

지능망 교환기에 대한 INAP 적합성 시험 스위트 개발 및 검증

도 현 숙[†] · 배 성 용[†] · 김 상 기[†]

요 약

본 논문에서는 형식 기법을 이용하여 차세대 지능망 응용 프로토콜(INAP: Intelligent Network Application Protocol) 적합성 시험 스위트를 생성하는 방법 및 IUT(Implementation Under Test)시뮬레이터를 구축하여 시험을 수행함으로써 시험 스위트를 검증하는 방법에 대해 기술한다. SDL(Specification and Description Language)과 같은 형식 언어를 사용하여 INAP FSM(Finite State Machine)을 모델링하고 MSC(Message Sequence Chart)로 시험 목적을 기술한다. 기술된 FSM모델과 시험 목적을 검증하기 위해 모의 시험을 거치며, 검증이 완료된 후 시험 스위트로 변환이 된다. 형식 기법을 이용하여 INAP규격을 정확하게 기술할 수 있을 뿐 아니라 시험 스위트를 자동으로 생성함으로써 시간과 비용을 절감할 수 있다. 또한 생성된 시험 스위트를 시험기에 탑재하여 IUT시뮬레이터를 대상으로 시험을 수행함으로써 시험 스위트를 검증할 수 있는 방안을 제시하였다.

AIN Protocol Conformance Test Suite Generation Using Formal Methods

Hyunsook Do[†] · Seongyong Bae[†] · Sangki Kim[†]

ABSTRACT

The objective of this paper is to present efforts to generate Intelligent Network Application Protocol (INAP) conformance test suite using formal method and to verify the test suite using IUT(Implementation Under Test) simulator. We describe the processes of the INAP FSM(Finite State Machine) modeling by SDL(Specification and Description Language) language and test purpose description by MSC(Message Sequence Chart). The simulation for the INAP FSM model and test purposes is performed to generate test suite. We can correctly describe the INAP specification using formal description technique, and reduce time and costs by generating test suite automatically. And also we implemented the IUT simulator to verify the generated test suite.

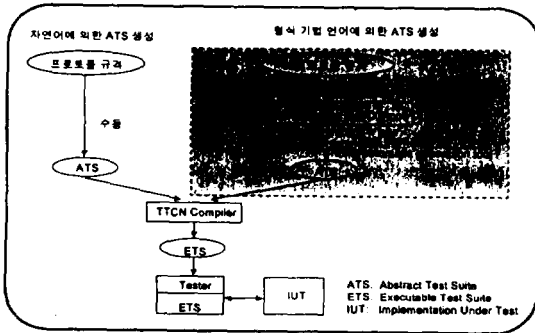
1. 서 론

적합성 시험은 시험자가 시험 대상 구현 제품에 대해 시험 스위트(ATS: Abstract Test Suite)를 작성한 뒤

이에 적절한 적합성 시험 환경을 구성하여 시험하는 것을 말한다. 일반적인 적합성 시험 방법들은 장치 구성, 규모 등에 있어 차이는 있지만 시험을 하기 위한 절차는 유사하다. 적합성 시험을 위한 ATS는 해당 프로토콜의 기능 또는 요소에 대한 적합성 여부를 판단하기 위해 시험 대상 구현 제품에 대해 발생시킬 사건, 예상 결과, 예상 결과와 관찰된 결과에 따른 적합성

[†] 정 회 원: 한국전자통신연구소 지능망연구부
논문접수: 1997년 10월 27일, 심사완료: 1998년 1월 16일

판정 기준 등을 기술한 시험 케이스(Test Case)의 집합으로 구성된다[2].



(그림 1) 자연어와 형식 기법 언어에 의한 ATS 생성 방법
(Fig. 1) Overall process of conformance testing

ATS를 기술하기 위해 ISO에서는 TTCN(Tree & Tabular Combined Notation)이라는 표준화된 언어를 사용하고 있다. ATS 생성 방법에는 자연어를 사용하는 방법과 SDL, LOTOS, Estelle과 같은 형식 기술 언어를 사용하는 방법이 있다. 자연어를 사용하는 방법은 프로토콜 규격으로부터 수동적인 방법으로 ATS를 작성한다. 이는 많은 시간과 노력이 소요되며 완전한 ATS를 생성할 수 없다는 단점이 있다. ATS 수동 생성의 이러한 단점을 극복하기 위하여 형식 기술 언어를 이용하여 프로토콜 명세화 및 검증용 하고, 시험 경우를 자동으로 생성하는 방법이 하나의 대안으로 대두되고 있으며 이에 대한 연구도 활발하게 이루어지고 있다. 이 방법은 자연어를 사용하는 방법보다 소요 시간 및 정확도 등이 개선될 수 있으나 이는 프로토콜의 복잡도에 좌우된다. (그림 1)은 이들 두 가지 방법에 의하여 ATS를 생성하는 과정을 보이고 있다.

일반적으로 ATS는 시험 행위(Test Behaviour Part)를 기술하는 부분과 시험 행위가 필요로 하는 데이터를 기술하는 부분으로 구성된다. 본 논문에서는 INAP ATS를 생성하는데 있어, 프로토콜 규격을 형식 기술 언어로 기술하여 시험 행위를 자동으로 생성하고 데이터 부분은 TTCN 편집기를 이용하여 수동적인 방법으로 기술하였다. 이는 프로토콜 규격을 모델링하는 형식 기술 언어가 INAP에서 데이터 정의 표준으로 사용하고 있는 ASN.1 표기를 완전하게 지원할 수

없기 때문이다.

본 논문에서는 형식 기술 언어로는 ObjectGEODE의 SDL편집기와 MSC 편집기를 사용하였으며, 데이터를 기술하는 도구로는 ITEX라는 TTCN편집기를 사용하였다. 형식 기법을 도입함으로써 ATS의 수정 및 유지 보수가 용이해지며, INAP의 기능 확장시 필요한 ATS 확장도 FSM모델을 재사용함으로써 적은 노력으로 쉽게 할 수 있다는 장점이 있다. 또한 생성된 ATS가 INAP규격에 맞게 제대로 개발되었는지에 대한 검증이 필요한데 본 논문에서는 IUT시뮬레이터를 구현하여 생성된 ATS를 적용하여 시험을 함으로써 그 문제를 해결하였다.

본 논문의 제2장에서는 차세대 지능망의 구조와 차세대 지능망 응용 프로토콜에 대한 설명을 하고, 제3장에서는 SSP시험을 위한 시험 형상 및 방법에 관해 기술한다. 제4장에서는 SSP시험에 필요한 시험 스위트를 생성하는 절차, 즉 시험 행위를 자동화된 방법으로 생성하는 절차와 데이터를 수동적으로 기술하는 방법에 관해 기술하며 제5장에서는 생성한 ATS를 IUT시뮬레이터에 적용하여 시험을 한 경험에 관해서 언급한다. 마지막으로, 제6장에서는 결론 및 향후 연구 방향에 대해 논한다.

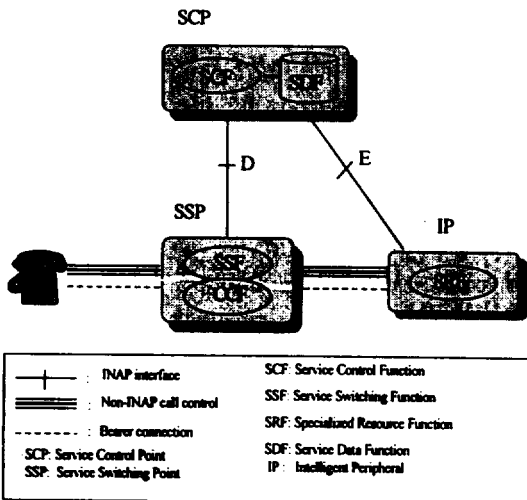
2. 차세대 지능망 구조와 차세대 지능망 응용 프로토콜

초기 지능망(IN: Intelligent Network)에서는 서비스 제어 기능과 호 처리 기능이 분리되어 있지 않아 새로운 서비스의 추가나 수정이 어려웠다. 또한 물리적 장치간의 프로토콜이 제공할 서비스에 종속되어 새로운 서비스를 도입할 때 관련된 소프트웨어의 추가 개발이 요구되므로 다양한 서비스 도입에 한계가 있고 전화망만을 대상으로 하는 단점이 있었다. 이러한 단점을 극복하기 위해서 국제 표준화 기구인 ITU-T (International Telecommunication Union Telecommunications Standardization Sector)에서는 재사용 가능한 모듈 구조를 가진 차세대 지능망(AIN: Advanced Intelligent Network)에 대한 연구를 활발히 하고 있다. 차세대 지능망에서는 서비스나 망에 독립적인 망 기능 구조를 정의하며, 이러한 기능들은 물리적으로 분산되어 망 운용자가 망의 특정 기능을 수행하는 플

플랫폼을 사용하여 전체망을 구축하는데 융통성을 제공한다. 기능의 분산은 결국 이들간의 통신을 필요로 한다. 즉, 망 기능들은 서비스를 제공하기 위해 상호 신호를 위한 프로토콜을 사용한다.

차세대 지능망에서는 서비스와 하부 통신망에 투명한 범용 지능망 응용 프로토콜(INAP: Intelligent Network Application Protocol)을 지향하고 있어 전화망과 ISDN(Integrated Services Digital Network), B-ISDN(Broadband-ISDN) 및 FPLMTS(Future Public Land Mobile Telecommunication System)를 기반망으로 다양한 서비스를 동시에 수용할 수 있는 통합 플랫폼의 구축이 기대되고 있다.

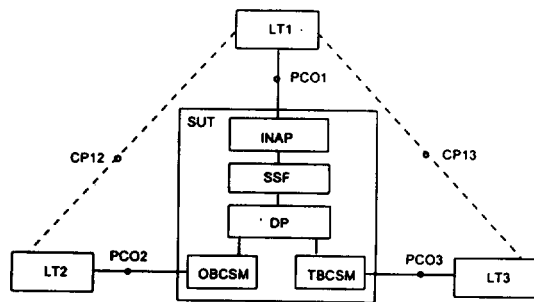
(그림 2)는 차세대 지능망 기능 모델과 INAP 인터페이스를 보여준다[9]. 지능망 서비스는 기능 실체들간의 상호 동작에 의해 이루어지며, 이들간의 상호 동작에는 서로 메시지를 주고 받기 위한 신호가 필요하다. INAP은 지능망의 기능 실체들간의 상호 동작에 필요한 신호를 지원하기 위한 응용 프로토콜이며, IN CS-1 INAP은 SCF-SSF, SCF-SRF 그리고 SCF-SDF기능 실체들간의 응용 계층 인터페이스를 지원한다. 본 논문에서는 서비스 제어 기능내에 서비스 데이터를 포함하고 있는 형태를 가정하고 있으므로 SCF-SDF간의 인터페이스는 고려하지 않는다.



(그림 2) 차세대 지능망 구조
(Fig. 2) AIN architecture

3. SSP 시험을 위한 시험 형상

SSP적합성 시험 방법으로는 원격 시험 방법을 채택하였다. 여러 가지 시험 방법 중에서 원격 시험 방법이 IUT에게 영향을 가장 적게 미치기 때문이다[2]. (그림 3)은 SSP시험 형상을 나타낸다. SUT(System Under Test)는 SSP가 되며, SSP는 SSF를 포함하고 있는 물리적인 장치이다.



(그림 3) SSP 시험 형상
(Fig. 3) Test configuration for testing SSP

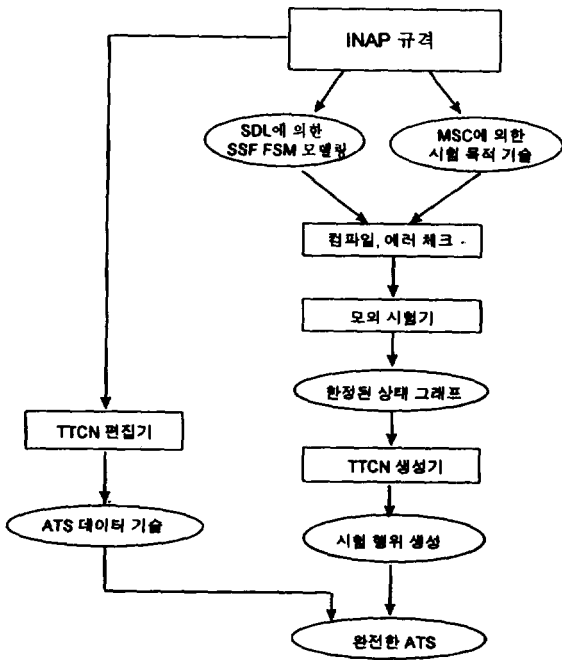
(그림 3)을 보면, 3개의 하위 시험기(LT: Lower Test)가 사용된다. LT1은 주(main) LT이며, LT2와 LT3은 병렬(parallel) LT이다. 병렬LT의 기능은 시험을 위해 필요한 INAP 절차를 IUT에서 발생하도록 하거나 비INAP(non-INAP)인터페이스를 통해 어떤 관찰을 할 수 있도록 한다. PCO1(Point of Control and Observation)은 LT1과 SUT가 INAP인터페이스를 통해 주고 받는 메시지를 제어하고 관찰하는 포인트이다. LT1과 SUT는 이 PCO1을 통해서만 메시지를 주고 받을 수 있다. PCO2와 PCO3은 각각 발신과 착신 신호를 위한 비INAP인터페이스에 존재한다. LT1은 SCP의 기능을 수행하며, LT2와 LT3은 비INAP인터페이스를 통해 발생하는 임의의 네트워크 사건의 기능을 수행한다. CP12(Coordination Point)는 LT1과 LT2간에 주고 받는 메시지를 조정하는 역할을 수행하며, CP13은 LT1과 LT3간의 메시지 조정을 수행한다.

4. SSP ATS 생성

(그림 4)는 형식 기법을 이용하여 INAP규격으로부

터 ATS를 생성하는 절차를 보이고 있는데, 크게 시험 행위를 생성하는 절차와 데이터를 기술하는 절차로 나뉘어진다. 서론에서 언급한 바와 같이 형식 기법 도구로는 Verilog사의 ObjectGEODE를 사용하였고, 데이터를 기술하기 위한 TTCN 편집기로는 Telelogic의 ITEX를 사용하였다[13, 14, 15, 16, 17, 18].

ObjectGEODE는 SDL 편집기, MSC 편집기, 모의 시험기, TTCN생성기를 제공한다. SDL과 MSC편집기를 이용하여 SSF FSM모델과 시험 목적을 작성하였으며, 모의시험기를 통하여 이들의 에러를 검출하고 검증을 하였다.



(그림 4) ATS 생성 절차

(Fig. 4) Abstract Test Suite generation processes

TTCN생성기는 SDL과 MSC로 작성된 시험 규격을 TTCN형태로 된 시험 행위로 변환하여 준다. SDL 편집기 및 MSC편집기, TTCN생성기에서는 ATS의 데이터 부분을 부분적으로 생성할 수 있지만, INAP에서 데이터 정의의 표준으로 사용하고 있는 ASN.1 표기를 완전하게 지원할 수 없기 때문에 ATS의 데이터 부분은 ITEX라는 TTCN 편집기를 사용하여 수동적

으로 작성한다.

4.1 SSF FSM 모델링

SSF는 호 관련, 비호 관련 오퍼레이션들을 처리한다. 비호 관련 오퍼레이션은 SSME(SSF Management Entity) FSM에 의해 수행되고, 호 관련 오퍼레이션은 SSF FSM에 의해 수행된다. 이들 FSM은 서로에게 영향을 미치지 않기 때문에 독립적으로 모델링될 수 있다. <표 1>은 두 FSM에 관련된 오퍼레이션을 보이고 있다.

(그림 5)와 (그림 6)은 각각 SSME FSM과 SSF FSM 상태 천이도를 나타낸다. 이 논문에서는 SSME FSM을 모델링하는 과정에 초점을 맞추도록 한다. SSF FSM을 모델링하는 방법은 SSME FSM을 모델링하는 것과 동일하다.

SDL로 FSM모델링을 할 때 우선, 구현하고자 하는 시스템의 정적인 측면과 동적인 측면을 시스템 구조도와 상호연결 다이어그램을 이용하여 기술한다. 시스템 구조도는 시스템을 구성하는 컴포넌트(system, block, process, procedure, service, data declaration)들의 계층적인 구조를 보여 주며, 시스템의 정적 표현을 보여 준다.

상호연결 다이어그램은 시스템 구조도의 각 레벨을 구성하는 컴포넌트들간과 외부환경과 주고 받는 메시지의 흐름을 보여주며, 시스템의 정적인 측면과 동적인 측면을 함께 보여 준다. 상호연결 다이어그램에서 정의하는 메시지와 컴포넌트들간의 연결 관계는 프로세서들이 주고 받는 시그널 등을 정의할 때 필요한 요소들이다.

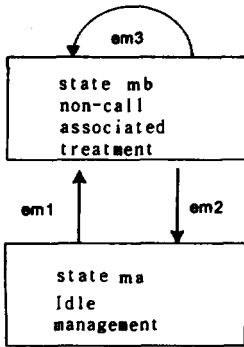
(그림 7)은 SSME시스템 구조도를 나타낸다. 이 시스템 구조도는 TCAP 메시지 처리(message treatment) 블록과 SSME 오퍼레이션 처리(operation treatment) 블록으로 나뉘어진다. 각 블록은 여러 개의 프로세서들로 구성되어 있다.

(그림 8)은 블록과 블록, 블록과 외부 환경간에 주고 받는 메시지를 나타내는 상호연결 다이어그램을 보여준다. 각 블록의 기능은 다음과 같다.

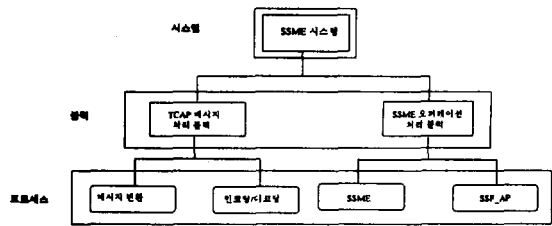
- TCAP 메시지 처리 블록: INAP은 TCAP(Transaction Capability Application Part)을 통하여 다른 상대 실체와 통신한다. INAP은 TC프리미티브로 TCAP과 메시지를 주고 받는다. TCAP에

〈표 1〉 SSF 오퍼레이션들
 〈Table 1〉 SSF operations

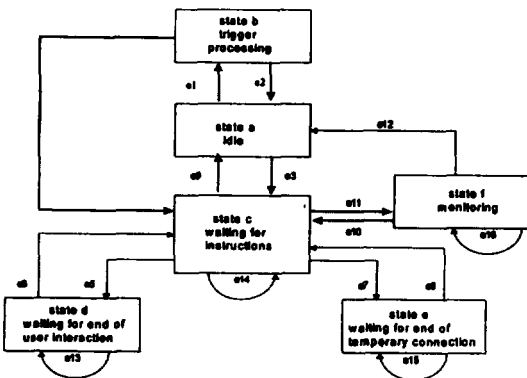
SSME FSM	SSF FSM	
ActivateServiceFiltering ActivityTest CallGap CancelStatusReportRequest RequestCurrentStatusReport RequestEveryStatusChangeReport RequestFirstStatusMatchReport ServiceFilteringResponse StatusReport	ApplyCharging ApplyChargingReport AssistRequestInstructions CallInformationRequest CallInformationReport Cancel CollectInformation Connect ConnectToResource Continue DisconnectForwardConnection EstablishTemporaryConnection EventNotificationCharging	EventReportBCSM FurnishChargingInformation InitialDP InitiateCallAttempt ReleaseCall RequestNotificationChargingEvent RequestReportBCSMEvent ResetTimer SelectFacility SendChargingInformation PlayAnnouncement, PromptAndCollectUserInfo, Information, SpecializedResourceReport



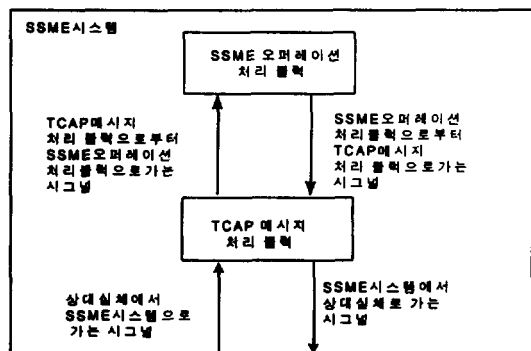
(그림 5) SSME FSM 상태 천이도
 (Fig. 5) SSME FSM state diagram



(그림 7) SSME 시스템 구조도
 (Fig. 7) SSME system hierarchy diagram



(그림 6) SSF FSM 상태 천이도
 (Fig. 6) SSF FSM state diagram



(그림 8) SSME 시스템 상호연결 다이어그램
 (Fig. 8) SSME system interconnection diagram

시지 처리 블록은 INAP과 TCAP간에 주고 받는 메시지를 상대가 이해할 수 있는 형태로 변환해주는 일을 한다.

- SSME 오퍼레이션 처리 블록: 이 블록에서는 (그림 5)의 SSME FSM 상태 천이도에 따른 SSME 오퍼레이션 동작을 모델링한다.

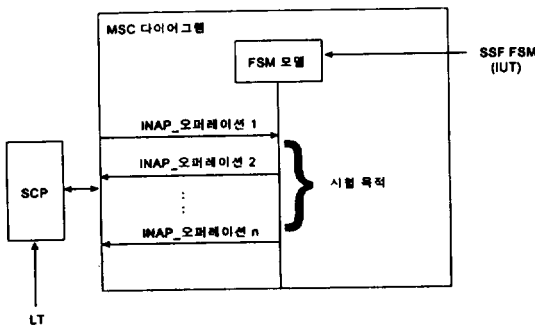
4.2 시험 목적 기술

시험 목적이란 프로토콜 규격에 맞게 IUT가 구현되었는지를 검증하기 위한 테스트 시나리오이다. 본 논문에서는 (그림 9)와 같이 시험 목적을 형식 기술 언어인 MSC로 작성하였다. (그림 9)에서, FSM모델과 SCP사이의 일련의 오퍼레이션들이 시험 목적이 된다. SDL로 기술된 FSM모델이 IUT가 되고, FSM 모델과 통신하는 외부 환경이 LT가 해당된다. 이 경우에 SSF는 IUT에 해당되고 SCP는 LT에 해당된다.

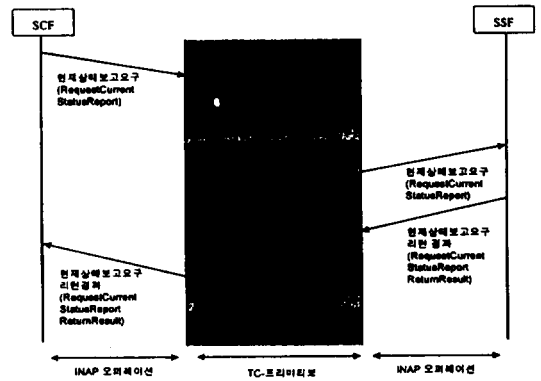
시험 목적은 IUT에 발생될 수 있는 대부분의 상황, 즉 성공적인 경우와 실패의 경우 모두를 기술한다. 하나의 MSC다이아그램은 하나의 시험 목적을 기술할 수 있다. (그림 10)은 SSF가 SCF로부터 전달받은 현재상태보고요구(RequestCurrentStatusReport) 오퍼레이션을 제대로 수행하는지를 확인하는 시험 목적을 나타낸다. 이 시험 목적은 다음과 같이 기술될 수 있다.

“유휴 상태에 있는 IUT가 필수 파라미터를 갖는 현재상태보고요구 오퍼레이션을 받으면, IUT는 현재 상태보고요구 오퍼레이션의 결과를 보내고 다이얼로그를 종료한다.”

현재상태보고요구 오퍼레이션은 SCF가 특정 자원의 현재 상태(유휴 혹은 무휴)를 관찰하기 위하여 SSF



(그림 9) 시험 목적 생성
(Fig. 9) Test Purpose generation



(그림 10) 현재상태보고요구 오퍼레이션 처리 절차 검증을 위한 시험 목적의 예
(Fig. 10) The example of test purpose for RequestCurrent-StatusReport operation

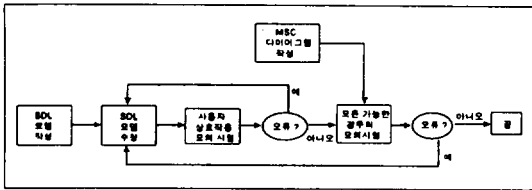
에게 요구하는데 사용된다. SCF는 TCAP을 통하여 SSF에게 현재상태보고요구 오퍼레이션을 보낸다. TCAP이 이 오퍼레이션을 받으면 TC_BEGIN request 프리미티브로 다이얼로그를 시작하고 TC_INVOKE request 프리미티브로 PDU를 전송한다. SSF는 이에 대한 결과를 전송하고 다이얼로그는 종료된다.

4.3 모의 시험

모의 시험에서는 FSM이 에러 없이 제대로 모델링되었는지, 또 요구 사항을 만족하는지를 시험한다. 먼저, 시스템 요구 사항에 맞게 작성된 SDL, MSC모델을 문법 오류나 타이핑 오류 등을 검출함으로써 정적인 검증 과정을 거친다. 정적 검증 과정이 완료되면, 시스템의 논리 오류가 있는지를 검사하고 시스템이 설계 의도에 맞게 설계되었는지를 검증한다.

ObjectGEODE에서 제공되는 모의 시험 방법에는 사용자 상호작용, 무작위, 모든 가능한 경우의 세 가지가 있다. 본 논문에서는 사용자 상호작용과 모든 가능한 경우의 모의 시험 방법을 이용하였다.

(그림 11)은 SDL과 MSC모델의 모의 시험 과정을 보이고 있다. 사용자 상호작용 모의 시험을 함으로써 그 모델이 제대로 동작하는지를 확인할 수 있는데, 대부분의 문제점은 이 단계에서 검출이 가능하다. 오류가 발견되면 SDL모델은 수정이 되며, 다시 사용자 상호작용 모의 시험을 반복적으로 거치게 된다. 이



(그림 11) 모의 시험 과정
(Fig. 11) Simulation phases

단계에서 더 이상 여려가 검증되지 않으면 MSC모델과 함께 SDL모델은 모든 가능한 경우의 모의 시험 과정으로 들어간다. 이때, 모든 가능한 경우란 MSC에서 기술된 시나리오에 관한 모든 가능한 경우를 의미한다. 즉, SDL모델에서는 MSC에 한정된 일부가 시험이 되는 것이다. 여기서 오류가 발견되면 다시 SDL모델은 수정되어 두 가지 모의 시험 과정을 거치며, 그렇지 않으면 모의 시험은 종료된다.

4.4 TTCN생성과 ATS 데이터 정의

TTCN생성기는 SDL로 작성된 FSM모델과 MSC로 작성된 시험 목적의 모든 가능한 경우의 모의 시험 결과를 TTCN형식으로 변환해 준다[17]. (그림 12)와 13은 4.2절에서 언급되었던 현재상태보고요구 오픈레이션을 테스트하는데 사용될 ATS의 행위 부분을 각각 TTCN.MP(Machine Processible) 형태와 TTCN.GR(Graphical Representation) 형태를 나타내 보이고 있다. 이들은 형태만 다를 뿐 내용은 동일하다.

```

SDynTestPart
StartCase
  Begin_TestCase
  StartCaseId : tcase_system_1
  StartCroupId
  StartPurpose //
  StartDefaultRef
  Comment // ECT/STEE Observer generation V1.1
  SBehaviourDescription
  SLabelId
  SLine (0) ttc_invoke_req
  SCraf ttc_invoke_req_1
  SVerdictID
  SComment //
  Send_BehaviourLine
  SLabelId
  SLine (1) ttc_begin_req START T()
  SCraf ttc_begin_req_1
  SVerdictID
  SComment //
  Send_BehaviourLine
  SLabelId
  SLine (2) ttc_and_cancel T, START T()
  SCraf ttc_and_cancel_1
  SVerdictID
  SComment //
  Send_BehaviourLine
  SLabelId
  SLine (3) ttc_result_1_cancel T, START T()
  SCraf ttc_result_1_cancel_1
  SVerdictID
  SComment //
  End_BehaviourLine
  EndTestPart
  
```

(그림 12) TTCN.MP형식으로 표기된 ATS의 행위 부분
(Fig. 12) ATS formed with TTCN.MP for dynamic behaviour part

Test Case Name : tcase_system_1					
Group :					
Purpose :					
Default :					
Comments : ECT/STEE Observer generation V1.1					
Nr	Label	Behaviour Description	Constraints Ref	Verdict	Comments
1		ttc_invoke_req			
2		ttc_begin_req START T()	ttc_begin_req_1		
3		ttc_and_cancel T, START T()	ttc_and_cancel_1		
4		ttc_result_1_cancel T	ttc_result_1_cancel_1		
5		ttc_and_req	ttc_and_req_1	P	Final
6		POTHEWISE			
7		TIMEOUT T			
8		POTHEWISE CANCEL T			
9		POTHEWISE			
10		POTHEWISE			

(그림 13) TTCN.GR형식으로 표기된 ATS의 행위 부분
(Fig. 13) ATS formed with TTCN.GR for dynamic behaviour part

Test Case Name : ttc_req_1					
Group : SSME/ASB/behaviour/					
Purpose :					
Configuration :					
Default :					
Comments : ECT/STEE Observer generation V1.1					
Behaviour Ref :					
Description : Valid test case for RequestCurrentStatusReport with all parameters					
Nr	Label	Behaviour Description	Constraints Ref	Verdict	Comments
1		LITC_INVOKE_req	RCGR_jw_1(TCV_reqID_1, TCV_jwID_1)		
2		LITC_BEGIN_req	TC_req_jw_1(RTC_reqID_1, TCV_reqID_1)		
3		LITC_END_req	TC_req_jw_1(TCV_reqID_1, TCV_jwID_1)		
4		LITC_RESULT_1_req	RCGR_jw_1(TCV_reqID_1, TCV_jwID_1)		
5		LITC_END_req	TC_req_jw_1(TCV_reqID_1, TCV_jwID_1)		

(그림 14) ATS의 행위 기술 부분
(Fig. 14) Behaviour description part

ASP Constraint Declaration		
Constraint Name : RCGR_jw_1 (diag: DialogId, Invok: InvokIDType)		
ASP Type : TC_INVOKE_req		
Definition Part : TC_INVOKE_req		
Comments : To invoke an RequestCurrentStatusReport.		
Parameter Name	Parameter Value	Comments
DialogId	diag	TCV_reqID_1
Class	RCGR_jw_1@	TCV_jwID_1
ComponentInvoke	Req_jw_1@reqid	TCV_reqID_1
Timeout	TSPC_Tcar	From FCS

(그림 15) ASP 제약 조건 정의부
(Fig. 15) ASP constraint declaration part

ASN.1 PDU Constraint Declaration	
Constraint Name : Req_jw_1@reqid: InvokIDType	
PDU Type : RCGR_invoke	
Definition Path :	
Receiving Role Name :	
Receiving Verdicts :	
Comments :	
Constraint Value	Constraint Value
InvokID	--TCV_jwID_1
ReceivingRole	RCGR_ConstVal.
Timeout	TSPK_reqLimit

(그림 16) ASN.1 PDU 제약 조건 정의부
(Fig. 16) ASN.1 PDU constraint declaration part

SDL편집기 및 MSC편집기, TTCN생성기에서는 ATS의 데이터 부분을 부분적으로 생성할 수 있지만, INAP에서 데이터 정의 표준으로 사용하고 있는 ASN.1 표기를 완전하게 지원할 수 없기 때문에 ATS의 데이터 부분은 ITEX라는 TTCN 편집기를 사용하여 수동적으로 작성한다. 즉, TTCN생성기에 의해 생성된 ATS의 행위 부분을 ITEX에서 읽을 수 있는 형식으로 변환한 후에 각 행위에서 필요로 하는 데이터를 INAP규격에 맞게 정의한다.

(그림 14)는 TTCN생성기로부터 생성된 ATS의 행위 부분을 ITEX에서 읽을 수 있는 형식으로 변환한 것이다. 각 행위(behaviour)들은 제약 조건 정의부(constraints declaration)를 참조하고 있다. 제약 조건은 같은 행에 기술되어 있는 행위가 만족해야 할 특정 조건을 명시하고 있다. (그림 14)에서 참조하고 있는 제약 조건은 (그림 15)에서, (그림 15)에서 다시 참조하고 있는 ASN.1 PDU 제약 조건은 (그림 16)에서 각각 보이고 있다. 즉, 임의의 행에 기술된 사건은 그 행에서 참조하고 있는 조건 정의부에 정의되어 있는 값이 만족될 때만 발생할 수 있다는 것을 의미한다.

5. SSP 적합성 시험 환경 및 SSP시뮬레이터를 이용한 모의 시험

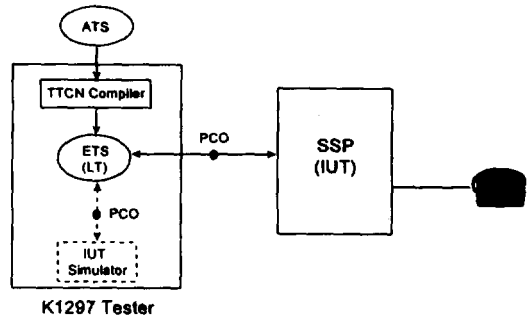
SSP적합성 시험을 하기 위해서는 적합성 시험 장비가 필요한데, 여기서는 Siemens K1297 적합성 시험 장비를 사용하였다[19]. ATS를 실행 가능한 형태로 바꾸기 위해서는 시험 장비 환경에 맞게 컴파일 되어야 한다. 4절에서 생성된 SSP ATS를 K1297 TTCN compiler를 이용하여 ETS(Executable Test Suite)로 컴파일하였다. 컴파일하는 과정 중에 ATS오류가 많이 발견되었는데, 이들 오류는 K1297환경과 맞지 않아 발생하는 것이 대부분이었다. 즉, 표준 ATS라 하더라도 실제 탑재되는 시험 장비의 특성에 맞게 수정이 필요한 부분이 발생할 수 있다.

현재로서는 시험 대상이 되는 시스템(IUT)이 없기 때문에, IUT시뮬레이터를 구현하여 생성한 ATS를 적용하여 시험하였다. (그림 17)은 K1297시험기를 이용한 SSP시험 환경을 보이고 있다. 점선으로 표시된 부분은 실제 IUT 대신에 IUT시뮬레이터를 사용한 시험 형상을 나타낸다. IUT시뮬레이터는 SSP 기능을

수행하며, 시험기는 SSP ATS로 IUT시뮬레이터를 시험한다.

SSP ATS는 총 515개의 시험 케이스를 포함한다. 이들은 정상적인 경우를 시험하는 케이스들, 오류가 있는 데이터를 포함하거나 올바르지 않은 동작을 시험하는 케이스들을 모두 포함한다. 515개의 시험 케이스들로 IUT시뮬레이터를 시험하는 과정에서도 ATS오류가 발견되었고, 이에 대한 ATS 보완이 이루어졌다. 실제 IUT를 시험하기 이전에 IUT시뮬레이터를 구축하여 생성한 ATS를 적용하여 시험을 하는 것은 ATS를 검증할 수 있는 좋은 수단이 된다. 어느 정도로 완벽하게 검증이 될지는 IUT시뮬레이터가 어느 정도의 기능을 갖도록 구축했느냐에 따라 달라진다고 볼 수 있을 정도로 IUT시뮬레이터의 역할은 중요하다고 볼 수 있다.

현재는 IUT시뮬레이터를 이용하여 시험을 하였는데, 향후에는 실제 IUT를 대상으로 시험을 할 계획이다. (그림 17)의 K1297과 SSP간의 실선이 그 형상을 나타내고 있다.



(그림 17) SSP 적합성 시험 환경
(Fig. 17) SSP conformance testing environment

6. 결 론

본 논문에서는 형식 기법을 이용하여 차세대 지능망 응용 프로토콜(INAP: Intelligent Network Application Protocol) 적합성 시험 시험 스위트를 생성하는 방법에 대해 기술하였다. 형식 기법은 형식 기술 언어를 사용하여 프로토콜 규격을 정확하게 기술할

수 있고, 시험 스위트를 자동으로 생성함으로써 시간과 비용을 절감할 수 있다는 장점이 있다. 형식 기법 도구로는 SDL과 MSC편집기를 사용하였다. 이 편집기들은 프로토콜 규격의 제어 흐름(control flow)을 모델링하기에 적합한 좋은 도구이지만 프로토콜의 데이터 ASN.1형식을 완전하게 지원을 하지 못한다. 이러한 이유 때문에 데이터 부분은 수동적인 방법으로 기술하였는데, 이 작업은 많은 시간이 소요되었다.

본 논문을 통하여 SSP ATS를 생성하는 방법과 IUT시뮬레이터를 이용하여 적합성 시험을 한 경험에 대해 언급하였다. 이러한 적합성 시험을 통해 ATS 검증 작업을 할 수 있었고 향후 실제 IUT를 대상으로 시험을 할 때 발생할 수 있는 문제점들을 예측할 수 있었다.

현재는 IN CS-1대상으로 INAP SSP ATS를 생성하였는데, 앞으로 지능망이 CS-2/3로 진화함에 따라 INAP도 이를 지원할 수 있어야 하고 이를 시험할 수 있는 ATS도 추가적으로 개발되어야 한다. 형식 기법에서는 이미 모델링된 FSM모델을 재사용함으로써 적은 노력으로 ATS의 추가, 수정할 수 있다.

감사의 글

이 논문은 한국통신의 출연으로 수행된 "AIN적합성 시험 기술 연구"의 결과로서, 과제를 원활하게 수행하도록 지원해 주신 한국통신에 감사사를 드립니다.

참 고 문 헌

[1] Behcet Sarikaya, "Conformance Testing: Architectures and Test Sequence", Computer Networks and ISDN Systems 17, pp. 111-126, 1989.
 [2] Keith G. Knightson, OSI Protocol Conformance Testing, McGraw-Hill, 1993.
 [3] M C Bale, "Signaling in the intelligent network", BT Technical Journal, Vol. 13 No. 2, pp. 30-42, April 1995.
 [4] R. Lai, W. Leung, "Industrial and academic protocol testing: the gap and the means of convergence", Computer Networks and ISDN Systems 27, pp. 537-547, 1995.
 [5] S T Vuong, H Janssen, Y Lu, C Mathieson and B Do, "TESTGEN: An environment for protocol

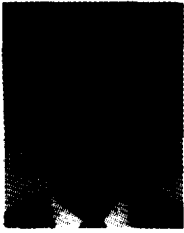
test suite generation and selection", computer communications, Vol. 17, No. 4, pp. 257-270, April 1994.

[6] T Ramalingam, Anindya Das and K Thulasiraman, "On testing and diagnosis of communication protocols based on the FSM model", computer communications, Vol. 18, No. 5, pp. 329-337, May 1995.
 [7] ETSI, IN; Global functional plan for IN Capability Set 1, TCR-TR 016, 1994.
 [8] ETSI, IN; IN CS 1 Core Intelligent Network Application Protocol Part 1: Protocol specification, ETS 300 374-1, 1994.
 [9] ITU-T Revised Recommendations, Interface Recommendation for IN CS-1, Q.1218, 1995.
 [10] ITU-T Baseline Document for IN CS-2, 1995.
 [11] ITU-T SG11 WP4 TD PL/11-11,12 Q.1214, "Distributed Functional Plan for IN CS-1", April 1995.
 [12] ITU-T Recommendation Q.1201, "Principles of Intelligent Network Architecture", 1992.
 [13] An Introduction to SDL and MSC, Verilog, 1995.
 [14] ObjectGEODE Method Guideline, Verilog, 1995.
 [15] GEODE Editor Reference Manual, Verilog, 1995.
 [16] ObjectGEODE Simulator, Verilog, 1995.
 [17] ObjectGEODE TTCgeN User's Guide, Verilog, 1995.
 [18] ITEX User's Guide, Telelogic, 1994.
 [19] K1297 Protocol Tester Basic Manual, Siemens, 1996.

도 현 속

1989년 2월 성신여자대학교 자연과학대학 전산학과 졸업(학사)
 1989년 2월~1991년 3월 한국전자통신연구원 근무
 1991년 4월~1994년 6월 동경공업대학 지능과학전공(석사)

1994년~현재 한국전자통신연구원, 지능망연구부 근무
 관심분야: 지능망, 적합성 시험, IMT-2000



배 성 용

- 1992년 충남대학교 자연과학대학 전산학과 졸업(학사)
- 1994년 충남대학교 대학원 전산학과 졸업(석사)
- 1994년 한국전자통신연구원
- 1994년~현재 연구원, 지능망연구부 근무

관심분야: 지능망, PCS, IMT-2000, 번호이동성, 신경회로망.



김 상 기

- 1981년 서울대학교 산업공학과(공학사)
 - 1983년 서울대학교 대학원 산업공학과(석사)
 - 1984년 한국전자통신연구소 근무
 - 1996년 지능망구조연구실 실장
- 관심분야: 지능망, UPT, TINA