로부터 디바이스내의 소프트웨어 컴포넌트의 라이프사이클을 관리 추가할 수 있다. 따라서, 소프트웨어 컴포넌트는 인스톨되고, 갱신되고, 디바이스의 운영에 방해 없이 사용 중 제거 기능 제공함 가능하고 소프트웨어 컴포넌트는 동적으로 다른 컴포넌트를 발견하고 사용할 수 있는 라이브러리 또는 어플리케이션임.

3.2.4 OSGi 서비스 플랫폼과 BMW 시리즈 5

BMW의ConnectedDrive는 자동차의 미래는 정보와 서비스 네트워크내의 통합 요소가 증가하므로, 고객 성향에 따른 개별화, 운전자 중심 형태의 필요하고 원하는 보다 많은 정보를 운전자에게 제공하여 자동차 외부세계와 도로 교통 처리 시스템을 연결하는 혁신적인 개념을 포함하여 텔레매틱스 서비스, 온라인 서비스 그리고 운전자 보조 시스템에서 추출한 정보를 제공한다.

BMW는 시리즈 5의 초기에 고품격 인포테인먼트(Infotainment) 플랫폼의 기반으로 사용하였고, BMW 관심을 끈 핵심 특징은 기능성의 집중화와 쉬운 방법으로 기능 향상이 옵션으로 자동차에서 제어장치 개수를 축소하여 복잡성과 비용을 줄임. 다른 생산라인을 통한 다른 시장에 다른 컨피규레이션의 관리가 용이성을 향상하였다. BMW와 같은 자동차 제조회사는 임의의 기술을 사용하지 않고 표준화와 이에 따른 다수의 기술 제공업체가 경쟁력을 부여하는 생명과 같다.

4. GST 구현 가이드

GST(Global System for Telematics)의 구현에 관한 개괄적 시각은 서비스 제공업체에 의한 컨트롤센터상의 서비스 어플리케이션의 배치, 고객 맞춤형 모바일 플랫폼인 클라이언트 시스템에 서비스 어플리케이션의 프로비저닝, 그리고 클라이언트 시스템에 서비스 어플리케이션의 인스톨 및 실행과 같은 특징을 제공한다.

개방형 시스템은 벤더가 구현해야 하는 GST 코어 스펙에 대한 GST 호환 인증, GST 스펙을 자바 OSGi 플랫폼에 바인딩, 그리고 바인딩을 위한 참조 구현하는 세 영역으로 구성된다. 개방형 시스템은 서비스 제공업체가 제공하는 서비스가 최종사용자에 공급되는 다양한 비즈니스모델을 연구한다.

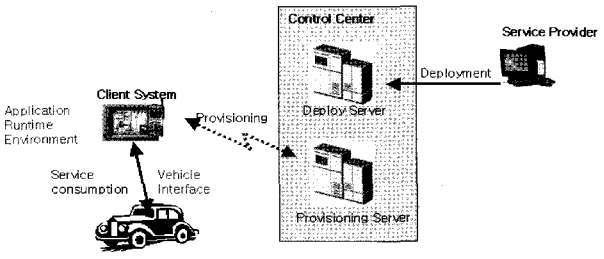


그림 5 GST의 개방형 시스템 구성도

개방형 시스템은 서비스객체와 서비스 제공업체로부터 고객으로 이동경로 간의 관계를 나타낸다. 개방형 시스템은 서비스 제공업체에게 컨트롤센터의 배치부에 표준 기반의 서비스 어플리케이션의 패키지와 배치를 제공한다. 개방형 시스템의 보안과 페이먼트 서비스는 고객에게 서비스를 주문과 가입를 제공하고, 필요한 소프트웨어 컴포넌트를 클라이언트 시스템에 다운로드하고 소프트웨어 플랫폼 상에 패키지 어플리케이션을 구동한다.

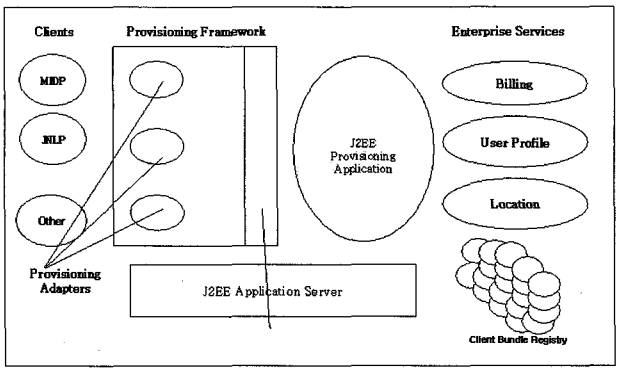


그림 6 프로비저닝 시스템의 구성도

프로비저닝 클라이언트는 프로비저닝 시스템의 클라이언트측 초석이고 SyncML 프로비저닝 클라이언트의 자바 버전 구현이다. 컴퍼넌트는 OSGi 번들로 제공되고 OSGi R3 호환 프레임워크 상에서 실행된다.

서비스 어플리케이션의 은 GPS 안테나로부터 위치 정보를 수집하며 실제 위치데이터를 위치 인터페이스에 의해 표준화하여 인터페이스하고 개방형 시스템은 서비스 어플리케이션 코드간의 호환과 결과적으로 서비스 어플리케이션 개발 노력을 줄이도록 촉진한다. 본질적으로 서비스 어플리케이션 개발자는 위치 인터페이스와 플랫폼상의 실제 위치 데이터를 링크만 하므로 완료된다.

클라이언트 시스템에서 실행되는 어플리케이션은 운용중인 원격 서버와 통신한다. 개방형 시스템은 플러그 연결 관리자 인터페이스를 제공하여 통신의 용이함이 서비스 어플리케이션에 필요하다. 참조 구현은 이런 컴퍼넌트의 자바/OSGi 구현을 제공하고 아파치 AXIS 오픈소스 SOAP(Simple Object Access Protocol) 구현 기반의 SOAP 연결 어댑터를 포함한다.

서비스 어플리케이션의 배치는 XML 기반 배치 디스크립터와 패키지 형식 스펙을 따르고, 배치 서버는 웹 어플리케이션으로써 구현된다. 웹 어플리케이션 서버는 Jboss 어플리케이션 서버상에서 구동하고 배치 데이터의 유지를 위하여 하이버네이트(Hibernate)를 사용한다. 웹 어플리케이션은 WAR, HAR와 EAR 파일을 사용한다. 패키지 서비스 어플리케이션을 적절히 배치 서버로 배치한다.

컨트롤 센터에 배치되면, 사용자는 특정 서비스가 프로비저닝 준비가 되면 가입할 수 있다. 프로비저닝은 개방형 시스템이 어플리케이션 생명주기와 위치/차량 인터페이스와 같은 전형적 클라이언트측 컴퍼넌트용 인터페이스를 정의한 클라이언트상의 프로비저닝 프로토콜의 스펙이 제공된다. 프로비저닝 서버는 J2EE 웹 어플리케이션 서버상에 구현되고 배치 서버와 함께 패키지화 한다. 프로비저닝 서버는 SyncML OMA DM 프로토콜을 지원하고 개방형 시스템상에서 Sync4J 1.4.8 기반이고 use Postgre 데이터베이스 시스템을 사용한다.

참고문헌

[1] OSGi Alliance, OSGI Specification Service Platform R3, 2003.

[2] Per Gustafson, Sofia Doncheva, Hans Ulrich Michel, Florian Wildschuette, Jurgen Wojatchek, volker Vieroth, Erwin Vermassen, DEL_GST_OS_3_1_Architecture_and_interface_specifications, GST Forum. Sept. 2005.

[3] GST Open Systems Consortium, SP Manager Erwin Vermassen, DEL_GST_OS_DEV_Reference_Implementation_3_2, GST Forum. Jan. 2006.

[4] GST Forum, DOC_GST_OS_DEV_7_PG_Control_Centre Implementation Guide, 2006.

[5] PM Peter Van der Perre, 2_Deli_20031208_D3.2Finalspecifications_v1.0, Ertico. Dec. 2003.

[6] 이형석, "텔레매틱스 단말기 플랫폼 기술", pp.85~91, TTA 저널 제89호. Dec. 2003.

[7] 송종화 외 공저, 텔레매틱스 개론, 홍릉과학출판사. 2004.

부 록

○ OSGi 프레임워크 구성 및 역할

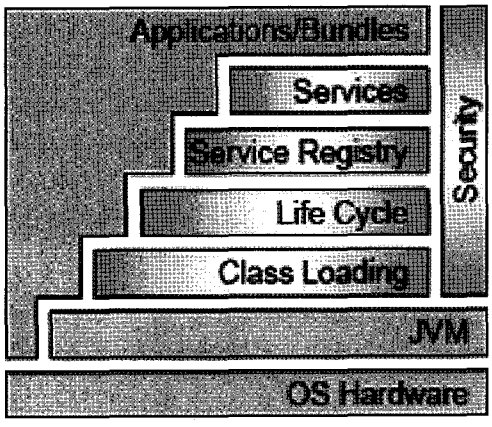


그림 7 OSGi 구조(출처: OSGi 웹 사이트)

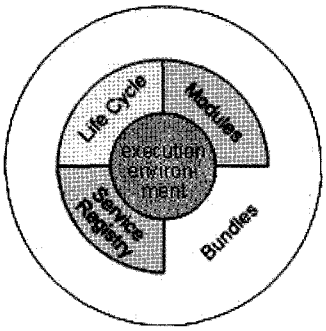


그림 8 OSGi 계층별 프레임워크
(출처: OSGi 웹 사이트)

■ **프레임워크**

- OSGi 스펙의 핵심 컴포넌트인 OSGi 프레임워크로 표준화 환경을 어플리케이션에 지원하고 4 계층으로 구성된다.

L0: 실행환경

자바 환경의 스펙으로 자바 2 컨피규레이션과 프로파일은 모든 타당한 실행 환경이다. OSGi는 FP(Foundation Profile) 기반 실행 환경과 OSGi 번들에 유용한 실행 환경에 최소한 요구를 특징짓는 보다 작은 버전에서 표준화한다.

L1: 모듈

모듈은 클래스 로딩 정책을 정의함. OSGi 프레임워크는 강력하고 엄격한 클래스 로딩 모델임. 이는 자바 상위에 기반 하지만 모듈화로 추가한다. 자바에서, 일반적인 모든 클래스와 자원을 포함하는 단일 클래스패스가 있음. OSGi 모듈 계층은 모듈 자체 뿐만 아니라 모듈간의 제어 링크를 위한 개별 클래스를 추가한다.

L2: 라이프사이클 관리

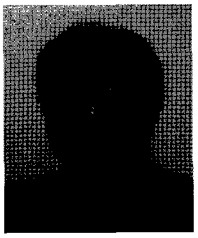
라이프 사이클 계층은 동적으로 인스톨, 스타트, 스톱, 갱신과 언인스톨 할 수 있는 번들 추가 가능하다. 번들은 클래스 로딩을 위한 모듈 계층에 의존하지만 모듈을 관리하기 위한 API를 실행 시간 추가를 제공한다. 라이프 사이클 계층은 어플리케이션에서 일반적이지 않은 동적으로 시작하고, 확장 종속성 메커니즘은 정확한 운용 환경을 보장한다.

L3: 서비스 등록

서비스 등록 계층은 다이나믹스(the dynamics)를 고려하여 번들을 위한 협력 모델을 제공함. 번들은 전통적인 클래스 공유를 통하여 협력하지만 클래스 공유는 동적 인스톨과 언인스톨 코드에 호환하지 않음. 서비스 등록은 포괄적인 모델의 번들간의 객체 공유를 제공한다.

다수의 이벤트는 서비스의 왕래를 조작하기 위한 정의함. 서비스는 무엇이든 표시할 수 있는 단지 자바 객체로 많은 서비스는 HTTP 서버나 객체와 같은 서버이고, 다른 서비스는 실세계에서 객체로 표기되고, 블루투스 폰이 근접한 실례라 할 수 있다.

장 승 운



1991 강원대학교 전자계산학과(학사)
1995 강원대학교 전자계산학과(석사)
2001 런던대학교 분산시스템 전공 박사수료
1996~1999 글라스고우대학교 멀티컴퓨터분야 연구스태프
1996~2003 기술 컨설턴트 휴대용 디바이스부문 ㈜CI 정보기술
2003~현재 차세대 성장동력 차세대이동통신 텔레매틱스분야 기획위원
2004~현재 기술표준원 산업표준심의회 정보산업부회 차세대이동통신 전문위원
2004~현재 산업자원부 차세대 성장동력 차세대이동통신 텔레매틱스분야 국제협력위원
2002~현재 (주)임베디드포코리아 대표이사
관심분야 : 분산시스템, 텔레매틱스, 미들웨어
E-mail : chang.seungwoon@gmail.com

정 성 원

1988 서강대학교 전자계산학(학사)
1990 M.S. in Computer Science at Michigan State Univ.
1995 Ph.D. in Computer Science at Michigan State Univ.
1997~2000 한국전산원 선임연구원
2000~서강대학교 컴퓨터학과 부교수
관심분야 : Mobile Computing Systems, Mobile Databases, Telematics, Spatial DB, Mobile Agents, Streaming Data Processing in Ubiquitous Computing Environments, Distributed Databases
E-mail : jungsung@sogang.ac.kr