

이동단말용 미들웨어 테스트 프레임워크

신석규* · 이상수** · 양해솔***

Middleware Test Framework for Mobile Devices

Seok Kyo Shin* · Sang Soo Lee** · Hae Sool Yang***

■ Abstract ■

Middleware for mobile devices, embedded middleware, is used for mobile devices, cellular phone and DMB related services. It is essential for middleware providers to provide high quality middleware in conformity with its standard and interoperability on hardware platforms. The standard conformance and interoperability of embedded middleware are the key factors to make service providers succeed in their business. The quality of embedded middleware could be secured when Tests for the standard conformance and interoperability of embedded middleware are performed with well established test framework. In this paper, we describe middleware testing methodology of standard conformance, interoperability and middleware test framework for mobile devices.

Keyword : DMB, Standard Conformance, Interoperability

1. 서 론

이동단말기는 임베디드 기기의 특성상 하드웨어와 운영체제가 다양하다. 그러므로 이동단말기용 미들웨어는 다양한 하드웨어와 시스템 소프트웨어에 구애받지 않으며, 미들웨어 표준을 준수하

는 공통 프레임워크를 제공해야 한다. 이를 위해서는 미들웨어가 표준에 적합하도록 설계되었는지를 확인하는 표준적합성과 다양한 이동단말기 및 운영체제에서 작동하는지를 확인하는 상호운용성에 대한 시험을 통해 소프트웨어의 품질을 확보할 필요가 있다.

* 한국정보통신기술협회 SW시험인증센터 센터장

** 한국정보통신기술협회 SW시험인증센터 전임연구원

*** 호서대학교 벤처전문대학원 교수

이동단말용 미들웨어는 이동단말기 하드웨어와 여기서 구동되는 운영체제인 소프트웨어가 다양하므로 미들웨어의 개발 방법 및 테스트 방법이 다양하다. 따라서 모든 이동단말용 개발 시스템 환경에서 일반적으로 사용할 수 있는 정형화된 이동단말용 미들웨어 테스트 프레임워크가 필요하다.

본 논문에서는 미들웨어의 표준적합성 및 시스템소프트웨어 상호운용성의 효과적 테스트 방법과 이를 보편화한 이동단말용 미들웨어 테스트 프레임워크(MTFM; Middleware Test Framework for Mobile devices)를 제안한다.

본론에서 미들웨어 결함모델, 상호운용성 및 표준적합성 테스트 방법과 이동단말용 미들웨어 테스트 방법을 제시하고, 이어서 이를 보편화한 이동단말용 미들웨어 테스트 프레임 워크를 제안한다.

2. 본론

2.1 이동단말용 미들웨어

이동단말용 미들웨어는 위성 및 지상과 DMB 등의 서비스를 휴대폰 및 이동단말기에서 제공하기 위해 사용된다. 이동단말용 미들웨어는 해당기기의 주 기능 동작에 영향을 미치지 않고 안정적인 상태로 동작해야한다. 이를 위해 미들웨어는 표준에 적합한지, 다양한 운영체제에서 동작 가능 여부를 확인하는 테스트를 통해 일정수준의 품질이 보장되어야한다.

2.2 미들웨어 시험방법

이동단말용 미들웨어를 테스트하는 방법은 미들웨어가 제공하는 API 자체를 화이트박스 기법으로 테스트하는 정적분석 방법과 API를 사용하여 구현한 기능을 실시간으로 실행하여 테스트하는 동적분석 방법이 있다.

정적분석 방법은 API의 입력파라미터를 변경하

여 출력의 정상 유무를 확인하는 방법으로 해당 API의 무결성을 검증하기 위한 방법이다. 이 방법을 통해 API 각각에 대한 정상동작 유무를 확인할 수 있으나, API 간의 상호작용에 따른 기능의 정상유무는 확인할 수 없다. 동적분석 방법은 표준 또는 기술문서에서 제안하는 기능을 여러 API를 사용하여 구현한 후, 실시간으로 실행하여 기능의 정상 동작여부를 확인하는 시험방법이다. 이를 사용하면 API 간의 상호작용의 정상동작 여부 및 미들웨어 표준 준수여부를 확인할 수 있다. 미들웨어의 정상동작 여부는 정적분석방법과 동적분석 방법을 병행해서 표준적합성 및 상호운용성 테스트를 통해 확인할 수 있다.

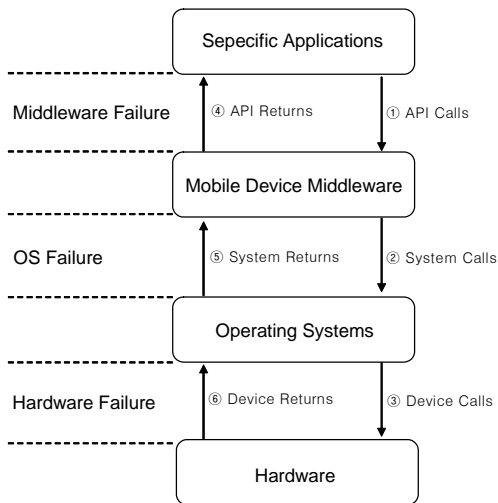
2.3 이동단말용 미들웨어 결함모델

이동단말용 미들웨어는 시스템 소프트웨어와 응용 소프트웨어의 중간에서 특정 응용분야에 최적화된 공통 프레임워크이다. 일반적으로, 미들웨어는 [그림 1]에서와 같이 응용프로그램과 운영체제 중간에 위치하고 있다. 응용프로그램, 미들웨어, 운영체제 및 하드웨어 계층은 각각 인터페이스 및 API함수를 통해 상호호출 될 수 있다. 본 논문에서는 각 계층 간의 상호호출 시 발생하는 결함을 미들웨어 결함, 운영체제 결함, 하드웨어 결함으로 정의한다. 응용프로그램으로부터 ① API호출을 받은 미들웨어는 ④ API 리턴을 통해 결과를 되돌리는 상호호출 시 발생하는 결함을 미들웨어 결함으로 정의하고, 미들웨어가 운영체제에 ② 시스템 콜을 보내면 ⑤ 시스템 리턴을 되돌리는 상호호출 시 발생할 수 있는 결함을 운영체제 결함이라고 하며, 운영체제가 상호호출(③, ⑥)을 하는 경우 발생하는 결함을 하드웨어 결함으로 정의한다.

일반적으로, 데스크톱용 미들웨어는 단일 운영체제를 사용하여 개발하므로 ② 시스템 콜 및 ⑤ 시스템 리턴의 상호호출에 관한 처리가 표준화 되어 있기 때문에 미들웨어 결함만을 다루면 된다.

그러나 임베디드용 미들웨어의 경우, 설계자는 여러 가지 운영체제에서도 동작할 수 있도록 개발하여야하므로 다양한 운영체제가 제공하는 시스템 콜 및 리턴 등의 상호호출에 대해서도 상호운용이 가능한지를 고려해야 한다. 이러한 설계단계의 복잡성으로 인해, 동일 미들웨어를 여러 단말기에 설치할 경우 일부 이동단말기는 정상적으로 동작하지 않는 경우가 발생하게 된다. 물론 이러한 문제를 해결하기 위해 하드웨어 종류에 구애받지 않도록 하기위해 이동용단말기 미들웨어를 자바로 개발할 수도 있으나, 이동용단말기 하드웨어 자원의 한계 등으로 인해 현실적으로는 어려운 실정이다.

이동단말용 미들웨어 테스트는 데스크톱용이 미들웨어 결합만 고려하는 것과 다르게 미들웨어 결합 뿐 만아니라 운영체제 결합도 함께 고려되어야 한다. 이동단말용 미들웨어 결합은 경우에 따라 운영체제 결합과 연관성이 있는 경우가 많다. 따라서 미들웨어의 상호운용성을 보장하기 위해서는 운영체제 결합 시험이 필요하다. 운영체제 결합시험을 위한 상호 운용성 테스트는 하나의 테스트 케이스에 대해 운영체제 개수만큼 반복적이고 기계적으로 이루어진다.

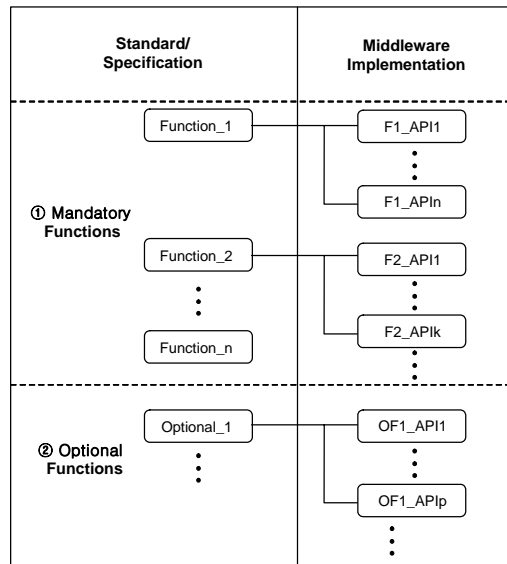


[그림 1] 이동단말용 미들웨어 결합 모델

2.4 표준적합성 테스트

일반적으로, 미들웨어는 미들웨어 표준 및 기술 문서를 기반으로 개발된다. [그림 2]에서와 같이, 미들웨어 표준이 ① 필수적인 기능과 ② 부가적인 기능으로 구성되어 있으면 개발자는 여러 개의 API 함수로 해당 기능을 구현하게 된다. 즉, 표준에서 요구하는 기능은 여러 개의 API가 조합으로 구현될 수 있기 때문에 미들웨어 표준적합성 테스트는 단일 API의 정상 동작을 확인하는 유닛 테스트와 API의 유기적 조합으로 구현된 기능의 정상 동작 여부 및 표준 적합성 여부를 살피는 기능 테스트가 병행되어야한다. 단일 API함수의 정상 동작여부 확인만으로는 미들웨어 표준이 제안하는 기능의 정상동작여부는 알 수 없으므로 기능 테스트는 표준 준수 여부 확인을 위해 필수적으로 수행되어야한다.

유닛 테스트 및 기능 테스트를 위해 활용할 수 있는 문서로 API함수 사용 설명서 및 기능 구현 설명서가 있다. 유닛테스트는 API 사용 설명서를, 기능 테스트는 기능 구현 설명서를 참조하여 테스트할 수 있다. [그림 2]에서는 유닛 테스트와 기능

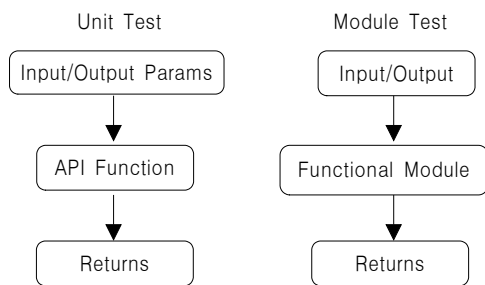


[그림 2] 이동단말용 미들웨어 표준 및 구현

테스트 모델을 보여주고 있다. 유닛 테스트는 API의 요구사항인 입, 출력 파라미터에 대해 함수의 리턴값을 검증하는 것이며 기능 테스트는 미들웨어를 사용하여 구현한 기능의 입, 출력 및 리턴값을 실시간으로 실행하여 검증하는 것이다. 기능 테스트는 미들웨어 표준에 명시되어 있는 기능에 대한 검증이므로 표준적합성 테스트에 해당한다.

미들웨어의 표준적합성 테스트는 앞에서 밝힌 바와 같이 유닛 테스트 및 기능모듈 테스트가 같이 수행되어야 의미가 있다. 미들웨어 표준에서 제안하는 기능은 API함수의 유기적인 결합으로 구현되기 때문이다.

앞서 제안한 것과 같이 유닛 테스트 및 기능 테스트는 정적분석 및 동적 분석방법을 사용한다. 주로 정적분석 방법은 유닛 테스트 시 화이트 박스 테스트를 사용하며 동적분석 방법은 기능 테스트에서 블랙박스 테스트를 사용하게 된다. 두 가지 시험을 수행하는데 있어서 주의할 점은 유닛 테스트와 기능 테스트 두 가지 테스트 중 하나를 빠뜨리고서는 표준적합성 만족 여부를 확인할 수 없다는 점이다. 왜냐하면 각 API함수가 정적분석을 통해 무결성이 증명되더라도 이를 조합하여 구현한 기능이 실시간으로 실행될 때 결합이 없다고 할 수 없으며, 표준에 명시된 기능이 정상동작하더라도 그 기능을 구현한 각 API에 결합이 없다고 할 수 없기 때문이다. 즉, 표준적합성 테스트는 유닛 테스트와 기능 테스트를 병행해서 이루어져야 한다.



[그림 3] 유닛 테스트 및 기능모듈 테스트 모델

2.5 시스템 소프트웨어 상호 운용성 테스트

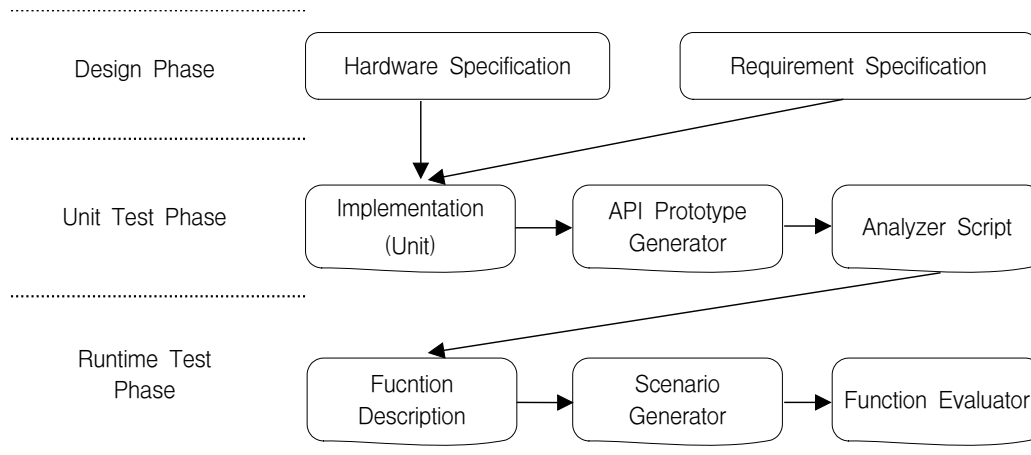
[그림 3]에서와 같이 이동단말용 미들웨어는 운영체제에 ② 시스템 콜을 하고 운영체제는 미들웨어에 ⑤ 시스템 리턴을 되돌린다. 일반 PC용 미들웨어의 경우 하드웨어 및 운영체제가 표준화되어 있어 시스템 콜 및 시스템 리턴이 단일화되어 있다. 그러나 이동단말용 운영체제는 다양한 하드웨어와 운영체제로 인해 다른 형태의 시스템 콜 및 리턴이 있어 각기 다른 하드웨어 및 운영체제를 지원해야 하므로 표준화 되어 있지 않다.

이동단말용 미들웨어 상호운용성 테스트는 미들웨어가 다양한 하드웨어 및 운영체제로 구성된 멀티 플랫폼에서 결합 없이 동작하는지를 확인하는 시험이다. 일반적으로 이동단말용 미들웨어는 단말기에 특정 하드웨어(예: DMB 수신기)를 설치한 후 이를 이용하여 서비스를 제공하는 특징이 있다. 이때 대부분 운영체제는 특정 하드웨어의 디바이스 드라이버를 제공하지 않아 미들웨어가 해당 하드웨어의 디바이스 드라이버를 제공해야 한다. 예를 들어, PDA 및 휴대폰에 WinCE 및 임베디드 리눅스 운영체제를 각각 설치한 후 DMB 서비스를 제공하려고 하면 WinCE 및 임베디드 리눅스용 DMB 하드웨어 디바이스 드라이버를 모두 제공해야 한다.

미들웨어가 멀티 플랫폼을 지원하기 위해서는 드라이버 포팅과 시스템 콜 포팅이 필요하다. <표 1>은 멀티 플랫폼 지원 체크리스트로써 미

<표 1> 멀티 플랫폼 지원 체크 리스트

| 대항목 | 멀티 플랫폼 지원 여부 | | |
|-------------------------|--------------|-------|-------|
| | 플랫폼 1 | 플랫폼 2 | 플랫폼 3 |
| System Call Porting | Y/N | Y/N | Y/N |
| Device Driver Porting | Y/N | Y/N | Y/N |
| System Call Interface | Y/N | Y/N | Y/N |
| Device Driver Interface | Y/N | Y/N | Y/N |



[그림 4] 이동단말용 미들웨어 테스트 프레임워크

들웨어가 각 플랫폼의 시스템 콜 및 디바이스 드라이버의 포팅과 시스템 콜 및 디바이스 드라이버 인터페이스를 지원하는지를 시험하기 위한 체크리스트의 예이다. 테스터는 <표 1>과 같은 체크리스트를 활용하여 각각의 운영체제에 대해 반복적인 테스트를 수행하여 미들웨어가 상호운영 가능한지 확인할 수 있다.

2.6 이동단말용 미들웨어 테스트 프레임워크

이동단말용 미들웨어는 임베디드 시스템으로써 해당 시스템의 운영체제가 다양할 수 있다. 이러한 다양성은 정적분석 및 동적분석을 적용하여 이동단말용 미들웨어를 시험할 때 다양한 운영체제에서 시험이 수행되어야함을 의미한다. 따라서 이러한 환경에서 일반화된 형태의 테스트 방법이 필요하다. 본 연구에서는 일반화된 테스트 프레임워크인 이동단말용 미들웨어 테스트 프레임워크(MTFM)를 제안하여 이동단말용 미들웨어 테스트에 표준적으로 사용할 수 있는 방법을 제시한다.

MTFM은 크게 디자인 단계, 유닛 테스트 단계, 실시간 테스트 단계로 이루어져있으며 해당 단계에 대한 설명은 아래와 같다.

2.6.1 디자인 단계

디자인 요구사항 및 개발 환경으로 구성되며, 미들웨어의 개발 계획 단계에서 검증과정이 포함된다. MTFM에서는 디자인 단계에 대한 검증은 미들웨어의 기획 및 개발 단계를 통해 Formal Method 방식으로 검증이 완료된 것으로 가정한다.

일반적으로 많이 알려진 요구사항 검증방법으로 Theorem Prover[1]나 Model Checking[2]가 있다. 그러나 요구사항이 많아지면 State Space Explosion문제가 발생하고 이를 해결하기 위한 알려진 기법으로 State Reduction Technique[3]과 Symbolic Approach[4]가 있다. 이러한 기법들의 문제점은 요구사항 검증을 통과하더라도 구현에서 문제가 없음을 보장하지 못하는데 있다 [5]. 따라서 스펙에 따라 구현된 함수들에 대해 유닛테스트 및 실행시간 테스트를 통해 검증할 필요성이 있다.

2.6.2 유닛 테스트 단계

유닛 테스트 단계는 Implementation Unit, API Prototype Generator, Analyzer Script의 3개 모듈로 구성되어 있다. 정적분석은 유닛테스트 단계에서 이루어진다. 개발된 API함수 정의를 활용하여 정적 분석을 실행한 후 그 함수의 정상동작 여

부를 확인하는 시험이다.

1) Implementation Unit

디자인 단계에서 각종 요구사항 또는 표준에 따라 개발된 API함수 라이브러리를 일컫는다. 즉 정적분석을 위한 객체에 해당한다.

2) API Prototype Generator

Implementation Unit은 API함수 라이브러리로 테스트를 위해서 필요한 각 함수의 입력 파라미터 및 리턴값이 제공되지 않는다. 정적분석을 위해서 입력 파라미터 및 리턴값에 대한 정의가 필요하다. API Prototype Generator는 각 API함수 별로 미들웨어 레퍼런스 매뉴얼을 참조하여 입력 파라미터 및 리턴값을 생성하는 모듈이다.

3) Analyzer Script

정적분석을 위한 객체는 API함수 즉 Implement Unit이며, 입력 및 출력은 API Prototype Generator를 통해 정의 되었으므로 Analyzer Script는 이를 이용하여 실제 정적 분석을 위한 테스트 코드이다.

유닛 테스트 단계는 정적분석을 적용하는 단계이다. 즉, 미들웨어 레퍼런스 매뉴얼 상에 정의된 입·출력 파라미터와 API함수가 실제 미들웨어 라이브러리에 정상적으로 구현되었는지, 정상동작하는지 여부를 확인하는 테스트 단계이다.

이 단계에서 자동화된 정적분석 도구를 사용할 수 있다. API함수(Implementation Unit)을 테스트하기 위해 Analyzer Script의 종류 중 정적분석 툴을 이용하여 화이트 박스 테스트를 수행하는 형태가 한 예이다.

2.6.3 실시간 테스트 단계

정적분석이 끝난 후 실시간 테스트 단계에서는 미들웨어에의 표준적합성 및 상호운용성을 시험하는 실행시간 테스트를 수행한다.

1) Function Description

미들웨어는 필수 및 부가기능이 제안된 기술문서 또는 표준문서를 요구사항으로 하여 개발된다. 기능모듈 테스트를 위해서는 해당 미들웨어의 주요기능 및 부가기능에 대해 숙지하고 있어야하며, 해당내용은 Function Description에 기술된다.

2) Scenario Generator

기능에 대한 테스트는 해당 기능실행 시나리오를 작성하고 이를 기반으로 테스트 케이스를 산출하는 방식을 사용할 수 있다. 본 모듈은 기능에 대해 테스트하고자 하는 시나리오를 작성하는 모듈이다.

3) Function Evaluator

미들웨어의 기능은 단순히 하나의 API로 구현되지 않고 다양한 API들의 유기적 구성으로 이루어진다. 따라서 미들웨어 표준 및 기술문서에서 제안하는 기능의 정상동작 여부를 확인하기 위해서는 API 라이브러리에서 제공하는 함수를 사용하여 구현한 후 테스트할 필요가 있다. 이 때 기능모듈 테스트를 위해 필요한 기능은 Function Description에 정의되어 있으므로 이를 참조하여 기능을 구현한다.

실시간 테스트 단계에서는 Function Description을 통해 정의된 기능을 Function Evaluator에서 구현하고 Scenario Generator에서 정의된 기능테스트 시나리오를 통해 표준적합성 및 상호운용성을 시험하게 된다. 이때 Function Evaluator는 자동화 도구를 사용한 자동 테스트나 수동 테스트가 모두 사용될 수 있다.

앞에서 설명한 이동단말용 미들웨어 테스트 방법 중 정적분석은 유닛 테스트 단계에 해당하며 동적분석은 실시간 테스트 단계에 해당한다. 실제 테스트가 진행될 때에는 유닛 테스트 및 실시간 테스트 단계에 필요한 자동화 도구 및 수동테스트를 병행하여 테스트를 진행할 수 있다. 또한 이동단말용 자동화 테스트 도구가 없을 경우는 해당

시스템 및 각 단계별로 적합한 자동화 도구를 자체 개발하여 사용할 수도 있다.

3. 결 론

본 논문에서는 이동단말용 미들웨어 테스트의 표준적합성 및 상호운용성을 시험을 위해서는 정적 분석과 동적 분석이 병행되어야함을 강조하였다. 또한 일반화된 이동단말용 미들웨어 테스트 프레임워크인 MTFM을 제안하여 이동단말용 미들웨어를 테스트 시 정적 및 동적 분석을 병행 적용하여 테스트하는 정형화된 방법을 제안하였다.

MTFM을 이용하여 이동단말용 미들웨어를 테스트할 경우, 정적분석과 동적분석을 병행한 상호호환적 테스트 방법을 이용하여 미들웨어의 표준적합성 및 상호운용성 시험에 정형화된 방식을 적용할 수 있다.

참 고 문 헌

- [1] Clarke, E. M., Emerson, E. A. and A. P. Sistla, "Automatic Verification of Finite-State Concurrent Systems Using Temporal Logic Specification", ACM Transactions on Programming Languages and Systems, Vol.8, No.2(1986).
- [2] Gordon, M. J. C., "HOL: a proof generating system for higher-order logic", VLSI Specification, Verification and Synthesis, 1988, pp.73-128.
- [3] Kurshan, R. P., "Reducibility in analysis of coordination", LNCS, volume Vol.103, 1987, pp.19-39.
- [4] Burch, J., Clarke, E., Dill, D., Hwang, L. and K. McMillan, "Symbolic model checking: 10^{20} states and beyond", Proceedings of the 5th IEEE Symposium on Logic in Computer Science, 1990.
- [5] Ben-Abdallah, H., Clarke, D., Lee, I. and O. Sokolsky, "PARAGON: A Paradigm for the Specification, Verification, and Testing of Real-Time Systems", IEEE Aerospace Conference, Vol.2, 1997, pp.469-488.

◆ 저 자 소 개 ◆



신 석 규 (skshin@tta.or.kr)

충남대 대학원에서 데이터베이스를 전공하고, 호서대 벤처전문대학원에서 SW 공학 석사를 이수 한 후, 동대학원 SW 공학 박사 과정에 재학 중이다. 주요 관심분야는 SW 품질 공학, SW 테스트, 평가 모델, 표준화 등이다.



이 상 수 (sslee@icu.ac.kr)

한국과학기술원, 경상대학에서 전자공학을 전공하고, 한국정보통신대학교에서 컴퓨터공학 석사과정을 마쳤다. 주요 관심분야는 유비쿼터스 컴퓨팅, 텔레매틱스, 소프트웨어 공학 소프트웨어 테스트 등이다.



양 해 술 (hsyang@office.hoseo.ac.kr)

홍익대학교 전기공학으로 학사, 성균관대학교 정보처리학과 석사, 일본 오사카대학 기초공학부 정보공학과 SW 공학 박사학위를 취득하였다. 일본 오사카대학교 객원연구원으로 재직했으며, 한국 SW 품질연구소장을 역임했으며, 호서대학교 벤처전문대학원 교수를 역임하고 있다. 주요 관심분야는 소프트웨어공학(특히, SW 품질 보증과 품질 평가, 품질 감리, 품질 컨설팅, SW 프로젝트 관리, CBD 개발 방법론)이다.