

□신기술해설□

# 프로토콜 적합성 시험과 Concurrent TTCN 컴파일러

김기영\* 하수철†

◆ 목 차 ◆

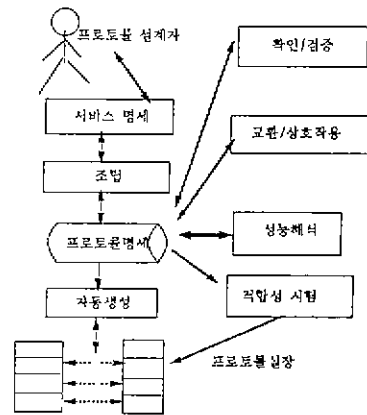
1. 서론	4. Concurrent TTCN 개념 모델
2. 적합성 시험	5. Concurrent TTCN 컴파일러
3. TTCN과 concurrent TTCN	6. 결론

## 1. 서론

현재 고도의 데이터 처리 기술, 신호처리 기술, 고속 디지털 동기망 기술, 광가입계 기술, 지적 단말기술 등과 같은 통신 망 구성에 관련된 기술의 발달과 가용한 통신 서비스의 범위가 확대되어 이용자 요구를 충족 시켜줄 수 있는 B-ISDN 및 초고속 망의 구축이 진행되고 있다.

그런데 B-ISDN의 각 장치들은 서로 다른 팀에 의한 개발, 또는 복수 회사에 의해 제작된 후 사용자에게 제공되고 프로토콜들의 복잡성 때문에 서로 다르게 해석되어 구현된 경우가 발생할 수 있고, 프로토콜의 많은 선택사항으로 인하여 구현된 제품 간의 연동 문제가 발생하는 경우도 있다. 따라서 모든 프로토콜 구현물에 대한 검증과정이 필요하다. 이를 위한 시험이 적합성 시험(conformance test)

으로 프로토콜 공학과의 관계 개요는 (그림 1)과 같다.



(그림 1) 프로토콜 공학 체계도

프로토콜/통신 소프트웨어 수명주기는 요구사항 정의 단계, 설계 단계, 제조 단계, 시험 단계, 유지보수 단계로 구분할 수 있는데, 각각에 적합한 언어들이 존재한다. 이들을 FDT(Formal Description Technique), 프로그래밍 언어, 프로토콜 기법

\* 본 연구는 한국전자통신 연구소의 위탁과제인 "Concurrent TTCN 컴파일러 연구"의 일부로 수행되었음

† 정 화 원 . ETRI 초고속망 연구실 선임연구원

†† 중신회원 : 대전대학교 컴퓨터공학과 교수

으로 나눌 수 있다. FDT에는 SDL [1], Estelle [2], LOTOS [3], Petri Net[4], Z[5], OBJ[6] 등이 있고, 프로그래밍 언어에는 CHILL[7], C, Ada 등이 있으며, 프로토콜 기법에는 ASN.1[8], TTCN[9] 등이 있다. 이들 중 본 논문에서는 TTCN에 대한 사항만을 논한다.

본 논문의 2장에서는 적합성 시험에 대해 기술하며, 적합성 시험 산정 과정에서 산출되는 추상 시험 규격과 TTCN에 대한 관계도 논의한다. 3장에서는 TTCN과 concurrent TTCN에 대해 논의한다. 4장에서는 concurrent TTCN의 개념 모델을 설명하기 위해 시험기를 구성하며 이들에 대한 원리를 기술한다. 5장에서는 concurrent TTCN 컴파일러의 도구를 통해 컴파일 과정과 시험 항목과 관련된 수행 과정을 기술하며, 6장에 논문의 결론을 내린다.

## 2. 적합성 시험

### 2.1 개요

적합성 시험은 OSI 구현의 정적 능력과 동적 행위를 검사하는 것이다. 이것은 구현 제품 공급자에 의해 만들어진 주장을 검증하는 역할을 한다. 이러한 주장은 OSI 프로토콜 표준 또는 기능적 프로파일(profile)을 참조할 수 있다.

2가지 특성이 적합성 시험의 표준화에 필요하다. 첫 번째는 다양한 시험 방법을 식별하는 적합성 시험 방법론으로 ISO/IEC 9646[9]에 지정되어 있다. 두 번째는 적합성 시험을 위해 사용되는 시험 규격(test suite)의 식별이다.

ISO/IEC 9646은 다음과 같이 7부로 구성된다.

- 1부 : 일반 개념
- 2부 : 추상 시험 규격(ATS) 명세
- 3부 : TTCN
- 4부 : 시험 실행
- 5부 : 적합성 산정과정을 위한 시험자와 수험자에 대한 요구사항

-6부 : 프로토콜 프로파일 시험 명세

-7부 : 구현 적합성 문장

### 2.2 적합성 시험의 목적

시험될 구현물을 IUT(Implementation Under Test)라 부른다. IUT는 SUT(System Under Test)로 알려진 전체 시스템이거나 실 시스템의 일부가 될 수 있다. SUT의 구성은 IUT에 적용될 시험 방법에 영향을 준다. 예를 들면, (그림 2)에서처럼 비 OSI 프로토콜을 포함하고 있는 SUT인 경우 적합성 시험시에 사용되는 PCO(Points of Control and Observation)은 SUT의 OSI 부분에 대해서 만으로 제한되어야 한다.

7	비 OSI
6	
N	
2	
1	

(그림 2) 비 OSI 프로토콜을 포함한 SUT

적합성 시험은 시험자와 수험자를 포함하는데 수험자는 적합성 시험을 위해 IUT를 제출한 조직인 반면 시험자는 시험기를 사용하여 적합성 시험을 수행하는 조직이다. 적합성 산정은 IUT 외부에 대한 하나 이상의 포인트에서 감시되고 통제되는 IUT의 외부 행위에 기반을 두고 있다.

### 2.3 적합성 요구사항

적합성 요구사항은 정적 적합성 요구사항 및 동적 요구사항으로 프로토콜 표준이나 기능적 프로파일로 정의된다. 정적 적합성 요구사항은 구현물의 최소 허용 능력을 정의한다. 즉, 프로토콜 구현물이 무엇을 포함해야 하는 가이다. 이의 예로는 구현물이 지원해야만 하는 서비스 원소와 기능적 단위를 들 수 있다. 동적 적합성 요구사항

은 통신의 구체 예에서의 허용된 행위를 정의하는데 프로토콜 표준에서 지정된 프로토콜 절차의 원소에 입각하여 정의된다.

시험이 시작되기 전에 수험자는 SCS(Systems Conformance Statement)에서 언급된 대로 구현될 프로토콜 마다 PICS(Protocol Implementation Conformance Statement)를 제공해야 한다. SCS는 OSI 프로토콜이 구현되고 적합성이 주장된 요약 문서이다. PICS는 SCS에서 언급된 각 프로토콜의 능력을 기술한다.

PICS가 작성된 형식을 PICS proforma라 부른다. PICS proforma는 프로토콜 또는 기능 프로파일 명세의 일부이어야 하는데, 각 적합성 요구사항에 대해 하나의 테이블을 가진 테이블들의 집합이다.

수험자가 시험자에 제공해야할 또 다른 양식이 PIXIT(Protocol Implementation eXtra Information for Testing)이다. 이 양식은 시험자에 SUT에 관한 필수 정보를 제공한다. SUT에 관한 정보는 네트워크 주소, 전력 요구사항, 시험기에 의해 요구되는 프로토콜 값 등을 포함한다. PIXIT proforma는 PIXIT가 작성된 양식이다. 이것은 시험자에 의해 제기되어 수험자에 의해 완성된다. 시험자는 PIXIT 정보를 MOT(Means of Testing) 결정에 사용한다.

## 2.4 적합성 시험의 형태와 적합성 산정 과정

### 2.4.1 적합성 시험 형태

일반적으로 무한개의 사건이 PCO에서 발생할 수 있기 때문에 소모적인 적합성 시험 대신에 시험의 적용범위와 실행 비용간의 절충이 필요하다. 비 적합 구현물은 주어진 환경에서 적합성 시험을 통과할 수 있지만 다른 환경에서는 실패할 수 있다.

적합성 시험을 다음과 같이 구분한다.

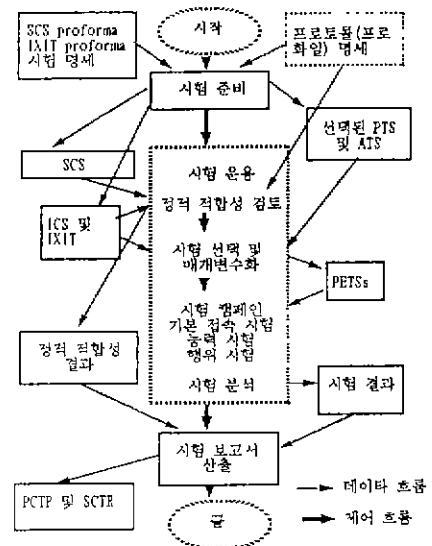
- (1) 기본 상호접속 시험(interconnection test): 철저한 시험을 수행하지 않고 상호접속에 대한 충분한 적합성을 달성하기 위한 주요 특징의 제

한적인 시험이다. 이러한 시험은 구현물의 공급자에 의한 적합성 주장의 기본으로 사용될 수 없다.

- (2) 능력 시험(capability test): 정적 적합성 요구사항의 각각에 대한 시험이다. IUT와 PICS의 일관성을 철저히 대조한다.
- (3) 행위 시험(behavior test): 가능한 한 동적 적합성 요구사항의 전 범위에 대해 이해력 있는 시험으로, 시험기에 의해 전송된 유효, 무효, 부적당한 PDUs(Protocol Data Unit)에 반응하여 IUT에 의한 행위를 시험한다.

### 2.4.2 적합성 산정 과정

적합성 산정 과정은 크게 시험 준비, 시험 운용, 시험 보고서의 산출로 나눌 수 있다.



(그림 3) 적합성 산정 과정

시험 준비 단계는 SCS, PICS, PIXIT 및 추상 시험 방법 및 추상 시험 규격의 산출 과정이다. 수험자가 적합성 시험의 실행을 위해 시험자와 접촉하기 전에 추상 시험 방법(ATMs) 중 적어도 하나를 사용하여 SUT가 시험 가능하다는 것을 먼저 보증해야 한다. ATM은 제어 및 관찰이 발생하는 IUT에 가장 근접한 포인트를 식별한다. 또

한 적합성 시험 시 IUT에 적용될 추상 시험 규격(ATs:Abstract Test Suite)이라 하는 시험 항목(test case) 집합을 결정한다. 수험자는 제안된 ATMs에 의해 약정된 대로 제어 및 관찰의 필수 수단을 SUT가 제공하도록 보증해야 한다. 이러한 기본적인 수험자의 노력 후에 수험자는 IUT에서 구현되며 적합성이 시험되기 위한 각 프로토콜 표준 또는 기능 프로파일에 대해 PICS를 완성해야 한다. 또한 적합성 산정 과정 동안 사용될 각 ATM에 대한 PIXIT도 완성해야 한다.

완성된 목록은 수험자의 ATMs 선택을 조정하는 시험자에 의해 검토된다. 관련된 ATs에 대해 시험자는 MOT를 식별한다. MOT는 시험 시스템, 실행 가능한 시험 규격, 필수 시험 지원 도구를 지정한다. 여기서 ETS(Executable Test Suite)는 어떤 실행 가능한 프로그래밍 언어로 작성된 ATs의 실행을 말한다.

시험 운용 단계를 위해 MOT와 SUT를 준비한다. 시험 운용 단계는 PICS 및 PIXIT에 기반을 둔 정적 적합성 검토, 시험 선택 및 매개변수화, 그리고 하나이상의 시험 캠페인(campaign)을 포함한다. 정적 적합성 검토 시 시험자는 일관성이 있는지를 대조하기 위해 PICSs를 분석한다. PICS의 정보에 기반을 두어 시험자는 IUT에 적합한 추상 시험 항목을 선택한다. 이러한 시험 항목은 능력과 행위 시험 항목 모두를 포함한다. 이러한 시험 항목은 선택된 집합을 SATS(Selected Abstract Test Suite)라 한다.

PIXIT에서 제공된 정보는 SATS의 적합한 매개변수값을 결정하는데 사용된다. 결과의 PETS(Parameterized Executable Test Suite)은 실행 준비가 되어 있다. 시험 조정 절차가 ATM에 대해 요구되어야 하기 때문에 MOT와 SUT 어느 것이 요구된 시험 조정 절차를 사용할 수 있는지도 조사되어야 한다.

시험 캠페인은 PETS를 실행하고 수험자에 의해 요청된다면 로그 정보를 산출하는 과정이다. 따라서 여기서 동적 적합성 시험이 발생하는 단

계이다.

시험 캠페인 동안 pass, fail, inconclusive, 시험 항목 오류, abnormal 시험 항목 종료와 같은 판정이 시험에 배정된다. 시험 보고서 산출 단계에 앞서, 시험자는 fail 판정을 나타낸 시험 항목을 수험자에게 통보한다. 협상 출구가 시험 캠페인 동안 만들어 질 수 있다.

시험 운용이 끝난 후 IUT의 적합성 산정이 만들어지고 SCTR(System Conformance Test Report)과 PCTR(Protocol Conformance Test Report)가 만들어진다. SCTR은 시스템 적합성의 전체 요약이고, PCTR은 연루된 각 프로토콜에 대해 수행된 시험의 세부사항이다.

## 2.5 추상 시험 규격과 TTCN

ATs는 각 시험 항목이 식별된 목적을 가진 다수의 시험 항목으로 이루어진다. 시험기와 SUT에 에워 쌓인 환경을 고려하지 않기 때문에 추상적이다. 따라서 시험 운용이 시작되기 전에 프로그래밍 환경을 고려한 ETS로 전환되어야 한다.

ATs의 설계는 PCOs 위치와 선택된 ATM에 의존한다. 일반적으로 각 적용 가능한 ATM에 대해 하나 이상의 ATs가 존재한다.

모든 시험 항목은 시험 목적(purpose)을 가진다. ATs의 시험 항목은 시험 그룹으로 집단화 될 수 있다. 필요시 시험 그룹은 시험 그룹 내에 내포될 수도 있다. 시험 항목을 그룹화 하는 기준은 다음과 같다.

- 능력 시험은 의무적 특성, 선택적 특성을 기준으로 그룹화 될 수 있다.
- 행위 시험은 유효한 행위의 시험, 무효 행위의 시험, 부적당 행위의 시험으로 그룹화 될 수 있다.
- 행위 서브 그룹은 접속 설치 단계, 데이터 전송 단계, 접속 해제 단계로 세분화 될 수 있다.
- 데이터 전송 단계는 IUT에 전송된 PDUs와 IUT로부터 수신된 PDUs에 초점을 두고 그룹화 될 수 있다.

ATS의 추상적 성격 때문에 시험 몸체는 형식 표기법으로 기술되어야 한다. ISO/IEC 9646-3은 이 목적으로 TTCN을 정의하고 있다. TTCN은 PDUs 교환하기와 관련된 행위 및 동작의 트리를 기술하기 위한 명세 언어이다.

### 3. TTCN과 concurrent TTCN

#### 3.1 TTCN의 정의

TTCN의 사용은 적합성 시험 환경에 매우 중요하다. TTCN이 어떠한 프로토콜과 계층에도 적용되는 포괄적인 표기법이기 때문에 시험도구가 단일 시험 규격 이상을 처리하도록 작성될 수 있다. 이것은 시험 제공자에게 비용절감 효과를 줄 수 있는데, 프로토콜 구현자가 시험결과의 향상된 비교가능성으로부터 이득을 얻을 수 있다는 의미에서 다른 구현 가능성을 감소시키는 것이다.

TTCN은 다수의 보충적인 방식으로 기술된다. 표준의 몸체는 예제에 의해 확장되는 문장형태의 기술로 이루어진다. 문서의 규범적인 부록은 TTCN에 대한 BNF(Backus-Naur Form)을 갖고 있으며, 별개의 규범적인 부록은 TTCN의 연산적 의미론 정의를 포함하고 있다.

TTCN 시험은 "TTCN.GR"(Graphical)이라 부르는 페이지 버전, 또는 "TTCN.MP"(Machine Processable)이라 부르는 기계 판독 버전 중의 하나로 작성될 수 있다.

TTCN에 대한 BNF는 LALR(1) 파싱도구에 의해 처리되도록 설계되었으며, TTCN 문법의 BNF 기술과 TTCN의 문장적 기술은 동일하다. 만일 텍스트가 BNF에 의해 지정된 것과 다르게 해석된다면, TTCN 문장에서 오류로 간주되며 표준 교정과정을 통해 교정되어야 한다. 이러한 과정이 완료될 때까지 BNF 명세는 문장보다 우선권(precedence)을 갖는다. 이것은 논리적인 우선 순위로서, TTCN 시험 항목의 처리를 구현하는 시험도구는 엄격하게 BNF를 구현하도록 요구되기

때문이다.

(그림 4)는 FTAM(File Transfer, Access and Management) 프로토콜에 대한 TTCN 기술의 예를 보여준다.

```

U! Initialize
Start(A, time-duration)
L? FINIRQin[is-valhd-fus AND is-valid-ctl]
Cancel(A)
+ abort-postamble
    
```

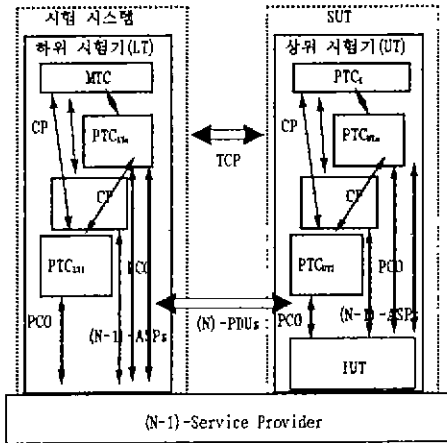
(그림 4) TTCN 기술의 예

이 예에서 "!"는 송신(send)을 "?"는 수신(receive)을 의미한다. "U"는 상위 시험기(upper tester), "L"은 하위 시험기(lower tester)를 의미한다. "+"는 행위 트리를 소규모로 분리하는 부착(attach)기법이다. 트리부착은 시험 항목을 이해하기 쉽게 하기 위해 사용한다. 예를 들면, 시험 항목을 전문(preamble), 몸체, 후문(postamble)으로 나누는 경우가 그 예가 된다. 시험 항목이 프로토콜의 FSM(Finite State Machine) 명세에서 어떤 연결선(edge)을 시험하기 위해 사용된다고 간주한다면, 시험 항목의 전문은 IUT를 연결선의 시작 상태로 이끄는 역할을 하고, 시험 몸체는 입력 사건을 시물레이션하고 출력 사건을 관찰하는데 사용한다. 후문은 IUT가 종료 상태에 대응하여 마무리되는 지를 결정하기 위해 사용된다.

#### 3.2 Concurrent TTCN으로의 접근

적합성 표준의 기본 전제는 IUT라 부르는 프로토콜의 구현이 블랙 박스라는 것이다.

IUT의 적합성에 관하여 고려 볼 수 있는 것은 IUT의 하위(lower) 및 상위(upper) 서비스 인터페이스에서 발생하는 사건을 관찰하고 제어함으로써 만들어진다. ISO/IEC 9646 항목에서 이러한 상호작용은 PCO에서 발생하며 ASP(Abstract Service Primitives)에 내장된 PDU의 견지에서 표현된다.



(그림 5) 일반화된 병렬 시험 구조(단일 계층)

IUT는 시험 시스템에 의해 시험된다. TTCN에서 시험 시스템의 다른 부분을 시험 컴퍼넌트(test component)라 부른다.

하위 인터페이스에서 PCO들을 통해 IUT와 통신하는 시험 컴퍼넌트들을 묶어서 하위 시험기(LT)라 부르며, 상위 인터페이스에서 PCO를 통해 IUT와 통신하는 시험 컴퍼넌트들을 상위 시험기(UT)라 부른다. 적어도 하나의 시험 컴퍼넌트는 항상 시험 시스템에 나타나야 한다. 이것을 MTC(Main Test Component)라 부르는데, 시험을 조정하고 제어하는데 책임이 있으며 시험의 최종 판결을 내린다.

LT에서 시험 컴퍼넌트들 간의 통신은 CP(Coordination Points)를 통해 달성된다. 마찬가지로 UT 시험 컴퍼넌트는 CP를 통하여 각각 다른 것파 통신할 수 있다. LT와 UT간의 조정은 TCP(Test Coordination Procedures)에 의해 달성된다. 하위 시험기는 송신하고 수신하는 ASP들에 내장된 PDU의 제어와 관찰에 책임이 있어 2개의 컴퍼넌트 중에서 더 복잡하다. 실제로 LT는 시험 항목을 실행할 때 주어진 시간 내에 적절한 프로토콜을 구현하고 있는 것이다.

IUT를 시험하기 위하여 시험 시스템을 제어하

고 관찰하기를 바라는 상호작용의 연속순서(sequence) 또는 시험 사건(event)을 지정할 필요가 있다. 완전한 시험 목적을 지정하는 사건들의 연속 순서를 시험 항목이라 하며, 특정 프로토콜에 대한 시험 항목의 집합은 시험 규격이라 부른다.

시험 항목이 궁극적으로 운용될 실제 시험 시스템의 구조로부터 추상화되는 어떤 레벨에서 TTCN은 시험 항목의 명세를 위해 개발된 표기법인 것이다.

ISO 적합성 표준은 시험이 (N-1)-layer ASP들, (N)-layer ASP들 및 (N)-layer PDU들에 따라 지정되도록 요구한다. 이러한 요구 사항을 만족시키기 위해 TTCN이 구비해야하는 최소의 기능은 다음과 같다.

- 시험 시스템에 의해 송신되거나 수신되는 ASP들을 지정하는 능력
- ASP들에 내장된 PDU들을 지정하는 능력
- 특정 PCO들에서 ASP들이 송신되거나 수신되는 순서의 명세 능력

### 3.3 Concurrent TTCN

통신 네트워크 분야에서 새로운 기술의 진화는 다양한 프로토콜과 다중 접속(multiple connection)을 포함하는 새로운 서비스의 정의를 유지 보수할 수 있게 만들고 있다. 이러한 프로토콜에 대한 시험환경은 더욱 복잡하여 기존의 OSI[ISO 9646-1]의 문맥에서 정의된 시험 방법을 벗어나고 있다. 이것은 TTCN의 확장인 병행(concurrent) TTCN의 정의를 제정하게 하였다.

TTCN과 달리 병행 TTCN은 시험 항목의 실행에 하나 이상의 활성화된 시험 컴퍼넌트가 참여할 수 있다. 모든 시험 컴퍼넌트는 병렬로 운용되고 조정 메시지(coordination message)를 교환함으로써 그들의 행위를 조정한다. 병행 TTCN의 잠재적인 이점은 복잡한 시험환경에 대한 시험 항목 기술이 훨씬 용이해 진다는데 있다. 반면 시험 항목의 유도(derivation)와 확인(validation)이 더욱 어려워진다는 단점이 있다.

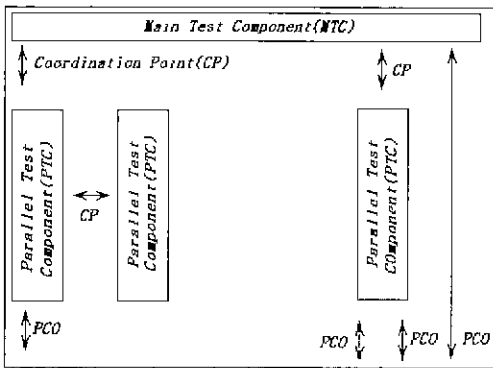
### 4. Concurrent TTCN 개념 모델

개념 모델은 구현 기술을 위한 모델이라기보다 병행 TTCN의 개념 및 의미론의 이해를 돕기 위한 것이다[10].

#### 4.1 Concurrent TTCN 시험기의 구성

TTCN의 관점은 시험 항목을 실행하는 단일 시험 컴퍼넌트(TC)에 있다 반면, 병행 TTCN은 병렬로 실행하는 다수의 TC에 의해 시험 항목을 실행케 한다. TC는 MTC(Main Test Component) 또는 PTC(Parallel Test Component)중의 하나이다. 시험기는 정확히 하나의 MTC와 몇 개인가의 PTC들로 구성된다. TC는 CM(Coordination Messages)을 운송할 수 있는 CP(Coordination Points)에 의해 연결된다. 조정 메시지들(CM)은 그들의 결합 행위를 조정하기 위하여 TC 사이에서 교환된다. 환경과 TC들의 통신은 - 즉 (N-1) service provider 또는 IUT - PCO에서 발생한다. TC는 어떠한 수의 PCO 및 CP와도 접속될 수 있다. PCO는 하나의 TC에만 접속되고 CP는 정확히 두개의 TC를 접속한다.

MTC, PTCs, PCOs 그리고 CPs에 입각한 시험기의 일반적인 구성은 (그림 6)과 같다.



(그림 6) 병행 TTCN 시험기의 구성

정의된다. 행위 트리(behavior tree)는 TC가 실행할 시험 사건들의 상대적인 순서를 정의한다. TTCN에서는 이러한 상호작용의 트리를 행위 트리라 부른다. 일반적으로 다수의 표준화된 서비스 정의와 프로토콜 명세에 대한 행위를 기술하기 위해 상태 다이어그램 또는 상태 테이블을 사용한다. 시험 항목은 이러한 명세로부터 유도된다. 적합성 시험이 서비스 인터페이스에서 상호작용의 PCO에 관심을 갖기 때문에 2개의 주어진 프로토콜 상태간에 발생하는 모든 가능한 연속순서에 대한 분기를 갖는 트리로서 시험 시스템을 기술하는 것이 더 타당하다.

병행 TTCN은 TTCN에 다음과 같은 내용을 확장 정의하고 있다.

시험 항목의 실행은 MTC의 실행으로 시작한다. 이것은 모든 TC들을 설정(set-up)하고, 모든 PCO들과 그것이 연결하는 CP들을 관리하며, 또 최종 관정을 계산한다. PTC들은 요구대로 MTC에 의해 생성될 수 있다. create 연산은 PTC와 행위 트리를 연관시킨다. 행위 트리가 매개 변수화되어도, create는 형식 매개 변수에 대한 실매개변수를 전달한다. 새롭게 생성된 PTC는 MTC와 병행적으로 할당된 행위 트리의 실행을 개시한다. 만일 PTC가 행위 트리의 실행 중간에 있다면 PTC는 정지되고 가능한 다른 행위 트리로 재개된다. PTC가 MTC에 의해 시작된 후 PTC는 행위 트리의 잎(leaf)에 도달하거나 MTC에 의해 명시적 또는 묵시적으로 종료될 때까지 행위 트리의 실행을 계속한다. MTC는 매개변수로서 종료될 PTC와 함께 terminate 연산을 실행함으로써 PTC를 명시적으로 종료시킬 수 있다.

TC들간의 정보항목의 교환은 제한되지 않는다. 모든 종류의 정보 항목은 TC들간에 통신될 수 있다. 정보 항목의 형식은 조정 메시지(CM)형에 의해 주어진다.

TC들간의 통신은 비동기적(asynchronous)이다. CM들의 전송에 대해 CP는 ASP들 또는 PDU들의 전송에 대한 PCO와 같은 역할을 한다. CM들의 중간 저장소를 위한 2개의 큐(queue)(통신의 각

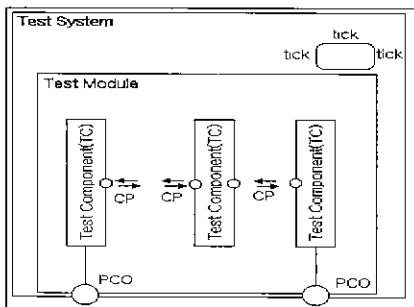
방향에 1개)가 하나의 CP와 연결된다.

각 TC는 자신의 변수와 타이머들을 관리한다. 병행 TTCN의 문맥에서, TC에 지역적인 변수들을 시험 컴퍼넌트 변수라 한다. 전역적으로 정의된 변수에 대한 접근(즉 시험규격 변수)은 MTC에 제한된다.

### 4.2 시험 시스템과 시험 모듈

(그림 7)의 시험 시스템은 추상화의 최상위를 정의하고 있으며, 시험 모듈과 PCO들로 구성되어 있다. PCO들을 통해 시스템은 환경으로부터 ASP들 또는 PDU들을 수신하며, 환경에 ASP들 또는 PDU들을 송신한다. 시험 시스템의 환경은 서비스 제공자(service provider)와 IUT에 의해 주어진다. 시간 제약 조건을 다루기 위해 시험 시스템에서 클럭은 시스템 전역 시간(global time)을 유지한다고 가정한다. 시간은 ticks로 계산되며, tick에 시간 경과가 연결된다. 시험 시스템에서 모든 프로세스는 전역 시간에 접근한다. 관찰자가 시험 시스템을 블랙박스 간주하여 보여지는 그대로 시험 시스템의 관찰 가능한 행위는 환경에 송신되거나 환경으로부터 수신되는 일련의 메시지 연속순서에 의해 결정된다.

PCO는 시험 시스템을 환경에 대한 인터페이스로 제공한다. 개념적으로 각 PCO는 2개의 큐를 가지는데 진입하는 ASP들 또는 PDU들의 중간 저장소에 대해 1개, 그리고 밖으로 나가는 ASP들 또는 PDU들의 중간 저장소에 대해 1개이다. CP들과 PCO들은 개념적으로는 동등하다.

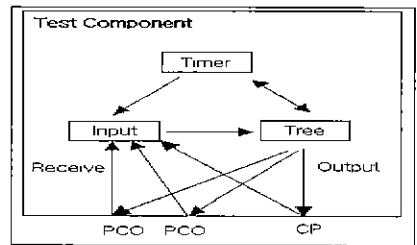


(그림 7) 시험 시스템의 개념 모델

시험기의 모든 TC들은 시험 모듈 안에 상주한다. 내부적으로 시험 모듈은 시험기를 위해 정의된 개수 한계까지 TC 개수를 가진다. TC들은 CP들에 의해 내부연결 되어 있다.

### 4.3 시험 컴퍼넌트(test components)

TC 구조는 (그림 8)과 같으며 3개의 통신 프로세스들로 구성된다.



(그림 8) 시험 컴퍼넌트의 개념 모델

- (1) 트리 프로세스(tree process) : 시험 항목의 일부를 표현한다. 시스템이 set-up(TC가 MTC인 경우)이거나 TC가 생성될 때(TC가 PTC인 경우) 행위 트리로 구체화된다. 트리 프로세스는 TC의 변형 부이다.
- (2) 입력 프로세스(input process) : 모든 진입하는 ASPs, PDUs, CM 그리고 time-out 메시지들을 다룬다. 이 프로세스는 도착 순서로 time-out 메시지들에 개입되는 CP들과 모든 PCO들의 CP들, 모든 ASP들, 그리고 PDU들을 포함하는 입력 스트림(input stream)이라 불리는 자료 구조를 유지한다. ISO 9646-3에서 언급된 진입하는 ASP들 또는 PDU들에 대한 PCO 큐들은 입력 스트림에 의해 모델화된다. 메시지가 입력 스트림 내에 있다면, 그것은 TC에 의해 처리될 수 있다. 입력 스트림의 내용은 스냅샷(snapshot)을 표현한다. 트리 프로세스는 입력 스트림의 현재 내용에 따라 일련의 대안(alternatives)들을 평가한다.
- (3) 타이머 프로세스(timer process) : 모든 유행



타이머를 추적한다. 타이머 프로세스는 실행 타이머의 유한 순서화된 연속순서를 유지한다. 다음에 소멸되는 타이머는 연속 순서의 첫 번째이다.

## 5. Concurrent TTCN 컴파일러

여기에서는 Telelogic 사가 개발한 ITEX[11]를 이용하여 시험 규격이 작성되고 실행 가능한 코드가 생성되는 과정을 논의한다.

### 5.1 컴파일링 도구

ITEX-DE는 TTCN 시험 규격을 생성하고, 편집하며, 유지보수하기 위한 통합도구이다. 이들은 다음과 같은 구성을 하고 있다.

- (1) 브라우저  
시험 규격의 서로 다른 컴퍼넌트를 표현하고 접근하기 위해 사용하며, 텍스트 지향으로 시험 규격 구조와 테이블 목록을 표시한다.
- (2) 테이블 편집기  
concurrent TTCN 테이블을 포함하여 표준화된 TTCN.GR 테이블을 편집한다.
- (3) 그래픽 트리 편집기  
행위 테이블을 편집한다. 트리 지향 뷰는 TTCN 코드를 표시하는데 사용한다.
- (4) 문장 편집기  
TTCN 테이블, 브라우저, 트리 편집기, 다이얼로그 등에서 가능하다.
- (5) 분석 도구  
TTCN과 ASN.1에 따른 시험 규격의 구문과 정적 의미를 대조한다.
- (6) TTCN.GR 출력 도구  
ISO/IEC 9646-3에 따라 TTCN.GR형식의 시험 규격을 출력한다.
- (7) Export 도구  
TTCN.MP 형식의 시험 규격을 출력한다.

- (8) Import 도구  
TTCN.MP 형식의 시험 규격을 입력한다.
- (9) ITEX Access  
TTCN으로 작성된 임의의 ATS에 대한 C++ 응용 프로그램 인터페이스이다.
- (10) C 코드 생성기  
TTCN을 ANSI-C로 번역한다.
- (11) 시뮬레이터  
시뮬레이션될 환경에서 시험 규격을 실행하거나 디버그 하는데 사용한다.  
이외에 선택도구, 대치 도구, 탐색 테이블 도구, 보고서 도구, 개요 생성 도구 등이 있다.

### 5.2 컴파일링 과정

ATS는 각 시험 연속순서의 구조적 기술과 마찬가지로 신호와 값의 형식 정의를 포함하고 있다. TTCN은 이러한 시험 연속순서를 기술하기 위해 사용되는 표기법으로, TTCN으로 작성된 ATS에 대한 C++ 응용 프로그램 인터페이스인 ITEX Access는 일종의 TTCN 컴파일러이다. 엄밀하게 말하면 컴파일러의 전반부 단계를 수행하는 front-end 컴파일러이다.

- (1) 어휘 분석기  
ATS의 형식에 따라 2 단계로 수행된다.
  - 원시 코드가 .mp 형식이면 Import 도구를 통해 ITEX와 결합한다. 어휘 분석기는 import 단계에서 시험 규격이 올바른 .mp 형식으로 작성되었는지 검증한다.
  - 원시 코드가 편집기를 통해 ITEX로 직접 작성되면 분석 도구를 사용하여 어휘 분석을 한다.
- (2) 구문 분석기  
시험 규격의 의미를 분석하여 TTCN과 ASN.1 표준 표기법과 부합하는 지를 검증한다.
- (3) 파스 트리

파스(parse) 트리를 생성한다. 어휘 및 구문 분석이 성공적일 때만 수행된다.

(4) 기호 테이블 관리

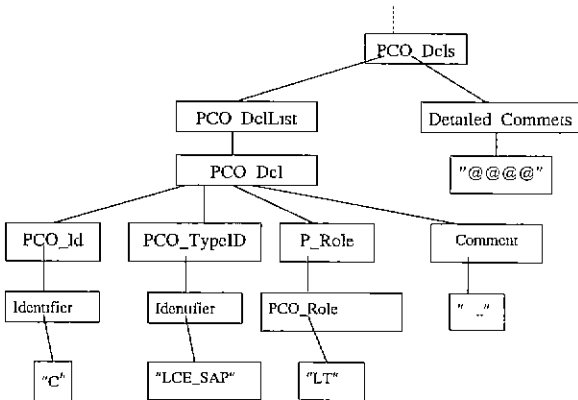
기호(symbol) 테이블은 Access가 실행 중에는 어느 때이든지 접근 가능하다.

이 도구는 시험 규격을 파스 트리로 보기 때문에 파스 트리 처리를 위한 기본 기능을 제공한다. 그것으로는 파스 트리 운행 및 원시 코드 형식인 main() 및 다수의 예제에 대한 템플릿 등이 있다.

(그림 9)는 예제로 PCO 선언을 기술한 것이고 이 테이블을 파싱한 결과는 (그림 10)에 보인다.

PCO Declarations			
PCO Name	PCO Type	Role	Comments
C	LCE_SAP	LT	....
Detailed Comments: @@@@			

(그림 9) PCO 선언 테이블의 예



(그림 10) 그림 10의 파스 트리

이 파스 트리는 목시적으로는 깊이 우선 탐색 방식으로 운행을 하지만 탐색 방식을 조정할 수 있다.

ATS를 실행 가능한 시험 규격으로 변환하는 작업은 비트 패턴과 같은 실제 ASP와 PDU 신호를 표현하는 문제를 갖고 있다. TTCN에서 비트 패턴으로의 전환 단계는 시험 시스템과 시험 환

경과 같은 실제 프로토콜의 방대한 지식을 갖고 구현되어야만 하기 때문이다. 이를 위해 다음과 같은 가정을 가진 인코더를 사용한다.

- 같은 형의 원소들은 동일하게 인코딩된다.
- ASP 또는 PDU의 인코딩은 컴퍼넌트를 재귀적으로 인코딩하면서 수행된다.

시험 항목의 간단한 예를 정의하고 이에 해당된 인코딩 함수를 (그림 11)과 (그림 12)에 보인다.

TTCN PDU Type Definition		
PDU Name: AUTHENT_REQT		
PCO Type: LCE_SAP		
Comments:		
Field Name	Field Type	Comment
message_flag	INTEGER	
message_type	BITSTRING[8]	
auth_type	AUTH_TYPE	
Detailed Comments:		

(그림 11) 시험 항목의 예

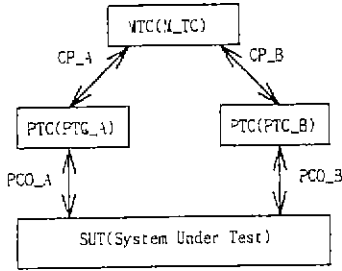
```
void AUTHENT_REQ1_encode(AUTHENT_REQ1* me, char* enc_buf)
{
    INTEGER_encode(&me->message_flag, enc_buf);
    BITSTRING_encode(&me->message_type, enc_buf);
    AUTH_TYPE_encode(&me->auth_type, enc_buf);
}
```

(그림 12) 대응 인코딩 함수

5.3 Concurrent TTCN 컴파일러를 적용한 예

ATM(Asynchronous Transfer Mode) 망을 구성하는 망 장치들은 기본적으로 ATM 계층 프로토콜이 탑재되어야 하고 ATM 계층 프로토콜 시험 규격이 작성되어야 한다. 여기에서는 병행 TTCN 컴파일러(ITEX)를 사용하여 중간 시스템을 위한 ATM 계층 장애 시험 부분의 ATS를 작성하고, 이 ATS를 번역하여 ETS를 작성하는 예를 들어 보인다. TC는 크게 나누어 PTC와 MIC로 구성되고, SUT와 PTC와의 관찰점은 PCO로 정의하고

PTC와 MTC 간의 관찰점은 CP로 정의한다. PTC\_A와 PTC\_B는 독자적으로 수행되며 수행된 결과를 MTC에 전달한다.



(그림 13) 예제의 구성도

여기에서 적용된 시험 항목은 VP-AIS를 시험기가 PCO\_A에서 발생하면 PCO\_A로 수신된 메시지가 VP-RDI이고, PCO\_B로 수신된 메시지가 VC-AIS인가를 확인하는 시험이다. PTC\_A와 PTC\_B는 독립적으로 수신된 메시지를 처리하며 각각의 관정 결과를 MTC에 CP\_A와 CP\_B를 통하여 전달하게 된다. <표 1, 2, 3>는 병행 TTCN으로 위와 같은 시험 항목을 작성한 TTCN.GR 화일이다. PTC\_A와 PTC\_B에서 수행된 것들은 시험 스텝에서 기술되어 있고 시험 항목 전체적인 흐름은 동적 행위에 기술되어 있다.

(표 1) PTC\_A의 시험 스텝

ATM_UNI_IS Aug 3, 1996 ITEX 02					
Test Step Dynamic Behaviour					
Test Step Name: TRES3					
Group: PTC_A VP-AIS AP-RDI VC-AIS test					
Objective: PTC_A VP-AIS AP-RDI VC-AIS test					
Default: Comments:					
No.	Label	Behaviour Description	Constraints Ref.	Verdict	Comments
1		PCO_A OAM_ER	OAM_VP_AIS/Physical		
2		PCO_A OAM_ER	OAM_VP_RDI/Physical	PASS	
3		(RESULT = 1)			
4		CP_A Result/Result + RESULT	Any_Result		
5		PCO_A OTHERWISE		FAIL	
6		(RESULT = 0)			
7		CP_A Result/Result + RESULT	Any_Result		
Detailed Comments:					

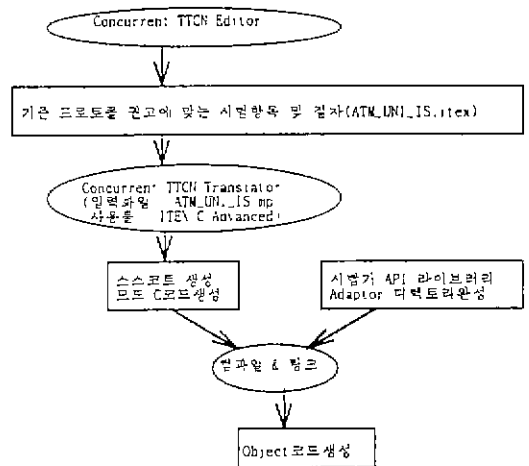
(표 2) PTC\_B의 시험 스텝

Test Step Dynamic Behaviour					
Test Step Name: TRES3					
Group: PTC_B VP-AIS AP-RDI VC-AIS test					
Objective: PTC_B VP-AIS AP-RDI VC-AIS test					
Default: Comments:					
No.	Label	Behaviour Description	Constraints Ref.	Verdict	Comments
1		PCO_B OAM_ER	OAM_VP_AIS/Physical		
2		PCO_B OAM_ER	OAM_VP_RDI/Physical	PASS	
3		(RESULT = 1)			
4		CP_B Result/Result + RESULT	Any_Result		
5		PCO_B OTHERWISE		FAIL	
6		(RESULT = 0)			
7		CP_B Result/Result + RESULT	Any_Result		
Detailed Comments:					

(표 3) MTC의 동적 행위

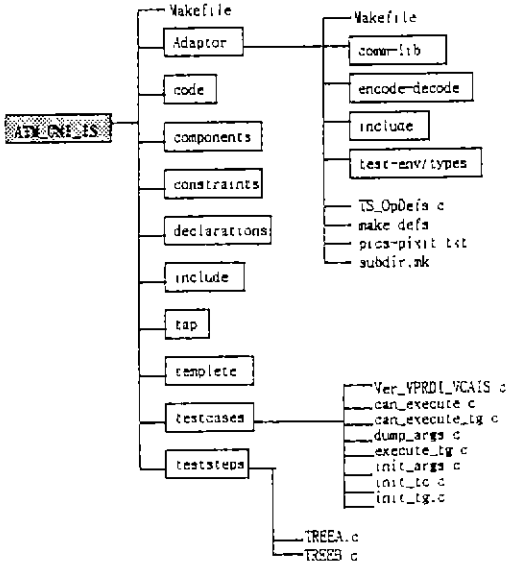
ATM_UNI_IS Aug 3, 1996 ITEX 02					
Test Case Dynamic Behaviour					
Test Case Name: test_PTCO_VCAIS					
Group: M_Suite/Full_Management/Alarm_Surveillance					
Purpose: Verify that the IUT as a VP interface node will send a P-AIS to the downstream node when an upstream physical layer failure is detected.					
Configuration: CONF G					
Default: Comments: Requires a VP connection Ref: 3.3.3 / PCS 3.14.5					
No.	Label	Behaviour Description	Constraints Ref.	Verdict	Comments
1		CREATE PTC_A TRES3			
2		CREATE PTC_B TRES3			
3		CP_A Result/R1 + Result/Result	Any_Result		
4		CP_B Result/R2 + Result/Result	Any_Result		
5		(R1 = 1) AND (R2 = 1)		PASS	
6		(R1 = 0) OR (R2 = 0)		FAIL	
Detailed Comments: Selection Ref: VP_SERV. Makes the IUT send an OAM VP-AIS by causing a physical layer failure between the IUT and the PCO_A.					

이들을 병행 TTCN으로 실행하는 과정은 (그림 14)와 같다.



(그림 14) 병행 TTCN 컴파일러의 수행 과정

병행 TTCN 컴파일러가 생성한 화일 구조는 (그림 15)와 같다.



(그림 15) 병행 TTCN에서 생성된 화일 구조

위의 목록을 보면 시험의 일반 사항과 시험 목적을 기술하는 시험 규격 개요부 디렉토리, PDU 구조에 대한 정보, 프로토콜 관련 타이머, 시험 규격서에서 사용되는 변수를 정의하는 선언부 디렉토리, 선언부에서 선언된 것들의 실제 값을 갖고 있는 Constraint 부 디렉토리, 그리고 프로토콜 시험기와 시험 대상 구현물 사이의 송수신되는 PDU의 흐름과 시험 결과를 판정하는 직접적인 프로시주어인 Dynamic behavior 부 디렉토리들이 생성되고, 각 디렉토리 아래에는 해당 원시 코드들로 이루어져 있다.

## 6. 결론

현재 국내 및 선진국에서는 ITU-T 및 ISO/IEC 9646의 권고 안을 기준으로 TTCN을 사용한 적합성 시험 방법론과 기본 틀을 이용하는

연구를 하고 있다. 기존에는 Point-to-Point 시험 방법이 주류를 이루었고, 현재는 Point-to-Multipoint 방법으로의 접근을 시도하고 있으며, Multipoint-to-Multipoint 시험 방법은 미약한 실정이다.

최근 다자간 화상회의와 같은 Multipoint-to-Multipoint 서비스가 대두되어 이의 서비스를 위한 시스템이 개발 중이므로, Multipoint-to-Multipoint 프로토콜을 위한 시험 환경 구성 방안 및 시험 방법론의 정립이 필요한 실정이다. 그렇지만, 대부분의 기존 방법이 순서적 프로그램에 대한 것이기 때문에, 그들을 병행 프로그램에 대한 시스템 시험에 직접 사용할 수 없다.

따라서 B-ISDN 및 초고속 망 관련 망측 프로토콜의 적합성 시험 규격의 시험가능성도를 향상시키고 작성의 효율성을 향상시키기 위하여 동시 처리 기능을 가지는 병행성을 표현할 수 있는 표기법인 병행 TTCN에 대한 지식이 필요하다. 또한, 이를 컴파일링하는 방법들에 대한 연구가 요구된다.

### [부록] 용어 정의 표

약어	원어	해설
ASP	Abstract Service Primitives	추상 서비스 프리미티브
ATM	Abstract Test Method	추상 시험 방법
ATS	Abstract Test Suite	추상 시험 규격
CM	Coordination Message	조정 메시지
CP	Coordination Point	조정점
IUT	Implementation Under Test	시험 대상 구현
MOT	Means of Test	시험 수단
MTC	Main/Master Test Component	마스터 시험 컴퍼넌트
PCO	Point of Control and Observation	제어 및 관찰점
PDU	Protocol Data Unit	프로토콜 데이터 유닛
PICS	Protocol Implementation Conformance Statement	프로토콜 구현 적합성 문장
PIXIT	Protocol Implementation eXtra Information for Testing	시험을 위한 프로토콜 구현 부가 정보
PTC	Parallel Test Component	병렬 시험 컴퍼넌트
TC	Test Component	시험 컴퍼넌트
TCP	Test Coordination Procedure	시험 조정 절차
SUT	System Under Test	시험 대상 시스템

### 참고문헌

[1] CCITT, Functional Specification and Description Language(SDL), Recommendation Z.100, 1988.

[2] ISO, Estelle : A Formal Description Technique Based on a Extended State Transition Model, ISO 9074, Jul.1989.

[3] ISO, Information Processing Systems-Open Systems Interconnection-LOTOS-A Formal Description Technique based on the Temporal Ordering of Observational Behaviour, ISO 8807, 1989.

[4] G.Berthelot, R.Terrat, "Petri nets theory for the correctness of protocols", IEEE Trans. Commun., Vol.30, No.12, pp.2497-2505, 1982.

[5] A. Diller, Z: An Introduction to Formal Methods, John Wiles & Sons, 1990.

[6] K.Futatsugi, J.A. Goguen and et al, "Principles of OBJ2, Proc. 12th ACM Symp. on Principles of Programming Languages, pp.52-66. Jan. 1985.

[7] CCITT, Recommendation Z.200: CCITT High Level Language(CHILL), ITU, 1989.

[8] ISO, Information technology-OSI-Abstract Syntax Notation(ASN.1), ISO/IEC 8824, 1988.

[9] ISO, Information processing systems-OSI- OSI Conformance Testing Methodology and Framework-ISO 9646-1-7, 1994.

[10] T. Walter and B.Platner, "An operational semantics for concurrent TTCN", IWPTS'92, pp. 101-114, 1992.

[11] ITEX 3.02, User's Guide, 1994.

[12] A.Tang, S. Scoggins, Open Networking with OSI, PTR Prentice-Hall, 1992.



### 김기영

1988년 전남대학교 전산통계학과 졸업  
 1993년 전남대학교 대학원 전산통계학과 졸업(이학석사)  
 1988년-현재 : 한국전자통신 연구소 초고속망 연구실 선임 연구원

관심분야 : 프로토콜 시험 기술, Concurrent Engineering, 멀티미디어 통신



### 하수철

1981년 2월 홍익대학교 전자계산학과 졸업  
 1986년 홍익대학교 대학원 전자계산학과 졸업(이학석사)  
 1990년 홍익대학교 대학원 전자계산학과 졸업(이학박사)

1981년-1984년 Army Logistic Command, System Analyst  
 1992년-1993년 The Florida Univ. Dept. of Computer Science, 객원 교수 / The Univ. of Texas, Dept. of Computer Science Engineering, 객원 교수

1987년- 현재 : 대전대학교 컴퓨터공학과 부교수  
 관심분야 : 소프트웨어 공학, 객체 지향화, 시각 프로그래밍, 멀티미디어, 프로그래밍 언어