

Design/CPN을 이용한 Estelle로부터의 프로토콜 시험열 자동 생성 기법

이 현 정[†] · 조 진 기^{††} · 우 성 희^{†††} · 이 상 호^{††††}

요 약

페트리 넷은 병행적이고 비동기적 특성을 가지는 시스템을 분석하고 설계하는 효율적인 모델링 기법 중의 하나이고, CPN은 페트리 넷의 확장형으로서 칼라 토큰을 가진다. 본 논문은 CPN을 이용한 시험열 자동 생성 방법을 제안한다. 제안 방법은 먼저 Estelle 명세를 부분적 자료흐름을 고려한 CPN 형태로 변환하고, 이를 Design/CPN에 입력하여 얻은 결과인 OG와 디스크립터를 이용하여 시험열의 핵심인 서브투어와 UIO를 생성한다. 이 방법은 기존의 방법보다 더 나은 테스트 커버리지를 제공하여 프로토콜 적합성 시험을 효율적으로 지원하며, 자동화된 프로토콜 시험 환경의 기반을 제공한다.

Automatic Generation of Protocol Test Cases from Estelle Using Design/CPN

Hyun-Jeong Lee[†] · Jin-Ki Cho^{††} · Sung-Hee Woo^{†††} · Sang-Ho Lee^{††††}

ABSTRACT

Petri net is one of the effective modeling techniques which analyzes and designs concurrent and asynchronous systems. CPN is an extended Petri net which has color tokens. In this paper, we propose a new test case generation method using CPN. It transforms Estelle Specification into CPN, which is applicable to Design/CPN. It also generates UIO and subtour from OG and descriptor, which are resulted from Design/CPN. Using the proposed method, we can get more improved test coverage than existing methods. Therefore, more effective protocol conformance testing could be conducted. The test case generating method will be the basis of the automatic testing environment.

1. 서 론

통신 프로토콜 적합성 시험은 프로토콜 공학의 구현 단계 중 마지막 단계로, 구현된 프로토콜이 명세서에 기술된 사항을 만족하는가를 시험하는 것으로 구현제품인 IUT에 입력/출력인 시험열을 적용하는 것이다.

프로토콜의 규모가 방대하고 복잡해짐으로써 모든 제어 흐름과 자료 흐름을 적합성 시험에 고려하는 일은 현실적으로 매우 어려운 작업이지만, 제어 흐름만을 고려할 경우 생성된 시험열은 부분적으로 실행 불가능하거나 비결정성이 존재할 수 있다[1].

페트리 넷(Petri net)의 확장된 모델인 CPN(Colored Petri Nets)[2]은 컴퓨터 언어 문장과 관련된 특수화된 시각적인 네트워크로서, 시스템 설계와 시뮬레이션에 전형적으로 사용된다. 이것은 또한, 제어흐름은 물론 자료흐름, 병행성(concurrency) 문제까지도 해결할 수

† 정 회 원 : 한국전자통신연구원
†† 정 회 원 : 부산경상대학 전자계산과 교수
††† 정 회 원 : 청주과학대학 컴퓨터과학과 교수
†††† 송신회원 : 충북대학교 컴퓨터과학과 교수
논문접수 : 1999년 5월 20일, 심사완료 : 1999년 10월 12일

있는 강력한 도구이다.

본 논문에서는 시험열을 생성하기 위해 시스템 시뮬레이션 도구이자 시스템의 설계, 명세 분석을 위한 도구인 Design/CPN을 이용하였다. 기존의 시험열 생성 방법은, Estelle, SDL, Lotus 등 프로그래밍 언어지향적 모델로 기술된 프로토콜을 FSM, EFSM 등의 전이 모델 형태로 변환하여 시험열을 생성하였다. 이러한 작업을 수작업으로 진행하게 되면 많은 오류 가능성을 내포하게 된다. 본 논문에서는 Estelle로 표기된 프로토콜을 CPN 형태로 변환한 후, 이것을 Design/CPN에 입력하여 얻은 결과인 OG(Occurrence Graph)와 디스크립터(Descriptor)로부터 시험열을 생성한다. 이러한 자동화된 방법을 사용함으로써 오류 가능성을 줄일 수 있을 뿐 아니라, 시간과 노력을 감소시킬 수 있다. 또한 Estelle 표현을 CPN으로 변환시 부분적으로 자료흐름을 고려하였다.

본 논문은 첫 번째 단계로 Estelle를 Design/CPN에 입력 가능한 CPN으로 구성하는 알고리즘을 설계하였고, 두 번째 단계로 Design/CPN으로 생성된 OG와 디스크립터로부터 시험열의 핵심인 서브투어와 UIO를 생성하는 알고리즘을 제안하였다.

본 논문의 구성은 다음과 같다. 2절에서는 Estelle와 Design/CPN에 대해 간략히 소개하였고, 3절에서는 Estelle로 기술된 프로토콜 명세의 CPN 변환방법, 4절에서는 사례 연구로서 제안된 방법을 Estelle로 기술된 TPO(OSI Class 0 Transport Protocol)[1]에 적용하여 CPN으로 변환하고, 이로부터 자동적으로 생성된 내부 정보로부터 UIO와 서브투어의 생성 과정과 결과 분석을 하였다. 마지막으로 5절에서는 결론 및 향후 연구 방향을 제시하였다.

2. 명세 언어와 모델링

2.1 제약 사항

실세계의 프로토콜을 모델링하여 시험하면 규모가 방대하고 복잡하여 실제적 시험이 불가능하다. 따라서 본 논문에서 사용하는 통신 프로토콜은 다음의 사항을 만족한다[1]:

- (1) 완전히 명세되었다(completely specified).
- (2) 강하게 연결되었다(strongly connected).
- (3) 최소 밀리 머신(minimal Mealy machine)이다.

(4) 단일 모듈 명세(single module specification)이다.

(5) 리셋 기능(reset feature)을 가진다.

2.2 Estelle와 Design/CPN

Estelle는 분산 환경에서 병행성(concurrency)을 요구하는 정보처리 시스템을 기술하기 위하여 ISO에서 개발, 표준화된 FDT(Formal Description Techniques) 중의 하나로서, 특히 ISO에서 제정하고 있는 통신 프로토콜 및 서비스의 표현에 적용할 목적으로 만들어졌다. Estelle는 파스칼(PASCAL)에 기반을 둔 구문(syntax)과 의미(semantic)를 가지고 있고, EFSM(Extended FSM)의 정의를 쉽게 할 수 있는 등 여러 기능을 가진 언어이다. 특히 Estelle 구문은 프로토콜의 함수(function)를 표시하는 모듈(module)이라는 개념과 모듈들간의 통신을 제공해 주는 채널(channel)이라는 개념을 이용하여 구성되므로 통신 프로토콜을 기술할 때 융통성, 용이성 및 효율성이 높아 널리 이용되고 있다[3].

페트리 넷은 시스템을 분석하고 설계하는 모델링 기법으로서, 오브젝트와 정보의 흐름을 다루는 복잡한 분산시스템의 설계와 분석을 용이하게 한다[2]. 단일 칼라를 갖는 페트리 넷에 칼라의 개념을 도입하여 시스템의 구조 및 기능을 더욱 명확히 나타내도록 확장한 것이 CPN이다.

Design/CPN은 CPN의 사용을 지원하는 툴 패키지로써, 복잡한 데이터 타입 및 복잡한 데이터 처리를 하는 CPN 모델을 지원, 잘 정의된 인터페이스를 가지는 분리된 모듈, 즉 계층적 CPN을 지원한다.

본 논문에서는 Estelle로 표기된 프로토콜 명세를 셀 프루프(self-loop)와 자료흐름을 고려하여 CPN 표기로 변환한 후 그 결과인 OG와 디스크립터로부터 시험열의 핵심인 서브투어와 UIO를 생성하였다. SDL도 Estelle와 같이 프로그램 언어지향적 모델이면서 FSM, EFSM, 페트리 넷 등과 같은 전이모델로 표현가능하므로 CPN에 적용가능하다.

3. Estelle 명세의 CPN 변환

3.1 Estelle 명세에서의 가정

Estelle는 내부 동작과 데이터 표현을 파스칼 구문인 통신 상태 오토마타(communicating state automata)의 구조로 기술한다. 따라서 Estelle 모듈의 내부적인 동적 행위(behavior)는 CPN의 트랜지션(transition)으로 변

환된다. 변환하기 전의 Estelle 모듈에 대한 가정은 다음과 같다[1][4][5].

- (1) 모듈은 다음과 같이 유한 개의 상태집합이다: C_0, C_1, \dots, C_n
- (2) C_0 는 초기화 부분에서 정의된 초기상태이다.
- (3) 전이 선언은 2개의 부분으로 구성되고 환경과 모듈과의 상호 동작 뿐 아니라 내부적 행위를 서술한다.
 - 전이 조건
전이 다음 상태로 점화 가능함을 결정한다.
 - "FROM C": C는 제어 상태(control state)로서 현재 상태
 - "WHEN I": I는 인터랙션 포인트
 - "PROVIDED B": B는 부울 표현
 - 전이 동작
점화 가능한 전이 중 하나의 동작을 수행하고 다음 상태에 도달하게 된다.
 - "TO D": D는 제어 상태로서 다음 상태
 - 전이블럭(TB): "BEGIN"과 "END" 사이의 파스칼 구문
- (4) Estelle로 기술된 프로토콜 명세는 정규화(normalize)된 형태로 입력된다.

3.2 CPN에서의 가정

CPN은 페트리 넷에 칼라의 개념을 추가하여 다음과 같이 6개의 튜플(P, T, C, I, O, μ)로 구성된다.

$$CPN = (P, T, C, I, O, \mu)$$

여기서

- P: 플레이스의 집합 = $\{p_1, p_2, \dots, p_n\}, n \geq 0$
플레이스는 상태를 나타내며 토큰의 유무에 따라 점화가능 여부가 결정된다.
- T: 트랜지션의 집합 = $\{t_1, t_2, \dots, t_m\}, m \geq 0$
트랜지션은 현 상태 플레이스로부터 다음 상태 플레이스로의 전이를 나타낸다.
- C: 칼라(color) 함수, $C(p) = \sum_k k \cdot c, k \geq 0$ 이고 c는 칼라 토큰의 종류를 구별하여 나타내기 위한 것이다.
- I: 입력함수
현 상태 플레이스가 토큰을 가지고 점화하여 다음 상태 플레이스로 전이시키는 함수.
- O: 출력함수
현 상태 플레이스가 토큰을 가지고 점화하여, 다

음 상태 플레이스와 다음 상태 플레이스에서 사용되는 토큰을 출력시키는 함수

$\mu(P \rightarrow (0, 1, \dots))$: 초기에 토큰을 가지고 있는 플레이스의 집합

$P \cap T = \emptyset$ and $P \cup T \neq \emptyset$: 플레이스와 트랜지션의 교집합은 공집합이고, 플레이스와 트랜지션의 합집합은 공집합이 아니다.

3.3 변환 알고리즘

Estelle로 표현된 프로토콜 명세의 CPN 변환방법을 형식적(formal)으로 표현하면 다음과 같다.

$$P = CUDU(IO \cdot B)$$

$$T = \{t_{c,k} \mid c \in C \text{ and } k \in (IO \cdot B)\}$$

$$C(p) = C(q), \text{ if } (p \in C \ \& \ p \text{ doesn't have self-loop})$$

$$C(r), \text{ if } (p \in C \ \& \ p \text{ has self-loop})$$

$$C(iob), \text{ if } iob \in (IO \cdot B)$$

$$I(t_{c, IO \cdot B}) = \{C, (IO \cdot B)\}$$

$$O(t_{c, IO \cdot B}) = D$$

$$\mu(p) = C(0) \cup \sum (IO \cdot B)$$

$C(0)$: 초기상태 플레이스

CPN은 플레이스 P와 트랜지션 T, 칼라 함수 C, 입력함수 I, 출력함수 O, 초기상태 μ 로 구성된다. 여기서 플레이스 P는 입출력/조건 카테선 프러덕트(product)로 표현된 입출력/조건 플레이스와 현재 상태와 다음 상태를 나타내는 일반적인 상태 플레이스, 셀프루프를 가진 상태 플레이스의 세 종류가 있다. [6]에서는 CPN으로 변환시 셀프루프를 고려하지 않았으나, 이 연구에서는 실행불가능 경로를 갖지 않도록 셀프루프에 순서를 부여하여 셀프루프 플레이스로 변환하였다. 실행불가능 경로의 제거 방법과 셀프루프의 고려는 [7]과 [8]에 기술하였다. [7]에서는, 실행불가능 경로가 발생하지 않도록 셀프루프에 순서를 부여하여 실행불가능 경로를 제거하였으며, [8]에서는 셀프루프 플레이스가 자신에게 토큰을 송수신함으로써 셀프루프를 처리하는 방법을 제안하였다.

트랜지션 T는 상태 플레이스가 입출력/조건 플레이스와 함께 점화하여 다음 상태로 전이하도록 한다.

칼라 함수인 C는 플레이스의 종류에 따라 칼라를 부여한다. 토큰은 초기상태 플레이스와 입출력/조건 플레이스들에 위치하게 된다. 토큰은 토큰의 개수를 나타내는 계수와, 칼라로 구성된다. 초기상태 플레이스에

는 임의의 칼라를 가진 하나의 토큰이 위치하게 되고, 각 입출력/조건 플레이스에는 입출력/조건에 따른 칼라를 가진 토큰이 위치하게 된다. 플레이스는 상태 플레이스와 입출력/조건 플레이스로 구분되며, 상태 플레이스는 또한 셀프루프를 갖는 플레이스와 갖지 않는 플레이스로 구분된다. 플레이스가 상태 플레이스이면 셀프루프를 갖지 않는다면 임의의 한 칼라가 모든 상태 플레이스에 부여되며 초기상태 플레이스만이 칼라에 따른 임의의 토큰을 갖는다. 셀프루프를 갖는 상태 플레이스는 임의의 고유한 칼라를 갖는다. 만약 초기 상태 플레이스가 셀프루프를 가진다면 초기상태 플레이스는 다른 상태 플레이스와 다른 고유의 칼라 및 토큰을 가지게 된다.

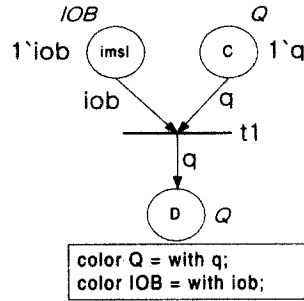
입력함수인 I는 현재 상태 플레이스인 C의 토큰과 입출력/조건 플레이스의 토큰을 입력으로하여 점화한다. 트랜지션에서 출력함수인 O는 다음 상태인 D에 토큰을 전송한다. 초기 상태인 μ 는 초기 상태 플레이스와 입출력/조건 플레이스에 각각 토큰이 부여되어 있는 상태이다.

아래와 같은 트랜지션 단위로 구성된 정규화된 형태의 Estelle 형식의 프로토콜은 (그림 1)과 같은 CPN으로 변환된다.

```

-----
WHEN I      : I는 인터랙션 포인트로서, PROVIDED 절의 B와 결합하여 입출력/조건 플레이스로 대응
FROM C      : C는 제어 상태(control state)로서, 현재상태 플레이스로 대응
PROVIDED B  : B는 전이조건으로서 부울 표현
TO D        : D는 제어 상태로서, 다음 상태 플레이스로 대응
t1 : BEGIN  : t1은 전이의 번호
.
.
.
output O: O는 출력 인터랙션
END;
-----
    
```

(그림 1)과 같이 Estelle의 입력 인터랙션과 출력 인터랙션인 WHEN 절과 output 절은 자료흐름이 고려되도록 PROVIDED 절과 결합하여 CPN의 입출력/조건 플레이스로 변환되고, FROM 절은 현재 상태 플레이스



(그림 1) Estelle의 CPN 변환 예

스로, TO 절은 다음 상태 플레이스로 변환된다. 트랜지션 t1은 현재 상태 플레이스인 C가 토큰 q를 가지고 입출력/조건 플레이스인 imsi의 토큰 iob와 함께 점화하여 다음 상태 플레이스인 D에 토큰 q를 전송하면서 전이하도록 한다. (그림 1)의 하단에는 칼라에 대한 정의(declaration) 부분이 나타나 있다. 칼라 Q는 토큰 q를 가지며, 칼라 IOB는 토큰 iob를 가진다. 트랜지션의 입력이 되는 모든 플레이스에 토큰이 한 개 이상이면 그 트랜지션은 점화 가능하다.

본 논문에서는 Estelle로 표기된 프로토콜을 CPN 형태로 변환한 후, 이것을 Design/CPN에 입력하여 얻은 결과인 OG와 디스크립터로부터 시험열을 생성한다. 이러한 자동화된 방법을 사용함으로써 오류 가능성을 줄일 수 있을 뿐 아니라, 수작업으로 진행하는 것에 비해 시간과 노력을 감소시킬 수 있다. 또한 Estelle 표현을 CPN으로 변환 시 PROVIDED 절을 입출력/조건 플레이스로 표현함으로써 부분적으로 자료흐름을 고려하였다.

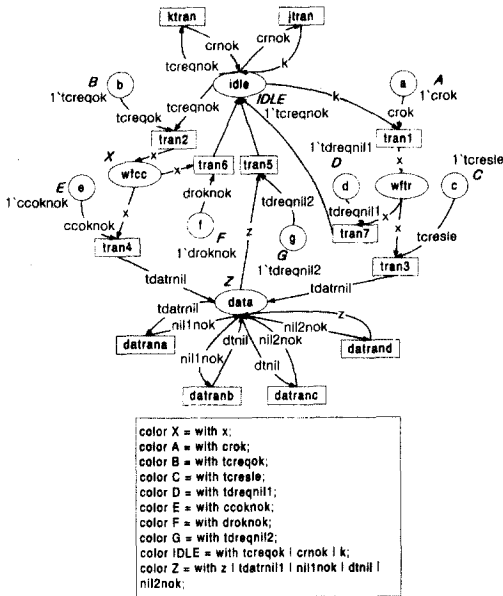
4. 사례 연구

4.1 개요

본 논문에서는 Estelle로 기술된 TPO[1][9][10] 프로토콜 명세를 CPN으로 변환하였다. 변환 수행 시 셀프루프는 실행불가능 경로가 제거된 임의의 순서를 가지고 있다는 것과, 한 간선에 두 개 이상의 입출력/조건이 존재하면 임의의 한 입출력/조건을 사용한다는 가정을 하였다. 또한, 초기상태 플레이스의 indegree 에지는 라벨을 갖지 않는다. 왜냐하면 초기상태 플레이스로 indegree 에지가 들어오게 되면 한 경로가 완성되기 때문이다.

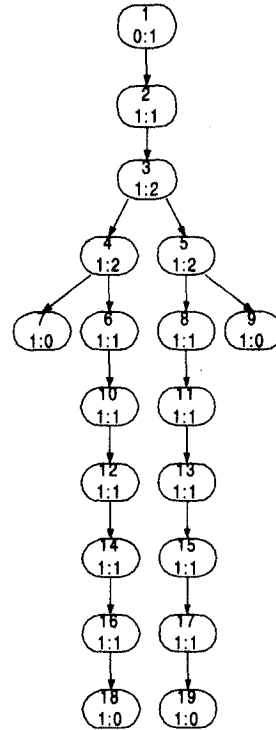
4.2 FSM의 CPN 변환

TP0의 CPN 표현은 (그림 2)에 나타나 있다. (그림 2)에서 초기 상태 플레이스인 idle은 셀프루프를 가지는 플레이스로서 다른 상태 플레이스와는 다른 칼라인 IDLE과 토큰 1'tcreqnok를 가지고, data 플레이스도 셀프루프 플레이스로서 칼라 Z를 가지며, wfcc와 wftr은 상태 플레이스로서 같은 칼라인 X를 가지며 토큰은 갖지 않는다. 입출력/조건 플레이스인 a, b, c, d, e, f, g는 각각의 입출력/조건에 따른 고유한 칼라와 토큰을 가진다. 여기서 입출력/조건을 고려하여 토큰에 이름을 부여하는데, 출력은 고려하지 않고 입력과 조건만을 고려하여도 고유한 이름을 부여할 수 있으므로 출력은 고려하지 않았다. 토큰의 이름에 조건을 나타내는 PROVIDED 절을 고려함으로써 자료흐름을 고려하였다.



(그림 2) Estelle로 표기된 TP0의 CPN 표현

(그림 3)은 (그림 2)를 Design/CPN으로 수행한 결과로 얻은 OG이다. OG는 초기 상태 플레이스부터 시작하여 점화가능한 트랜지션을 차례로 실행하여 다음 상태로 전이하여 다시 초기 상태로 되돌아오는 경로를 트리 형태로 표현한 것이다. OG에서 각 노드는 노드 번호, 부모노드의 개수, 자노드의 개수에 대한 정보를 갖는다[6]. (그림 4)는 (그림 3)의 노드 1의 토큰 보유 정보를 보여주는 디스크립터로서, (그림 3)의 모든 노드



(그림 3) TP0의 OG

1	New'idle 1: 1'tcreqnok
	New'a 1: 1'crok
	New'b 1: 1'tcreqok
	New'wfcc: 1'empty
	New'wftr: 1'empty
	New'e 1: 1'coknok
	New'c 1: 1'tcesle
	New'data: empty
	New'g 1: 1'tdreqnil2
	New'd 1: 1'tdreqnil1
	New'f 1: 1'droknok

(그림 4) (그림 3)의 노드 1에 대한 디스크립터

에 대해 (그림 4)와 같은 디스크립터가 존재한다.

4.3 서브투어와 UIO 생성

이 연구에서는 (그림 3)과 (그림 4)에 [6]의 알고리즘을 적용하여, TP0의 UIO와 서브투어인 <표 1>과 <표 2>를 얻었다.

<표 1>은 (그림 3)과 (그림 4)에 의한 서브투어로서, CPN 표현시 정해진 순서에 의하여 셀프루프를 고

<표 1> (그림 3)과 (그림 4)에 의한 서버투어

서브투어	state
서브투어 1	idle - idle - idle
서브투어 2	idle - wftr - data - data - data - data - data - idle
서브투어 3	idle - wftr - idle
서브투어 4	idle - wfcc - data - data - data - data - data - idle
서브투어 5	idle - wfcc - idle

<표 2> (그림 3)과 (그림 4)에 의한 UIO

상태 플레이스	UIO
idle	tcrcqnok
wfcc	ccoknok
wftr	tcresle
data	tdatmil

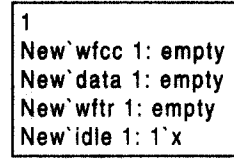
려하여 기존의 방법[6]에서 얻어진 서버투어와 동일하다. TPO는 초기상태 플레이스인 idle이 셀프루프를 갖기 때문에 이 경로도 하나의 서버투어로서 <표 1>의 서버투어 1로 나타내었다.

<표 2>는 (그림 3)과 (그림 4)에 의한 UIO로서, Estelle로 표기된 프로토콜을 EFSM으로 변환한 방법에 의해 얻어진 UIO와 같은, 길이가 1인 UIO를 가진다.

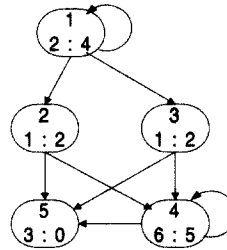
4.4 비교 및 결과 분석

제어 흐름만을 고려한 TPO의 CPN 형태는 단일 칼라와 토큰을 가지며 이로부터 생성되는 OG와 디스크립터는 (그림 5)와 (그림 6)과 같다. (그림 5)는 노드 1과 4에서 셀프루프를 가진다. 이러한 경우 셀프루프를 가지는 상태에서 무한 루프에 빠지게 되어 시험열의 수가 무한히 증가하며, 셀프루프를 가지는 상태 플레이스인 data에서 순서가 고려되지 않는 실행불가능 경로가 발생할 수 있다[7]. (그림 6)으로부터 생성되는 서버투어의 개수는 셀프루프에 순서를 고려하지 않기 때문에 무한개가 되며, UIO는 자료흐름을 고려했을 때와 같다. 따라서 이러한 문제점은 토큰에 칼라를 주어 자료 흐름을 표현하고, 셀프루프에는 순서를 부여함으로써 해결하였다.

이 연구에서 제안된 바와 같이 Estelle로 기술된 프로토콜 명세를 실행불가능 경로가 제거되도록 셀프루프를 고려하여 CPN 표현으로 변환하였다. 변환시 Estelle의 트랜지션 부분에서 PROVIDED 절을 입력력/조건 플레이스에 표현함으로써 자료흐름을 고려하였다. 이



(그림 5) (그림 5)의 노드 1에 대한 디스크립터



(그림 6) 제어흐름을 고려한 TPO의 OG

와 같이 자료흐름을 고려한 CPN 표현 방식의 프로토콜을 Design/CPN에 입력하면 UIO와 서버투어를 생성할 수 있는 정보를 자동적으로 얻을 수 있다.

5. 결론 및 향후 연구 방향

이 연구에서는 Estelle로 표기된 프로토콜 명세로부터 부분적으로 자료 흐름이 고려된 시험열을 얻기 위해 PROVIDED 절을 입력조건으로 추가하여 Design/CPN에 적용 가능한 CPN으로의 변환 방법을 제안하였다. 그리고 변환된 CPN을 Design/CPN에 입력하여 얻은 OG와 각 노드에 대한 내부 정보인 디스크립터를 이용하여 서버투어와 UIO를 생성하였다. 또한 기존의 연구[6]에서 고려하지 않았던 셀프루프를 고려하여 Design/CPN에 적용하였다. 기존의 시험열 생성 방법[4]은 Estelle로 기술된 프로토콜 명세를 EFSM으로 변환 후, 서버투어와 UIO를 생성하는 각각의 알고리즘에 따라 시험열을 생성하였다. 이러한 방법은 여러 단계의 복잡한 과정을 수작업으로 수행하기 때문에 많은 오류 가능성을 내포한다. 따라서 이 연구에서는 자동화된 툴을 이용하여 보다 간편하고 효율적으로 시험열과 추가 정보인 UIO를 얻을 수 있도록 CPN 변환 알고리즘을 제안하였다. 또한, PROVIDED 절을 입력력/조건 플레이스에 표현함으로써 부분적으로 자료흐름을 고려하였고, 제안한 방법은 자동화된 프로토콜 시험

환경의 기반을 제공한다.

향후 연구 방향은 프로토콜의 동작을 구체화하는 다른 요소들을 추가 및 고려하여 자료흐름이 고려된 시험열을 생성하는 것과, 프로토콜을 직접 CPN으로 설계하여, 프로토콜의 설계 및 구현, 분석이 하나의 통합된 환경에서 자동적으로 수행될 수 있도록 하는 것이다.

참고 문헌

- [1] 오병호, "자료 흐름의 변칙성을 고려한 프로토콜의 테스트 스위트 생성", 충북대학교 박사학위 청구 논문, 1996.
- [2] K. Jensen, Coloured Petri Nets, Vol.1, 2nd Ed., Stringer-Verlag, 1997.
- [3] W. Chun and P. D. Amer, "Test Case Generation for Protocols Specified in Estelle," FORTE'90, Marrid, No.11, pp.197-210, 1990.
- [4] 우성희, 오병호, 이상호, "Estelle로 표현된 프로토콜의 테스트 스위트 생성에 관한 연구", 정보과학회 춘계학술발표 논문집 제21권 제1호, pp.463-466, 1994.
- [5] 인소란, "Extended States Transit Language(ESTELLE)", 한국정보과학회지, 제6권, 제1호, 2. pp. 36-42, 1988.
- [6] 이현정, 우성희, 이상호, "Design/CPN을 이용한 테스트 케이스 생성", 한국통신학회 하계학술발표 논문집 Vol.17, No.2, pp.1285-1288, 1998.
- [7] 이현정, 우성희, 오병호, 이상호, "실행불가능 경로가 제거된 테스트 케이스 생성", 정보과학회 춘계학술발표 논문집 제25권 제1호, pp.443-445, 1998.
- [8] 우성희, 이현정, "셀프루프를 고려한 테스트 케이스 생성", 청주과학대학 논문집 제24권, pp.161-173, 1998.
- [9] H. Ural and B. Yang, "A Test Sequence Selection Method for Protocol Testing," IEEE Transactions on Communications, Vol.39, No.4, pp.514-523, 1991.
- [10] B. Sarikaya, G. V. Bochmann and E. Cerny, "A Test Design Methodology for Protocol Testing," IEEE Transactions on Software Engineering, Vol. SE-13, No.5, pp.518-531, 1987.



이현정

e-mail : hjlee294@etri.re.kr

1997년 충북대학교 컴퓨터과학과 졸업(학사)

1999년 충북대학교 전자계산학과 졸업(이학석사)

1999년~현재 한국전자통신연구원 연구원

관심분야 : 프로토콜 공학, 인터넷, ATM 등



조진기

e-mail : jincho@soback.kornet21.net

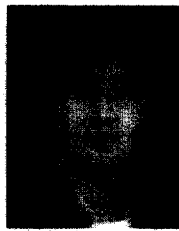
1986년 충북대학교 전산통계학과 졸업(학사)

1988년 숭실대학교 전자계산학과 졸업(공학석사)

1998년 충북대학교 전자계산학과 박사수료

현재 부산경상대학교 전자계산과 조교수

관심분야 : 프로토콜공학, 전자상거래, 암호학 등



우성희

e-mail : shwoo@cjnet.chongjunc.ac.kr

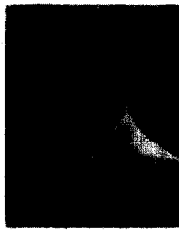
1990년 청주대학교 전자계산학과 졸업(공학사)

1993년 충북대학교 대학원 전자계산학과 졸업(이학석사)

1999년 충북대학교 대학원 전자계산학과 졸업(이학박사)

1995~현재 청주과학대학 컴퓨터학과 조교수

관심분야 : 프로토콜 공학, 컴퓨터 네트워크, 데이터 통신, 소프트웨어공학 등



이상호

e-mail : shlee@cbucc.chungbuk.ac.kr

1976년 숭실대학교 전자계산학과 졸업(공학사)

1981년 숭실대학교 대학원 전자계산학과 졸업(공학석사)

1989년 숭실대학교 대학원 전자계산학과 졸업(공학박사)

1981년~현재 충북대학교 컴퓨터학과 교수

관심분야 : Protocol Engineering, Network Security, Network Management, Network Architecture