

형식기술기법에 의한 AIN 프로토콜 적합성 시험 계열 생성

김 상 기[†] · 김 성 운^{††} · 정 재 윤^{†††}

요 약

본 논문은 형식 기술 기법에 의한 차세대 지능망의 INAP(Intelligent Network Application Protocol) 프로토콜 적합성 시험 계열 생성을 위해 형식 기술 기법(formal Description Technique)을 사용하여 프로토콜을 명세화한 후 이것으로부터 최소 길이의 최소 비용을 요구하는 시험계열 생성을 위한 최적화 기술에 의한 방법을 제안한다. 제안된 방법을 구현하고 효율성을 증명하기 위해, INAP 프로토콜 SRF(Service Resource Function)의 SRSR(SRF Call State Machine)을 형식 기술 기법 중의 하나인 SDL(System Description Language)로 명세화하여 관련 도구로 I/O FSM(Input/Output Finite State Machine)을 생성하고, 이 참조 모델에 직접 적용하여 시험 계열을 생성하였다. 이렇게 생성된 시험 계열의 길이가 기존의 UIO(Unique Input Output) 방법에 의한 것보다 31%나 개선된 짧고 효율적임을 보였고 또 오류 판단 능력면에서도 훨씬 효과가 있음을 실험적으로 증명하였다.

A Formal Method on Conformance Testing for AIN Protocol Test Generation

Sang Ki Kim[†] · Sung Un Kim^{††} · Jae Yoon Chung^{†††}

ABSTRACT

This paper proposes a formal method on conformance testing for INAP(AIN) test sequence generation by optimization technique. In order to implement and prove the effectiveness of the proposed method, we specify the SRSR of INAP protocol SRF in SDL and generate I/O FSM by using our S/W tool. We generate an optimal test sequence by applying our method to this reference I/O FSM. We prove experimentally that the length of the generated test sequence by our method is more effective and shorter(i.e., 31% improved) than the one generated by UIO method, and estimate that The test coverage space of our test sequence is larger than that of UIO method.

1. 서 론

ISO(International Standard Organization)나 ITU(International Telecommunication Union)의 지능망

프로토콜 표준으로부터 관련 제품 구현에 있어, 적합성 시험은 구현 제품의 프로토콜 표준에 대한 적합성을 검증하는 시험으로 프로토콜 표준으로부터 시험 계열(Test Case)을 생성하여 구현에 적용함으로써 적합성 여부를 판단하는 일련의 과정이다[1].

자연어(영어)로 서술된 프로토콜 국제 표준(ISO, ITU 표준 등)들은 자연어로 서술된 명세(Specification) 자체가 구현자의 해석방법 여부에 따라 다르게 구현될

† 정 회 원: 한국전자통신연구소 지능망 구조 연구실
†† 정 회 원: 부경대학교 정보통신공학과 교수
††† 준 회 원: 부경대학교 정보통신공학과 석사과정
논문접수: 1996년 8월 27일, 심사완료: 1996년 12월 11일

수도 있고, 또 자연어로 된 명세로부터 시험 계열 생성 자체가 수동적이어서 컴퓨터를 이용한 생성 과정의 자동화가 어려워, 복잡한 프로토콜의 경우 효과적인 적합성 시험을 위한 시험 계열 생성에 많은 어려움을 겪어왔다. 재래의 적합성 시험 방법에서는 자연어로 작성된 명세에서부터 하나하나의 시험 항목을 수작업에 의하여 도출하였다. 이러한 방법은 자연어로 된 규격이 갖는 단점과 수작업에 의한 시험항목들의 도출이 갖는 단점을 모두 지닌다. 즉, 자연어로 된 규격은 모호성(ambiguity), 모순성(inconsistency), 불완전성(incompleteness) 등으로 인한 부정확성을 갖는 한편, 수작업 시험 항목 도출 방법의 단점은 시험 스위트, 시험기, 시험등의 품질이 시험 수행자의 능력에 따라 달라지고 수행자의 실수에 의하여 시험 과정에 오류가 포함될 수 있다는 것이다.

이러한 단점들은 형식 기술 기법(FDT: Formal Description Technique)에 의한 규격명세기술에 의하여 극복될 수 있다. 이 접근 방법으로 형식언어의 사용을 통하여 규격 기술의 정확성을 기할 수 있고, 또 형식기술기법에 의해 명세화된 규격으로부터 최적의 시험 스위트 생성과 시험과정의 컴퓨터를 이용한 자동화를 통하여 시험시간과 비용을 감소시킬 수 있다[2].

본 논문에서는 차세대 지능망의 물리실체들간의 상호작용을 지원하기 위한 응용 계층 프로토콜인 INAP 프로토콜을 대상으로, 지능망 서비스 생성자의 관점에서 서비스의 외부 동작의 확인을 위한 서비스 적합성 시험 모델을 제시하고, 제시된 시험 모델을 기반으로 여러가지 엄격한 적합성 시험 기준을 정의한 후 형식 기술 기법을 적용한 적합성 시험 계열 생성 방법에 대해 기술한다.

본 논문의 2장에서는 적합성 시험의 필요성, 개념, 정의 등에 대해서 논하고, 3장은 제한한 최적화 기술에 의한 시험 계열 생성 방법에 관계되는 적합성 관계, 시험 계열, 시험 스위트, 완전 시험 스위트, 시험 생성 방법 등에 대해서 정의한다. 4장은 INAP 프로토콜로부터의 시험 계열 생성을 위해 자연어에 의한 접근 방법과 형식 언어에 의한 접근 방법에 대해 기술하고, 시험 계열 생성을 위한 모델링으로 SRF의 SRSM에 대한 기능 설명과 이것의 SDL 표현으로부터 I/O FSM 생성에 대해 논한다. 또 이러한 I/O FSM에서부터의 UIO 방법에 의한 시험 계열 생성과 최적

화 기술에 의한 시험 계열 생성을 연구하고 두가지 기법의 비교에 대해서도 서술한다. 마지막으로 5장은 간단한 결론과 향후 연구 사항에 대해 정리한다.

2. 적합성 시험 및 용어 정의

프로토콜 적합성 시험의 목적은 어떤 프로토콜의 구현 I(Implementation)가 원래 명세(혹은 표준) S(Specification)에 합당하게 구현되었는지를 시험하는 것으로 통신 프로토콜 제품 구현 과정에서 중요한 역할을 한다. 일반적으로 적합성시험의 정의는 주어진 명세 S를 기초로 하여 생성된 시험 계열(Test cases)로서 구현 I가 명세 S에 대해 프로토콜 행위(Behaviour)와 능력(Capacity)이 일치하는지를 시험하는 것이다.

시험계열생성을 위한 일반적인 많은 방법들은 명세 S가 입출력한정상태머신(I/O FSM: Input/Output Finite State Machine) 형태로 표현된 프로토콜로부터 출발하는데, 일반적으로 I/O FSM은 통신프로토콜의 제어(control) 부분을 모델화하는데 많이 사용되어져 왔다. I/O FSM의 정의는 다음과 같다[3].

정의 2.1 입출력한정상태머신(I/O FSM)

프로토콜 명세의 제어 부분을 나타내는 I/O FSM은 다음의 다섯가지 요소로 구성된다.

$M = \langle S, s_0, I, O, t_r \rangle$ 여기서

- S: 한정된 상태들의 집합(a set of finite states)
- s_0 : 초기 상태(initial state)
- I: 한정된 입력 알파벳(finite input alphabet)
- O: 한정된 출력 알파벳(finite output alphabet)
- t_r : 천이함수(transition function), $t_r \subseteq \{s-i/o \rightarrow s' \mid s' \in S \wedge i \in I \wedge o \in O\}$

다음의 4개 요소(p, a, b, q)는 I/O FSM M의 천이(transition)로 불려지는데 아래와 같이 정의된다.

$$(p, a, b, q) \in S \times I \times O \times t_r$$

위의 정의에서 각 천이는 하나의 입력과 하나 1의 출력에 의해 일어난다.

I/O FSM $M = \langle S, s_0, I, O, t_r \rangle$ 은 방향화된 그래프(Oriented graph) $G = (V, E)$ 로 표기되어질 수 있는데

여기서

$V=S$ 의 스테이트에 해당하는 노드의 집합(a set of nodes)

E =추이함수 t 에 해당하는 이크의 집합(a set of arcs)

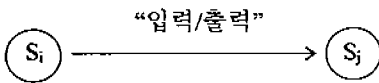
따라서 본 논문에서는 그래프와 I/O FSM, 오토마타는 동의어로 사용하고 각각에 적용되는 용어들을 구별없이 사용한다.

일반적으로 I/O FSM으로 기술된 명세는 통신 프로토콜의 기능과 동작에 기반을 두고 만들어지게 된다. 다음 정의는 I/O FSM으로 기술된 명세 S 에서 시작하는 적합성 시험을 나타낸다.

정의 2.2 시험의 정의

프로토콜의 명세를 나타내는 I/O FSM 모델이 다음과 같이 주어질때 적합성 시험의 정의는 아래의 3 단계로 이루어진다.

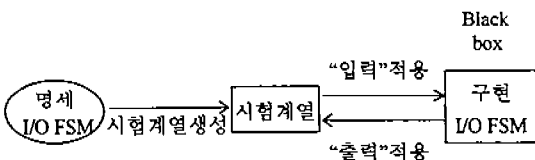
(명세 I/O FSM)



(적합성시험단계)

- (1) 명세 I/O FSM의 스테이트 S_i 에 해당하는 스테이트로 구현 I/O FSM을 위치시킨다.
- (2) 명세 I/O FSM에서 얻어진 시험 계열 "입력"을 구현 I/O FSM에 적용시킨 후 생성되는 "출력"을 판단한다.
- (3) 구현 I/O FSM에서 생성된 출력이 명세 I/O FSM에서 기술된 "출력"과 같은지를 확인하고 도착한 스테이트가 명세 I/O FSM의 그것과 같은지를 검증한다.

정의 2.2의 적합성시험과정을 좀더 쉽게 표현하면 (그림 2-1)와 같이 도식화된다.



(그림 2-1) 적합성시험과정

이러한 I/O FSM 모델로 표현된 프로토콜들로부터의 시험 계열 생성은 I/O FSM 명세 자체가 강하게 접속되어야 하며(strongly connected) 스테이트 수가 최소(minimal)여야 하며, 또 결정형(deterministic)이어야 한다는 가정에서 출발한다. Strongly connected 특성은 Deadlock이나 Livelock으로부터 자유롭게 하며, 최소 스테이트 특성은 최소 비용의 가장 짧은 시험 계열 생성과 결정형 특성은 각 천이에 대한 결정에 필요한 조건이다.

출력 오류와 천이 오류를 걸러내기 위한 (1)에서 (3)의 적합성 시험 단계에서 일반적으로 두가지 문제가 야기된다. 단계 (1)에서 일반적으로 구현 I/O FSM을 원하는 스테이트로 위치시킬 수 없다는 것이고, 두번째로 단계 (3)에서 시험계열의 "입력"을 적용한 후에 원하는 "출력"을 구현 I/O FSM으로부터 얻은 경우라도 구현 I/O FSM이 해당 스테이트에 위치하였는지 알 수 없다는 것이다. 첫번째 문제를 controllability limit이라 하고, 두번째 문제를 observability limit이라 한다.

일반적으로 controllability limit 문제를 해결하기 위해서 초기 상태 s_0 로부터 시작하여 원하는 스테이트까지 가장 짧은 패스(shortest path)를 이용하여 스테이트에 도착한 후 해당 천이를 시험하게 되고, observability limit 문제를 해결하기 위해 시험하는 천이 후에 도착한 스테이트의 유일한 입력/출력 시퀀스를 시험 계열에 포함시켜 적용한 후 구현 I/O FSM의 결과 스테이트 확인하는 방법을 사용한다. 이러한 유일한 입력/출력 시퀀스로 UIO(Unique Input/Output) 시퀀스, DS(Distinguish Sequence) 시퀀스, CS(Characterizing Set) 시퀀스 등을 사용한다.

본 논문에서는 UIO 시퀀스를 사용하여 observability limit 문제를 해결한다. 왜냐하면 DS 시퀀스나, CS 시퀀스 등은 존재를 위한 기본 조건이 모델 I/O FSM 자체가 완전 명세(completely specified) 되어야 하나, UIO 시퀀스는 이러한 요구 조건이 없으며 또, 일반적으로 통신 프로토콜을 모델링하는 I/O FSM은 부분적으로 명세되어 있기 때문이다. 또 DS 시퀀스나 CS 시퀀스는 찾기가 어렵고 존재 여부도 일반적으로 판단하기에 매우 어렵다. 다른 한편으로는 UIO 시퀀스는 대부분의 I/O FSM이 모든 스테이트들에 대해 존재하며 길어도 DS 시퀀스나 CS 시퀀스에 비해 짧기

때문에 결과적으로 짧은 시험 스위트 생성을 보장한다[4].

UIO 시퀀스를 사용하여 I/O FSM으로 표현된 프로토콜 명세로부터 적합성 시험 경우(Test case) 생성은 명세에 나타나 있는 각 천이에 도착 스테이트의 UIO 시퀀스를 concatenation 하여 생성하는데 다음과 같은 식으로 나타낼 수 있다.

$$T_{ij} @ UIO(S_j)$$

위 식에서 T_{ij} 는 시험되어야 할, 즉 명세 I/O FSM의 스테이트 S_i 에서 S_j 로 보내는 천이를 나타내고, $UIO(S_j)$ 는 도착 스테이트의 UIO 시퀀스를 나타낸다. 또 @는 concatenation 심볼이다. 여기서 UIO 시퀀스는 시험되는 천이에 의해 도착된 스테이트가 명세에서 원하는 올바른 S_j 인가를 시험하는데 사용된다.

UIO 시퀀스를 사용하여 I/O FSM 모델로부터의 시험 계열 자동 생성을 위한 여러가지 방법들이 제안되었다. 이중 UIO 방법과 UIO_v 방법은 IUT(Implementation Under Test)에서 신뢰성 있는 reset을 가정한 다중 시험 approach에 속한다. UIO 방법이 100% 완전 오류 영역을 보장하지 못하는데 비해 UIO_v 방법은 완전 오류 영역을 보장한다[5].

3. 최적화 방법

본 논문에서 제안하는 방법은 최적화(optimization) 기술을 사용하여 단일 시험 approach로써 2장에서 언급된 방법들에 비해 IUT에 의해 제공되어야 하는 reset 기능을 요구하지 않는 기술이다. 이 방법의 주된 내용은 구현 I/O FSM 내에 명세 I/O FSM에 명세된 각 천이가 존재하는 지를 시험 하기 위해 명세 I/O FSM으로부터 단일 시험 계열을 생성하는 방법이다.

위에 제안된 방법을 구현하기 위해 적합성 관계로써, 명세 I/O FSM과 구현 I/O FSM 사이에, conf를 다음과 같이 정의한다.

정의 3.1(입력 시퀀스 집합에 대한 스테이트 동치)

I_i 를 구현 I/O FSM의 한 스테이트라 하고 S_j 를 명세 I/O FSM의 하나의 스테이트라고 할 경우, $\forall \Psi(S_j)$ 즉 입력 시퀀스의 집합이라 하면

$$I_i \approx S_j \text{ if } \lambda_1(I_i, P) = \lambda_S(S_j, P), \text{ for } \forall P \in (I)^*$$

여기서, λ 는 I/O FSM의 출력 함수이다.

정의 3.2(Conf)

구현 I/O FSM conf 명세 I/O FSM iff $I_1 \approx \Psi(S_1) S_1$

여기서 I_1 그리고 S_1 은 두 I/O FSM의 초기 스테이트이다.

위의 정의된 conf 관계에 대해 적합성 시험 관련 스위트에 관계되는 일반 개념들은 다음 정의들과 같다.

정의 3.3(시험 계열: Test Case)

시험 계열은 입력들의 한정된 길이의 $\Psi(S_1)$ 에 속하는 스텐스이다.

정의 3.4(시험 스위트: Test Suite)

시험 계열들의 집합을 시험 스위트라 한다.

정의 3.5(Pass of a test suite)

TS를 시험 스위트라 하면, 주어진 구현 I/O FSM (I)는 시험 스위트를 패스한다고 이야기 하면서 다음과 같이 표기한다.

$$I/O \text{ FSM}(I) \text{ pas TS, iff } \lambda_1(I, p) = \lambda_S(S_1, p) \text{ for } \forall s \in TS$$

정의 3.6(완전 시험 스위트: Complete test suite)

주어진 시험 스위트 TS와 명세 I/O FSM(S)에 있어서, 모든 구현 I/O FSM(I)가 다음 조건을 만족할 때 TS는 I/O FSM(S)에 대한 conf 관계에 있어서 완전한 시험 스위트라 한다.

$$I/O \text{ FSM}(I) \text{ conf FSM}(S) \text{ iff } \text{FSM}_1 \text{ pass TS}$$

그러면 다음 절에서는 위에 언급된 적합성 관계에 대한 INAP 프로토콜을 위한 완전 시험 스위트 TS를 자동 생성하는 방법에 대해서 살펴보자.

최적화 기술에 의한 시험 계열 생성 방법에 기초한 기본 개념은 다음과 같다[6][7].

- (1) 명세 I/O FSM 내에서 명세된 각 천이를 위해 시험 subsequence를 생성하는데, 각 시험 subsequence 들은 TUT(Transition Under Test)의 입력 심볼과 이 천이에 의해 도착한 스테이트의 UIO 시퀀스의 concatenation으로 구성된다.
- (2) (1)에서 구성된 모든 시험 subsequence들을 적어도 한번 방문하는 단일 최적 시험 계열을 Rural Chinese Postman(RCP) tour 개념을 사용하여 생성한다.

위의 최적화 방법을 구현하기 위해 주어진 I/O FSM의 모든 스테이트들의 가장 짧은 길이의 UIO 시퀀스를 결정한 후 주어진 I/O FSM 내의 모든 천이에 대해 시험 subsequence를 다음과 같은 형태를 생성한다.

$$T_{SSij} = TR_{ij} @ UIO(s_j)$$

위의 식에서 T_{SSij} 는 스테이트 i에서 스테이트 j로의 천이를 위한 시험 subsequence를 나타내고 TR_{ij} 는 앞에 언급한 TUT(Transition Under Test)이며, @ concatenation 심볼을 나타내고, $UIO(s_j)$ 는 TR_{ij} 에 의해 도착하는 천이의 UIO 시퀀스를 나타낸다.

모든 구해진 시험 subsequence를 원래의 I/O FSM (여기서부터는 reference 그래프라고 표기한다) 즉, reference 그래프 $G(V, E)$ 에 새로 첨가하면 새로운 그래프 $G'=(V', E')$ 가 생성되는데 G' 는 다음과 같다.

$$G'=(V', E'), V'=V' \cup E' \cup E_c$$

여기서 $E_c = \{(v_i, v_k : e_1 \cdot UIO(v_j)) ; (v_i, v_j) \in E \text{ 그리고 } tail(UIO(v_j) = v_k)\}$.

위의 식에서 $tail(UIO(v_j))$ 는 $UIO(v_j)$ 의 수행 후에 도착한 스테이트이고 E_c 는 시험 subsequence $T_{SSij} = TR_{ij} @ UIO(s_j)$ 를 나타낸다. subsequence가 포함된 새로운 그래프 $G'=(V', E')$ 에서 초기 스테이트에서 출발하여 모든 T_{SSij} 들을 1번 포함하고 다시 초기 스테이트로 돌아오는 tour가 최적의 완전 시험 스위트에 해당한다. 이러한 tour를 Euler tour라고 이야기 하는데, Euler tour가 존재하는 조건은 얻어진 그래프 $G'=(V', E')$ 가 symmetric해야 한다. 즉 모든 vertex에 들어오는 edge수와 나가는 edge수가 같아야 한다는 것이다. 이러한 과정이 symmetric augmentation 과정이다.

그래프 $G'=(V', E')$ 가 모든 시험 subsequence를 1번 포함하는 Euler tour를 계산하는데 있어 G' 를 symmetric하게 하기 위해서는 E_c 를 중심으로 “flow maximum cost minimum” 개념으로 E_c 는 한번 포함하고 원래 기존의 천이 E를 부분적으로 포함하여 새로운 형태의 symmetric한 그래프 $G^*=[V^*, E^*]$ 를 구해야 한다.

symmetric한 그래프 $G^*=[V^*, E^*]$ 을 얻기 위해서 $G'=[V', E']$ 의 모든 E_c 는 한번만 포함하고, 기존의 E

를 부분적으로 포함시켜 생성하는데 이렇게 “flow maximum cost minimum” 알고리즘을 적용하여 구해진 $G^*=[V^*, E^*]$ 에서의 RCP tour 계산은 다음의 알고리즘으로 수행된다.

- 1) 포인터 ptr을 초기 스테이트에 두고, 각 스테이트에 관계된 하나의 벡터를 생성한다.
- 2) ptr에 의한 포인터 된 스테이트로 아직까지 탐색되지 않은 천이를 생성한 후, 이 천이의 시작 스테이트 S의 벡터에 ptr에 의해 포인터 된 스테이트를 저장한다.
- 3) 스테이트 S에 들어오는 탐색되지 않은 천이가 있으면 step 2)를 수행하고 ptr을 스테이트 S에 둔다. 그렇지 않으면 step 4)를 수행한다.
- 4) ptr에 의해 포인터 된 스테이트에 들어오는 천이에 의해 연결된 스테이트를 선택하고, 이들 중에서 이들로 들어오는 탐색되지 않은 천이를 가진 또 그들의 벡터값이 있는 스테이트 S가 존재하면 step 3)을 수행하고 그렇지 않을 경우에는 step 5)를 수행한다.
- 5) 초기 스테이트에 관계된 벡터의 마지막 요소로부터 각 천이를 탐색한다. 위에서 벡터의 요소의 값도 다음에 탐색 되어질 스테이트를 나타내고 이와 같은 방법으로 모든 벡터들의 값이 탐색될 때까지 계속한다.

이러한 방법으로 탐색된 천이의 연속이 Euler tour에 해당하며, 궁극적으로 RCP tour에 해당되고 최적의 완전 시험 스위트에 해당한다.

위에 언급된 최적화 방법은 부분적으로 이미 발표된 알고리즘을 응용하였다.

4. INAP 프로토콜로부터 시험계열생성

국제적인 통신 시장의 개방화 경쟁화 추세에 의하여 상호 운용성이 통신 기기 수용을 위한 필수적인 사항이 되어가고 있다. 적합성은 상호 운용을 위한 전제 조건으로 최근 ISO와 ITU 같은 국제 표준화 기구에 의하여 적합성 시험에 관련한 표준화 작업이 이루어지고 있다. 특히 적합성 시험의 방법과 체계가 ISO/IEC JTC1/SC21에서 ISO 9646 Conformance Testing Methodology and Framework(CTMF)로 표

준화되어 각종 적합성 시험 관련 표준화 작업의 기초가 되고 있다[8].

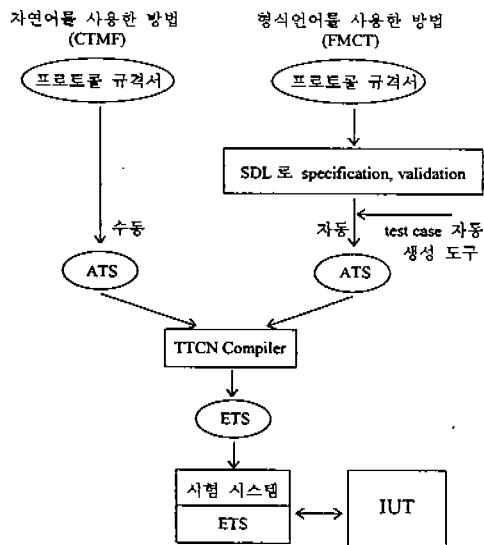
현재 통신 프로토콜의 명세를 위한 FDT의 활용이 점차 확대되어감에 따라 FMCT(Formal Method on Conformance Testing) 방법이 더욱더 강조되고 있다. FMCT와 CTMF의 관계는 "FMCT=CTMF+FDT"의 동식에 의하여 표현될 수 있다. 즉, FMCT는 CTMF의 기본적인 틀 위에서 CTMF를 보충하며, 적합성 시험의 출발점을 FDT로 작성된 규격으로 하는데서 나타나는 특성들을 활용하는 방법을 제시한다.

FMCT는 형식적 기법을 이용함으로써 궁극적으로 적합성 시험까지의 과정을 가능한 한 자동화하기 위하여, 형식 규격에서부터 시험 스위트 개발과 시험에 이르는 과정의 각 단계들 사이의 관계를 밝히고 자동화의 개념적 기반을 제공한다.

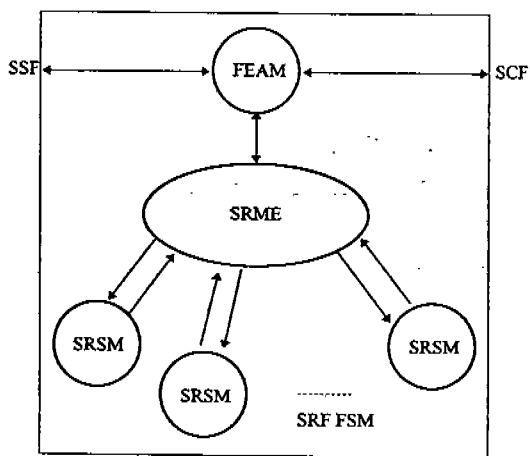
본 논문에서는 프로토콜 규격화 언어로 SDL을 선택해서 사용했는데, 그 주된 이유는 INAP 관련 프로토콜 규격의 복잡성으로 SDL을 이용하여 규격화하는 경향이 그것이고, 또한 다른 한편으로는 SDL을 이용한 표현 및 실행 가능한 프로토타입 구현을 지원하는 SDT, Object-Geode 같은 도구들이 많이 상용화되어 있기 때문이다. 다음 그림(4-1)은 Object-Geode의 관련 도구들을 사용해서 CTMF와 FMCT 방법에 의한 ATS(Abstract Test Suite)와 ETS(Executable Test Suite) 생성 과정이다. 본 논문에서는 FMCT 방법에 의한 ATS 생성을 제한한다.

INAP 프로토콜은 차세대 지능망의 물리실체들간의 상호작용을 지원하기 위한 응용계층 프로토콜로 SSF(Service Switching Function), SCF(Service Control Function), SRF(Service Resource Function)로 구성되고, 이를 지원하는 하부 프로토콜은 공통선 신호 방식(CCS N.7)이다. 여기서 SRF의 구조는 FEAM(Functional Entity Access Manager), SRME(SRF Management Entity), SRSM(SRF Call State Model)으로 구성되며 구조는 다음(그림 4-2)과 같다[9][10].

여기서 FEAM은 하위레벨 인터페이스를 유지보수하는 기능을 제공하고 SCF 및 SSF간 인터페이스 설정 및 유지와 SCF 및 SSF로부터 수신한 메시지를 SRME로 전달하거나 큐에 저장하고, SRME로부터 수신한 메시지를 포매팅하여 SCF 및 SSF로 전송 및 큐에 저장한다.



(그림 4-1) CTMF와 FMCT에 의한 ATS, ETS를 생성하는 방법 (Fig 4-1) ATS and ETS Generation by CTMF and FMCT

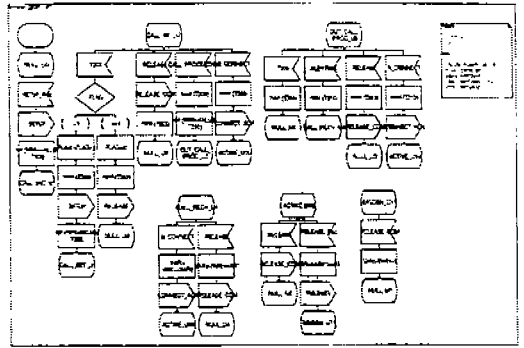


(그림 4-2) SRF FSM의 구조 (Fig. 4-2) Architecture of SRF FSM

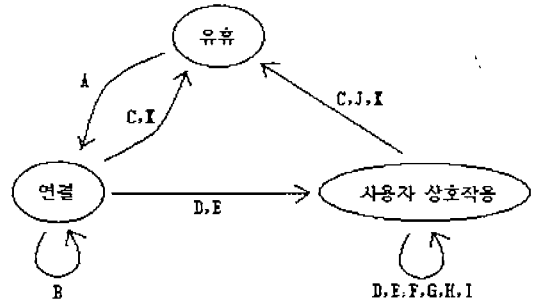
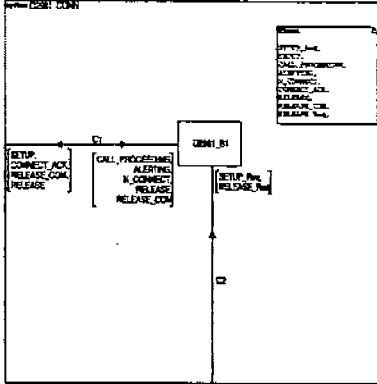
SRME는 SCF로부터 수신한 오퍼레이션의 수행에 관련된 관리기능과 SCSM의 모든 인터페이스를 위해 SSF와 SCF와의 다이얼로그를 유지하고, SRSM의 생성, 개시, 유지보수의 기능과 수신한 입력 메시지를 해석하여 해당 SRSM 사건으로 변환하며, SRSM의 출력 메시지를 다른 기능 실체에 해당하는 메시지로 변환한다.

SRSM은 수신 오퍼레이션에 오류가 있을 때 유지 보수기능에 알리고 SCF에게 보고하고 오류를 처리한다. 일련의 연속된 오퍼레이션을 조사하여 “안내방송” 또는 “사용자 상호작용” 오퍼레이션에 대한 취소 오퍼레이션이 있을 시 즉시 수행을 하며 수신된 오퍼레이션은 큐에 넣고 차례로 처리한다. 임의의 상태에서 SSF와 다이얼로그를 종료하면 SRSM 그 다이얼로그에 할당된 자원을 해제한다. 지나치게 호가 중단 되는 것을 막기 위해 타이머 Tsrfl를 사용한다.

SRF-FSM에 대한 프로토콜 기능을 형식기술기법 중의 하나인 SDL로 명세화하면 (그림 4-3)와 같이 나타내어지고, 프랑스 Verilog사에서 제작한 SDL 검증 도구인 Object-GEODE를 사용하여 여러가지 검증기법을 적용한 후 시뮬레이션을 통해 얻어낸 I/O FSM이 (그림 4-4)에 나타나 있다[9][10].



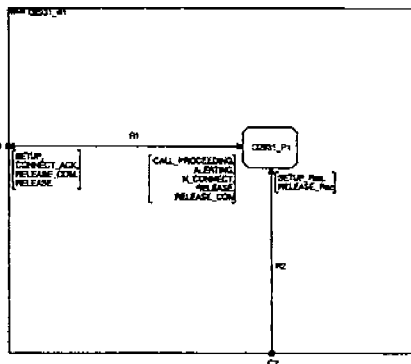
(그림 4-3) SDL을 사용한 SRF-FSM 명세
(Fig. 4-3) SRF-FSM Specification in SDL



(그림 4-4) SDL 명세로부터 생성된 I/O FSM
(Fig. 4-4) I/O FSM Generated from SDL specification

SRF-FSM 구조에 있어서 위에 설명한 SRSM의 각 사건(Event)에 대한 입력/출력을 간단히 요약설명하고, 간결성을 위해 알파벳 A에서 K까지 다음과 같은 순서로 대체한다.

- * (E1): Connect_Request_from_SSF/ ϵ → A (베어러 신호 연결 요구 메시지)
- * (e5): Assist_Request_Instructions_needed/Assist_Request_Instructions_TO_SCF → B (보조명령 요구 필요 오퍼레이션 수신/SRSM이 SCF로 보조명령 요구 오퍼레이션 전송)
- * (E3)(E10): Connect_Released_from_SSF/ ϵ → C (호 해제 메시지)
- * (E2)(E5): Play_Announcement/ ϵ → D (SCF로부터 안내방송 오퍼레이션 수신)



- * (E2) (E5): Prompt_and_Collect_user_Information/ $\epsilon \rightarrow E$
(SCF로부터 사용자 정보 수집 오퍼레이션 수신)
- * (E6): Cancel/ $\epsilon \rightarrow F$ (SCF)로부터 안내방송/사용자정보수집의 취소를 요청하는 취소 오퍼레이션을 수신)
- * (e7): SRF_Report_to_SCF/Send_PAPC_Result_to_SCF $\rightarrow G$
(SCF로 사용자 정보수집 오퍼레이션 반송과 특수자원 보고 오퍼레이션 전송/SCF에 안내방송 및 사용자 정보수집 오퍼레이션 결과 전송)
- * (e8): PAPC_Cancelled_to_SCF/Send_PAPC_Result_to_SCF $\rightarrow H$ (PAPC의 취소/SCF에 안내방송 및 사용자정보수집 오퍼레이션 결과 전송)
- * (e9): Cancel_Error_to_SCF/Send_PAPC_Result_to_SCF $\rightarrow I$ (취소 오퍼레이션의 반송에러를 SCF로 전송/SCF에 안내방송 및 사용자정보수집 오퍼레이션 결과 전송)
- * (e11): Disconnect_to_SSF/Send_PAPC_Result_to_SCF $\rightarrow J$ (SCF로부터 연결해제 시작하도록 하는 최종안내 방송/SCF에 안내방송 및 사용자정보수집 오퍼레이션 결과 전송)
- * (e12) (e4): Tsr/ $\epsilon \rightarrow K$ (Tsr동안 수행할 안내 방송 및 사용자정보수집 오퍼레이션이 없을 경우 발생)

4.1 UIO 방법에 의한 시험 계열 생성

위 (그림 4-4) SRSM의 SDL 명세에 대한 상용하는 I/O FSM인데 각 스테이트에 대한 UIO 시퀀스는 아래와 같다.

STATE	UIO
유휴	Connect_Request_from_SSF / $\epsilon(A)$
연결	Assist_Requeset_Instructions_needed / Assist_Request_Instructions_TO_SCF(B)
사용자 상호작용	Disconnect_to_SSF / Send_PAPC_Result_to_SCF(J)

(그림 4-5) (그림 4-4)의 I/O FSM에 대한 UIO 시퀀스 (Fig. 4-5) UIO sequences for I/O FSM in Fig 3.3

위의 (그림 4-5)에서 생성된 SRSM의 기능을 나타

내는 I/O FSM의 3가지 스테이트(유휴, 연결, 사용자 상호작용)에 대해 생성된 가장 짧은 길이의 UIO 시퀀스를 이용하여 UIO 방법에 의해 시험 계열을 생성하면 다음과 같다. 다음의 각 시험 계열에서 "reset"은 IUT를 초기 상태로 보내는 특별한 천이를 나타내고 "_" 위에 표기된 천이는 TUT를 나타내며, "reset"과 "_" 사이의 시퀀스는 초기 스테이트에서 "_" 위의 시험을 위한 천이의 시작 스테이트까지 가장 짧은 길이의 path를 나타낸다. 또 "_" 다음의 시퀀스는 "_" 위의 천이에 의해 도착한 스테이트의 UIO 시퀀스를 나타낸다. 다음의 각 시험 대상 천이에 대한 UIO 방법에 의해 생성된 시험 계열이다.

- * Connect_Request_from_SSF/ ϵ (A)
 $reset \cdot A \cdot B$
- * Assist_Request_Instructions_needed/Assist_Request_Instructions_TO_SCF(B)
 $reset \cdot A \cdot B \cdot B$
- * Connect_Released_from_SSF/ $\epsilon(C)$, 연결상태 \rightarrow 유휴상태)
 $reset \cdot A \cdot C \cdot A$
- * Tsr/ $\epsilon(K)$, 연결상태 \rightarrow 유휴상태)
 $reset \cdot A \cdot K \cdot A$
- * Play_Announcement/ $\epsilon(D)$, 연결상태 \rightarrow 사용자상호작용상태)
 $reset \cdot A \cdot D \cdot J$
- * Prompt_and_Collect_user_Information/ $\epsilon(E)$, 연결상태 \rightarrow 사용자상호작용상태)
 $reset \cdot A \cdot E \cdot J$
- * PPlay_Announcement/ $\epsilon(D)$, 사용자상호작용상태 \rightarrow 사용자상호작용상태)
 $reset \cdot A \cdot E \cdot D \cdot J$
- * Prompt_and_Collect_user_Information/ $\epsilon(E)$, 사용자상호작용상태 \rightarrow 사용자상호작용상태)
 $reset \cdot A \cdot D \cdot E \cdot J$
- * Cancel/ $\epsilon(F)$
 $reset \cdot A \cdot D \cdot F \cdot J$
- * SRF_Report_to_SCF/Send_PAPC_Result_to_SCF(G)
 $reset \cdot A \cdot D \cdot G \cdot J$
- * PAPC_Cancelled_to_SCF/Send_PAPC_Result_to_SCF(H)
 $reset \cdot A \cdot D \cdot H \cdot J$
- * CANCEL_Error_to_SCF/Send_PAPC_Result_to_SCF(I)

reset·A·D·I·J

*Connect_Released_from_SSF/ε(C, 사용자상호작용 상태 → 유티상태)

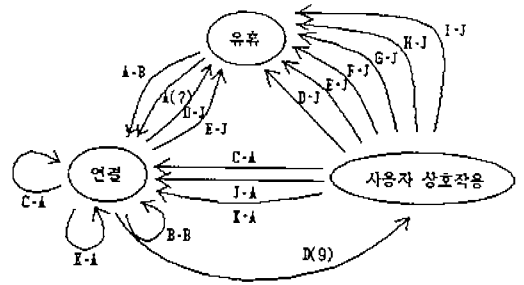
reset·A·D·C·A

*Disconnect_to_SSF/Send_PAPC_Result_to_SCF(J)

reset·A·D·J·A

*Tsrif/ε(K, 사용자상호작용상태 → 유티상태)

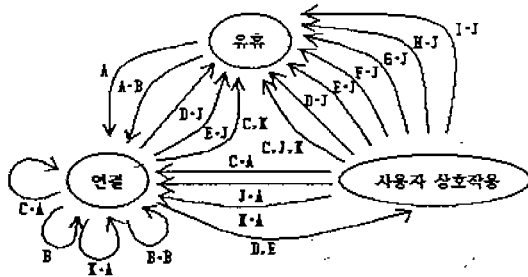
reset·A·D·K·A



(그림 4-7) Symmetric augmented 그래프 G*
(Fig. 4-7) Symmetric augmented graph G*

4.2 최적화 기술에 의한 시험 계열 생성

최적화 기술에 의한 시험계열 생성을 위해 3장에서 기술된 최적화 방법을 적용하면, 먼저 SDL 명세로부터 생성된 (그림 4.4) 기본 I/O FSM에 (그림4.5)에서 도출된 각 스테이트에 대한 UIO 시퀀스를 적용하여 모든 천이 즉, TUT에 대해 시험 subsequence(T_{ssij} = TR_{ij} @ UIO(s_j))가 첨가된 형태로 생성한 그래프 G'가 (그림 4.6)에 나타나 있다.



(그림 4-6) Subsequence를 포함한 그래프 G'
(Fig. 4-6) Graph G' including Test Subsequence

Symmetric한 그래프 G*=[V*, E*]을 얻기 위하여 G'=[V', E']의 모든 시험 subsequence를 한번만 포함하고 기존의 천이 E(유티 상태에서 연결 상태로의 천이 A를 7번 반복, 연결 상태에서 사용자 상호 작용 상태로의 천이 D를 9번 반복)를 포함하여 생성된 Symmetric한 형태의 그래프 G*가 (그림 4.7)이다.

아래 시험 계열은 3장의 최적화 방법에서 RCP Tour 계산 알고리즘을 적용하여 초기 유티상태에서 시작하여 모든 TUT를 한번 도달 또는 표시된 횟수만큼 방문하여 생성된 Euler Tour로써 생성된 완전 시험 스위트이다.

A·B·B·B·C·A·K·A·E·J·A·D·J·A·D·C·A·D·J·A·D·K·A·D·D·J·A·D·E·J·A·D·F·J·A·D·G·J·A·D·H·J·A·D·I·J

4.3 두가지 기법 비교

본 논문에서는 재래의 적합성시험 방법에서의 자연어로 작성된 명세로부터 출발하여 하나하나의 시험항목 수작업 생성 방법이 갖는 모호성, 모순성, 불완전성등의 문제점을 해결하기 위해, 차세대지능망 응용 프로토콜인 INAP의 SRF-FSM 프로토콜기능을 형식기술기법중의 하나인 SDL로 명세화하고 이 명세로부터 규격의 정확성을 해당도구(Object-GEODE)를 사용하여 검증한 후 시뮬레이션 과정을 거쳐 중간 모델인 I/O FSM을 추출하였다. I/O FSM 중간 모델로부터 최적의 시험 계열 생성과 시험 과정의 컴퓨터를 이용한 자동화를 통하여 시험 생성 시간 및 비용을 감소시켰는데, 여기서 최적의 시험계열이란 IUT의 명세에 대한 적합 판정을 위한 시험을 위해 가장 시간이 적게 걸리고 또 어떠한 형태의 구현이든 간에 명세와의 트레이스 동치 관계에 없는 것들을 판단하는 오류 판단 영역이 넓은 것을 의미한다. 이러한 최적의 시험계열 개념에 기본하여 UIO방법과 최적화 기술을 사용한 방법에 대해서 분석한다.

첫째, 시험계열의 갯수에 대해서 분석하기 위해 두 방법에 의해 생성된 각 시험경우가 하나의 시간 유닛을 요구한다고 가정하였을 경우 UIO를 사용한 시험 계열의 갯수는 총 68개 이며, 최적화 기술 기법을 사용한 시험 계열의 수는 총 46개이다. 즉 최적화 기술을 사용한 방법이 UIO 방법에 비해 31%나 개선된 값과 효율적이면서 강력한 시험 계열을 생성함을 알

수 있다.

둘째, "reset" 사용 유무이다. UIO 방법에서는 매 시험 subsequence마다 "reset" 시험 경우를 사용하였는데 일반적으로 IUT를 시험 도중 초기상태로 보내는 행위 자체가 매우 힘이 들고, 또 개념적인 의미만을 지니므로 실제 practical한 시험적용에 있어 문제점을 많이 내포하고 있다. 최적화 기술에 의해 생성된 방법에서는 특별한 친이 "reset"을 전혀 사용하지 않고 완전 시험계열을 생성하였으므로 이 부분이 원래 이 방법을 제안한 [13]의 방법보다도 개선된 방법이다. 또 [13]에 제안된 방법은 I/O FSM 모델로부터 적용된 방법이지만, 본 논문의 방법은 형식기술기법중의 하나인 SDL 명세로부터 최적화 방법을 적용한 것이 새로운 접근 개념이다.

마지막으로 시험계열의 오류 판단 영역 분석을 위해 SDL 명세화로부터 생성된 I/O FSM((그림 4-4))으로부터 Random number generation에 의해 여러가지 유사 transition들을 첨가하여 임의의 구현들을 만들어내어 두 방법에 의해 생성된 시험계열들을 적용한 결과 최적화 방법에 의해 생성된 시험계열이 월등히 높은 오류판단능력을 가짐을 확인했다.

5. 결 론

본 논문에서는 차세대 지능망의 물리실체들간의 상호작용을 지원하기 위한 응용 계층 프로토콜인 INAP 프로토콜을 대상으로, 지능망 서비스 생성자의 관점에서 서비스의 외부 동작의 확인을 위한 서비스 적합성 시험 모델을 제시하고, 제시된 시험 모델을 기반으로 여러가지 엄격한 적합성 시험 기준을 정의한 후, 형식 기술 기법중의 하나인 SDL로 명세하였다. 이러한 INAP 프로토콜의 SDL 명세로부터 ObjectGEODE 도구를 사용하여 시뮬레이션하여 얻은 I/O FSM으로부터 적합성 시험 계열 생성에 대해 연구하였다. 형식기술기법에 기초하여 제안된 최적화 기술로 생성한 최적화된 시험계열이 기존의 UIO 방법에 의한 것보다 31%나 개선됨을 보였고, 또 월등히 높은 오류 판단 영역을 가짐을 보였다. 향후 연구 사항으로 이러한 방법으로 생성된 ATS를 국제표준화 기구에서 제시한 시험 기술 기법인 TTCN(Tree and Tabular Combined Notation)으로 변환하고, 상업적인 도구를

사용하여 위에서 연구된 TTCN으로 표현된 ATS로부터 ETS를 생성하는 방법과 실제 INAP 프로토콜 구현의 표준에 대한 적합성 시험에의 적용 방법 등이 향후 추진되어야 할 연구 사항들이다.

참 고 문 헌

- [1] R.J.Linn, *Conformance Testing for OSI Protocols*, Computer Network and ISDN Systems 18, 1989.
- [2] J. P. Favreau, D. Hogrefefer, J. Kroon, *Formal Methods in Conformance Testing: Status and Expectations*, Proceedings of the 5th Int'l Work-shop on Protocol Test Systems, Montreal, Canada, Sep. 1992.
- [3] Z. Kohavi, *Switching and Finite Automata Theory*, New York, Mc Graw-Hill 1978.
- [4] K. K. Sabnani and A. T. Dahbura, *A Protocol Test Generation Procedure*, Computer Networks and ISDN Systems, Vol. 15, No. 4, 1988.
- [5] W. Y. L. Chan, S. T. Vuong, and M. R. Ito, *An Improved Protocol Test Generation procedure based on UIOv*, SIGCOM'89 Symposium: Communication Architecture and Protocols in Computer Comm. Review 19(4), 283-294, Sep. 1989.
- [6] Younghan Choi, Sungun Kim, Byoungmoon Chin and Sangki Kim, *Communication Protocol Conformance Testing*, ITC-CSCC'96, Seoul, Jun. 1996.
- [7] A. Cavalli, S.Kim and P.Maignon, *Improving Conformance Testing for LOTOS*, Proc. FORTE '93, Boston, USA, Oct. 1993.
- [8] ISO, *Information Processing Systems-Open System Interconnection-OSI Conformance Testing Methodology and Framework*, IS9646, 1995.
- [9] 배성용, 조평동, 차세대 지능망 프로토콜 기술, 정보과학회지, 제13권 제8호, pp.54-64, 8월, 1995.
- [10] ETSI, IN; *IN CS 1 Core Intelligent Network Application Protocol Part 1: Protocol specification*, ETS 300374-1, 1994.
- [11] Verilog, *ObjectGEODE Method Guideline*, Verilog, 1995.

[12] Verilog, *An Introduction to SDL and MSC*, Verilog, 1995.

[13] A.V. Aho, A. Dahbura, D. Lee and M. U. uyar, *An Optimization Technique for Protocol Conformance Test Generation Based on UIO Sequences and Rural Chinese Postman Tours*, in s. Aggarwal Editor, *8th International Symposium on Protocol Specification, Testing and Verification*, 75-86, North Holland, Amsterdam, 1988.



김 상 기

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



김 성 운

1982년 경북대학교 전자공학과 (공학사)
 1990년 프랑스 국립파리 7 대학교 정보공학과(공학 석사)
 1993년 프랑스 국립파리 7 대학교 정보공학과(공학 박사)
 1982년~1995년 한국전자통신연구원 데이터통신연구실(연구원)
 1986년~1995년 한국통신 연구개발원(선임연구원, 실장)
 1990년~1993년 프랑스 전기통신기술연구소(초빙연구원)
 1995년~1993년 부경대학교 정보통신공학과(교수)
 관심분야: 프로토콜 엔지니어링, 데이터 통신 통신프로토콜시험, 컴퓨터네트워크



정 재 운

1996년 부경대학교 정보통신공학과(공학사)
 1997년 부경대학교 대학원 정보통신학과(석사과정)