

SDL로부터 프로토콜 시험열 자동 생성 기법

정희원 이현정*, 최영일*, 이병선*

Automatic Generation of Protocol Test Sequences from SDL

Hyun-jeong Lee*, Young-il Choi*, Byung-sun Lee* *Regular Members*

요 약

적합성 시험은 프로토콜 구현물인 IUT(Implementation Under Test)가 프로토콜 규격과 일치하는지를 검증하는 것으로, 이 적합성 시험의 질은 생성된 시험열에 의존한다. 기존에는 시험열 생성이 수작업으로 이루어지기 때문에 오류의 가능성을 내포한다. 본 논문에서는 자동화된 툴(tool)인 Design/CPN을 이용하여 SDL(Specification and Description Language)로 기술된 프로토콜의 시험열을 생성한다. SDL로 기술된 프로토콜을 자료흐름을 고려한 CPN(Colored Petri Nets)으로 변환 후 Design/CPN에 입력하여 자동 생성된 OG(Occurrence Graph)와 디스크립터(descriptor)로부터 시험열을 생성함으로써, 기존의 시험열 생성 방법에 비해 오류 가능성을 줄일 수 있으며, 자료흐름이 고려된 시험열을 얻을 수 있다.

ABSTRACT

Conformance test verifies that a protocol implementation conforms to the protocol specification. The quality of the conformance test depends on the test sequences. Existing test sequence generation methods have potential errors because they are manually done. In this paper, we propose an automatic generation of protocol test sequences from SDL using Design/CPN, which has extended test coverage. We transform SDL into CPN form with data flow, and apply it to Design/CPN. Our method has test sequences with partial data flow and reduces errors compared with existing manual method.

I. 서론

프로토콜 명세화에 대한 연구는 ISO(International Organization for Standardization)나 CCITT(Consulative Committee International Telegraph and Telephone)와 같은 연구기관에서 많이 시행되었으며, 그 중에서 SDL, Estelle(a formal description technique based on an Extended State Transition model), LOTOS(Language Of Temporal Ordering Specification)와 같은 프로그래밍 형식의 명세 기법은 형식적(formal) 표현 기법으로서 국제 표준화되었다^[6].

페트리 넷(Petri nets)의 확장된 모델인 CPN^[2]은

컴퓨터 언어 문장과 관련된 특수화된 시각적 넷 워크로서, 시스템 설계와 시뮬레이션에 사용된다. Design/CPN은 CPN의 사용을 지원하는 시스템 시뮬레이션 툴로서 시스템의 설계, 명세, 분석을 위한 도구이고, 병행적이며 비동기적인 특성을 지닌 시스템에 적용 가능하다.

기존의 시험열 생성은, SDL, Estelle, LOTOS 등 프로그래밍 언어 지향적 모델로 기술된 프로토콜을 FSM(Finite State Machine), EFSM(Extended FSM), 페트리넷 등의 전이모델 형태로 변환하여 시험열을 생성한다. 이러한 작업을 수작업으로 진행하게 되면 많은 오류 가능성을 내포하게 된다.

본 논문에서는 SDL/PR(Specification and Description Language/textual Phrase Representation)

* 한국전자통신연구원 통신시스템연구부 개방형서비스팀(bslee@etri.re.kr)
논문번호 : 99496-1223, 접수일자 : 1999년 12월 23일

로 표현된 프로토콜 명세를 셸프루프와 부분적 자료흐름이 고려된 CPN 표현으로 변환 후 Design/CPN에 적용하여, 자동 생성된 결과인 OG와 디스크립터로부터 시험열을 생성한다. 제안된 방법을 이용하면, 기존의 수작업에 의한 시험열 생성 방법에 비해 오류 가능성을 줄일 수 있고, 자료흐름을 고려함으로써 향상된 커버리지(coverage)를 갖는 시험열을 생성할 수 있다.

본 논문의 구성은 다음과 같다. 2절에서는 프로토콜의 가정 사항 및 SDL과 Design/CPN에 대해 기술하고, 3절에서는 SDL/PR로 표현된 프로토콜 명세를 CPN으로 변환하는 방법을 설명한다. 4절에서는 사례 연구로서, SDL/PR로 기술된 Inres 프로토콜의 Initiator^[11]에 제안된 방법을 적용하여 CPN으로 변환하고, 이를 Design/CPN에 적용하여 자동적으로 생성된 내부 정보로부터 시험열을 생성하고, 생성된 시험열을 제어흐름만을 고려한 시험열과 비교 분석한다. 마지막으로 5절에서는 결론 및 향후 연구방향을 제시한다.

II. SDL과 Design/CPN

1. 제약사항

실세계의 프로토콜은 규모가 방대하고 복잡하여 실제로 모델링하기가 어렵다. 따라서 본 논문에서 사용하는 모델은 다음의 사항을 가정한다^{[8][9]}.

- 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)이 있다.

2. SDL과 Design/CPN

SDL은 다양한 통신 시스템이 SPC(Stored Program Control) 기법을 사용하게 됨에 따라 소프트웨어 구성 및 관리의 문제가 발생하여, 통신 시스템의 동작을 명확히 명세하고 서술하기 위한 형식 언어로서 사용되고 있다^[6]. 이 모델링 방법에서 시스템은 상호 연결된 수많은 모듈 즉 블록으로 구성되고, 각 블록의 동작은 하나 혹은 여러 개의 프로세스에 의해 모델링 되며, 이들 사이의 통신은 채널과 무한 용량의 큐를 통하여 비동기적으로 이루어진다. 또한 데이터는 추상적인 데이터 타입으로 정의된다.

SDL의 표현 방법에는 동일한 의미를 지닌 3가지 방법 즉, 순서도와 유사하며 이해하기 쉽고 구성이 용이한 SDL/GR(Specification and Description Language/Graphical Representation) 표현과, 프로그램과 유사하며 컴퓨터가 분석하기 용이한 SDL/PR 표현, 그리고 전통적인 통신 분야에서 사용되는 심볼만을 SDL/GR과 결합한 표현 양식인 SDL/PE(Specification and Description Language/Pictorial Element) 표현이 있다.

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 패키지(package)로서, 복잡한 데이터 타입 및 복잡한 데이터 처리를 하는 CPN 모델을 지원하며, 잘 정의된 인터페이스를 가지는 분리된 모듈, 즉 계층적 CPN을 지원한다. Design/CPN은 편집기, 시뮬레이터, OG 툴, Performance 툴로 구성된다. 편집기는 CPN 모델의 구성, 수정, 구문 체크에 이용되고, 시뮬레이터는 CPN의 인터랙티브 및 자동 시뮬레이션을 지원하며, OG 툴은 CPN 모델을 위한 OG의 생성과 분석에 사용되는데, OG는 도달가능 트리(reachability tree)로도 알려져 있다. Performance 툴은 CPN의 성능 분석에 기반한 시뮬레이션을 제공한다.

III. SDL/PR로 표현된 프로토콜의 CPN 변환

1. SDL/PR의 가정 사항

SDL/PR은 프로그램과 유사한 표현 방법으로서, 구조적 표현 규칙인 BNF(Backus Naur Form)를 사용한다. 이 방법은 프로그래밍 언어를 위한 구조적 표현에 넓게 이용되며 BNF 규칙^[10]을 이용한 다른 언어를 다루어 본 사용자들에게 친숙한 표현 방법이다.

SDL 모듈의 내부적인 동적 행위(behavior)는 CPN의 트랜지션(transition)으로 변환된다. 변환하기 전의 SDL 모듈에 대한 가정은 다음과 같다^[3].

- 1) 모듈은 유한 개의 상태집합이다: C_0, C_1, \dots, C_n
- 2) C_0 는 초기화 부분에서 정의된 초기상태이다.
- 3) SDL로 기술된 프로토콜 명세는 정규화(normalize)된 형태로 입력된다.

2. CPN의 가정 사항

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

$$= \{p_1, p_2, \dots, p_n\}, n \geq 0$$

T: 트랜지션의 집합

$$= \{t_1, t_2, \dots, t_m\}, m \geq 0$$

C: 칼라(color) 함수

$$C(p) = \sum k \cdot c, k \geq 0 \text{이고 } c \text{는 칼라}$$

I: 입력함수

O: 출력함수

μ (P → {0, 1, ... }): 토큰의 초기 상태 집합

$$P \cap T = \emptyset \text{ and } P \cup T \neq \emptyset$$

3. SDL/PR의 CPN 변환

SDL/PR로 표현된 프로토콜 명세의 CPN 변환 방법은 다음과 같다.

$$P = CUDU(IOP)$$

여기서, C는 STATE, D는 NEXTSTATE,

I는 INPUT, O는 OUTPUT, P는 PROVIDED

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

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

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

$$C(iop), \text{ if } iob \in (IOP)$$

$$I(t_{c \cdot iop}) = \{C, (IOP)\}$$

$$O(t_{c \cdot iop}) = D$$

$$\mu(p) = C(O) \cup \Sigma(IOP)$$

C(O) : 초기상태 플레이스

CPN은 플레이스 P와 트랜지션 T, 칼라 함수 C, 입력함수 I, 출력함수 O, 초기상태 μ 로 구성된다. 여기서 플레이스 P는 입출력/조건의 카테션 프러덕트(cartesian product)로 표현된 입출력/조건 플레이스와 현재 상태와 다음 상태를 나타내는 일반적인 상태 플레이스, 셀프루프를 가진 상태 플레이스의 세 종류가 있다. 본 논문에서는 실행불가능 경로를 갖지 않도록 셀프루프에 순서를 부여하여 셀프루프 플레이스로 변환하였다. 실행불가능 경로^[4]의 제거 방법과 셀프루프의 고려는 각각 연구 [5]와 [7]에 기술하였다.

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 플레이스에는 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 플레이스이면서 셀프루프를 갖지 않는다면 임의의 한 칼라가 모든 상태 플레이스에 부여되며 초기상태 플레이스만이 칼라에 따른 임의의 토큰을 가진다. 셀프루프를 갖는 상태 플레이스는 임의의 고유한 칼라를 갖는다. 만약 초기 상태 플레이스가 셀프루프를 가진다면 초기상태 플레이스는 상태 플레이스들과 달리 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.

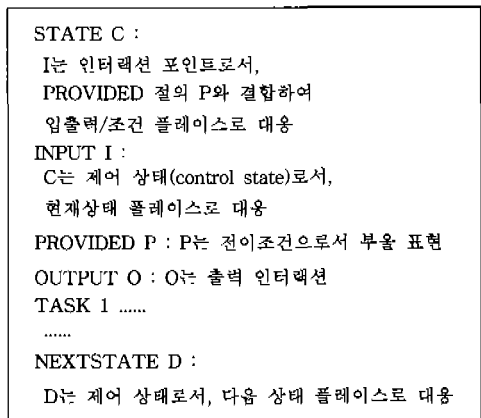


그림 1. SDL/PR의 트랜지션

그림 1과 같은 트랜지션 단위로 구성된 정규화된 형태의 SDL 형식의 프로토콜은 그림 2와 같은 CPN으로 변환된다.

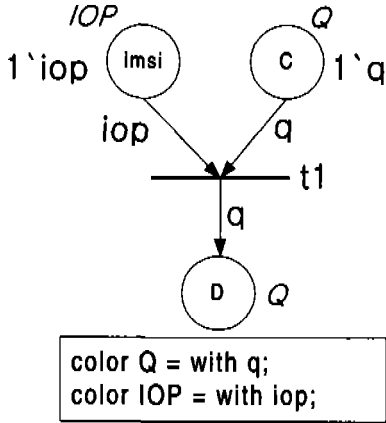


그림 2. SDL/PR의 CPN변환 예

그림 2와 같이 SDL의 입력 인터랙션과 출력 인터랙션인 INPUT 절과 OUTPUT 절은 자료흐름이 고려되도록 PROVIDED 절과 결합하여 CPN의 입출력/조건 플레이스로 변환되고, STATE 절은 현재 상태 플레이스로, NEXTSTATE 절은 다음 상태 플레이스로 변환된다. 트랜지션 t1은 현재 상태 플레이스인 C가 토큰 q를 가지고 입출력/조건 플레이스인 imsi의 토큰 iop와 함께 점화하여 다음 상태 플레이스인 D에 토큰 q를 전송하면서 전이하도록 한다. 그림 2의 하단에는 칼라에 대한 정의(declaration) 부분이 나타나 있다. 칼라 Q는 토큰 q를 가지며, 칼라 IOP는 토큰 iop를 가진다. 트랜지션의 입력이 되는 모든 플레이스에 토큰이 한 개 이상이면 그 트랜지션은 점화 가능하다.

본 논문에서는 SDL 표현을 CPN으로 변환 시 PROVIDED 절을 입출력/조건 플레이스로 표현함으로써 자료흐름을 고려한다.

IV. 사례 연구

1. 개요

본 논문에서는 SDL로 기술된 Inres 프로토콜의 Initiator 명세를 CPN으로 변환한다. 변환 수행 시 선프루프는 실행불가능 경로가 제거된 임의의 순서를 가지고 있다는 것과, 한 간선에 두 개 이상의 입출력/조건이 존재하면 임의의 한 입출력/조건을 사용한다고 가정한다. 또한 초기상태 플레이스의 indegree 에지는 사용불가능한 토큰을 갖는다. 이것은 초기상태 플레이스로 indegree 에지가 들어오게 되면 한 경로가 완성되기 때문이다.

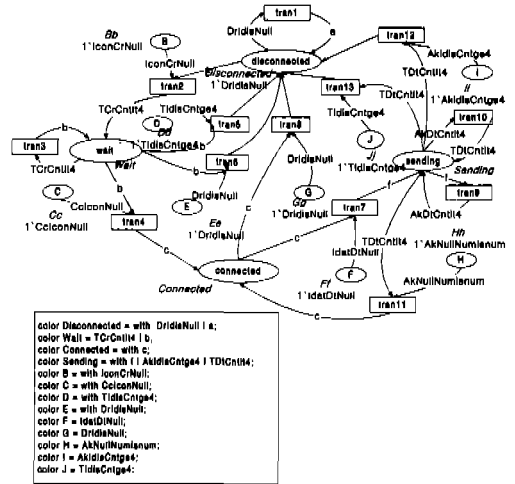


그림 3. SDL/PR로 표기된 Inres Initiator의 CPN 표현

2. SDL/PR의 CPN 변환

SDL/PR 표현으로부터 CPN 표현으로 변환된 Inres 프로토콜 Initiator는 그림 3에 나타나 있다.

그림 3에서 초기 상태 플레이스인 disconnected는 선프루프를 가지는 플레이스로서 칼라 Disconnected와 토큰 1'DrDisNull을 가지고, wait 플레이스와 sending 플레이스도 선프루프 플레이스로서 각각 칼라 Wait와 Sending을 가지며, connected 플레이스는 상태 플레이스로서 칼라인 Connected를 가진다. 초기에는 상태 플레이스 중 초기 상태 플레이스만이 토큰을 가진다. 입출력/조건 플레이스인 B, C, D, E, F, G, H, I, J는 각각의 입출력/조건에 따른 고유한 칼라와 토큰을 가진다. 토큰의 이름에 조건을 나타내는 PROVIDED 절을 고려함으로써 자료흐름을 고려한다.

3. 비교 및 결과 분석

본 논문에서는 제안된 바와 같이 SDL/PR로 기술된 프로토콜 명세를 실행불가능 경로가 제거되도록 선프루프를 고려하여 CPN 표현으로 변환한다. 변환 시 SDL/PR의 트랜지션 부분에서 PROVIDED 절을 입출력/조건 플레이스에 표현함으로써 자료흐름을 고려한다.

3.1 자료 흐름을 고려한 시험열 생성

자료흐름을 고려한 CPN 표현 방식의 프로토콜을 Design/CPN에 입력하면 그림 4, 그림 5와 같이 OG와 디스크립터가 생성된다.

OG는 초기 상태 플레이스로부터 점화가능한 트

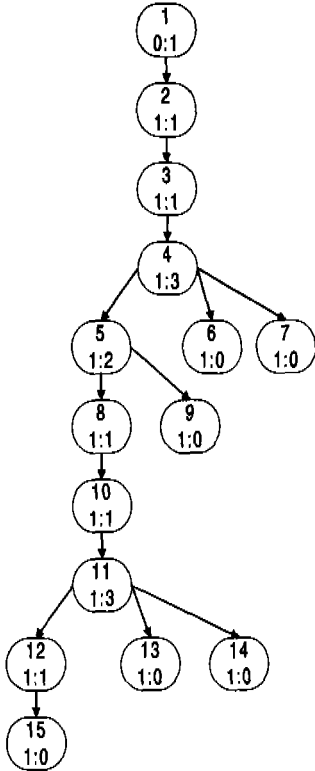


그림 4. 그림 3의 OG

랜지션을 차례로 실행한 후 다음 상태로 전이하여 다시 초기 상태로 되돌아오는 경로를 트리 형태로 표현한 것이다. OG에서 각 노드는 노드번호, 부모 노드의 개수, 자노드의 개수에 대한 정보를 가진다.

```

1
New'G 1: 1'DrIdisc
New'H 1: 1'AkNullNumisum
New'I 1: 1'AkIdisCntge4
New'J 1: 1'TIdisCntge5
New'B 1: 1'IconCrNull
New'wait 1: empty
New'C 1: 1'CelconNull
New'D 1: 1'TIdisCntge4
New'disconnected 1: 1'DrIdisa
New'E 1: 1'DrIdisb
New'connected 1: empty
New'F 1: 1'IdatDtNull
New'sending 1: empty
    
```

그림 5. 그림 4의 노드 1에 대한 디스크립터

그림 5는 노드 1의 토큰 정보를 나타내는 디스크립터로서, OG의 모든 노드는 디스크립터를 가진다^{[1][7]}. 디스크립터를 이용하여 OG의 노드와 간선에 플레이스 와 토큰을 대응함으로써 표 1과 같은 시험열을 생성한다. 시험열 생성 과정은 [1]에 나타나 있다.

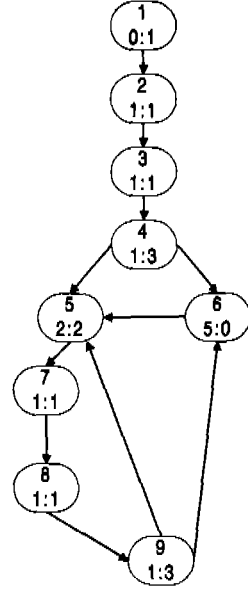


그림 6. 제어흐름을 고려한 OG

표 1은 자료흐름을 고려한 시험열로서, CPN 표현 시 셀프루프에서 실행불가능 경로가 발생되도록 순서를 부여하고, 셀프루프가 한번만 수행되도록 토큰을 제어하여 무한루프 실행을 방지하였다.

3.2 제어 흐름을 고려한 시험열 생성

```

1
New'wait 1: empty
New'connected 1: empty
New'sending 1: empty
New'disconnected 1: 1'DrIdisa
    
```

그림 7. 그림 6의 노드 1의 디스크립터

제어흐름만을 고려한 CPN 표현 방식의 프로토콜을 Design/CPN에 입력하면 그림 6의 OG와 그림 7의 디스크립터가 생성된다.

표 2는 OG와 디스크립터로부터 생성된 제어흐름을 고려한 시험열로서, 표 1에 나타난 시험열 2가 포함되지 않는다. 따라서 PROVIDED 절을 자료흐름으로 고려함으로써 더 나은 커버리지를 갖는 시험열이 생성된다.

V. 결론 및 향후 연구 방향

기존의 시험열 생성^[3]은 SDL로 기술된 프로토콜

표 1. 자료흐름을 고려한 시험열

시험열	플레이스	사용된 토큰(I/O/P)
1	disconnected disconnected wait wait disconnected	DR/IDISind/Null ICONreq/CR/Null T/CR/count<4 DR/IDISind/Null
2	disconnected disconnected wait wait disconnected	DR/IDISind/Null ICONreq/CR/Null T/CR/count<4 T/IDISind/count>=4
3	disconnected disconnected wait wait connected disconnected	DR/IDISind/Null ICONreq/CR/Null T/CR/count<4 CC/ICONconf/Null DR/IDISind/Null
4	disconnected disconnected wait wait connected sending sending sending disconnected	DR/IDISind/Null ICONreq/CR/Null T/CR/count<4 CC/ICONconf/Null IDATreq/DR/Null AK/Null/number<>number and counter<4 T/DT/count<4 AK/Null/number<>number and counter>=4
5	disconnected disconnected wait wait connected sending sending sending disconnected	DR/IDISind/Null ICONreq/CR/Null T/CR/count<4 CC/ICONconf/Null IDATreq/DR/Null AK/Null/number<>number and counter<4 T/DT/count<4 T/IDISind/count>=4
6	disconnected disconnected wait wait connected sending sending sending connected disconnected	DR/IDISind/Null ICONreq/CR/Null T/CR/count<4 CC/ICONconf/Null IDATreq/DR/Null AK/Null/number<>number and counter<4 T/DT/count<4 DR/IDISind/Null

명세를 EFSM으로 변환 후, 서브루어와 UIO를 생성하는 각각의 알고리즘에 따라 수행된다. 이러한 방법은 여러 단계의 복잡한 과정을 수작업으로 수행하기 때문에 많은 오류 가능성을 내포한다. 따라서 본 연구에서는 자동화된 툴을 이용하여 보다 간

표 2. 제어흐름을 고려한 시험열

시험열	플레이스	사용된 토큰(I/O)
1	disconnected disconnected wait wait disconnected	DR/IDISind ICONreq/CR T/CR DR/IDISind
2	disconnected disconnected wait wait connected disconnected	DR/IDISind ICONreq/CR T/CR CC/ICONconf DR/IDISind
3	disconnected disconnected wait wait connected sending sending sending disconnected	DR/IDISind ICONreq/CR T/CR CC/ICONconf IDATreq/DR AK/Null T/DT AK/Null
4	disconnected disconnected wait wait connected sending sending sending disconnected	DR/IDISind ICONreq/CR T/CR CC/ICONconf IDATreq/DR AK/Null T/DT T/IDISind
5	disconnected disconnected wait wait connected sending sending sending connected disconnected	DR/IDISind ICONreq/CR T/CR CC/ICONconf IDATreq/DR AK/Null T/DT DR/IDISind

편하고 효율적으로 시험열을 생성할 수 있도록, SDL/PR을 자료흐름이 고려된 CPN으로 변환 후, Design/CPN에 입력하여, 자동 생성된 내부 정보인 OG와 디스크립터로부터 시험열을 생성하는 방법을 제안하였다. 변환 시 PROVIDED 절을 입출력조건으로 추가하여 시험열에 자료 흐름을 고려하였고, 셀프루프를 한 번만 실행하도록 고려하여 실행불가능 경로 제거 및 무한 루프 방지의 효과를 나타내었다. CPN 표현 시 자료 흐름을 고려하면 제어흐름만을 고려한 CPN에 비해 더 나은 커버리지를 갖는 시험열이 생성된다. 제안된 방법은 자동화된 프로토콜 시험 환경의 기반을 제공한다.

향후 연구 방향은 프로토콜을 직접 CPN으로 설계하여, 프로토콜의 설계 및 구현, 시험이 하나의

통합된 환경에서 자동적으로 수행될 수 있도록 하는 것과 performance 틀을 이용하여 시스템의 성능을 분석하는 것이다.

참 고 문 헌

[1] 이현정, 우성희, 이상호, "Design/CPN을 이용한 테스트 케이스 생성," *한국통신학회 하계학술발표 논문집 17권 2호*, pp. 1285-1288, 1998.

[2] K. Jensen, *Coloured Petri Nets*, Vol.1, 2nd Ed., Springer - Verlag, 1997.

[3] 우성희, 오병호, 이상호, "Estelle로 표현된 프로토콜의 테스트 스위트 생성에 관한 연구," *정보과학회 춘계학술발표 논문집 21권 1호*, pp.463-466, 1994.

[4] 오병호, 우성희, 이상호, "조건 문맥 변수를 고려한 제어 흐름도 생성 기법," *한국통신학회 논문지*, 20권 12호, pp.3570-3579, 1995.

[5] 이현정, 우성희, 오병호, 이상호, "실행불가능 경로가 제거된 테스트 케이스 생성," *정보과학회 춘계학술발표 논문집 25권 1호*, pp. 443-445, 1998.

[6] 우성희, *EFSM으로 표현된 프로토콜의 SDL/PR 변환*, 충북대학교 석사학위 청구논문, 1993.

[7] 우성희, 이현정, "셀프루프를 고려한 테스트 케이스 생성," *청주과학대학 논문집 제24권*, pp. 161-173, 1998.

[8] 광명옥, *통합 흐름을 고려한 테스트 스위트 생성 기법*, 충북대학교 석사학위 청구논문, 1996.

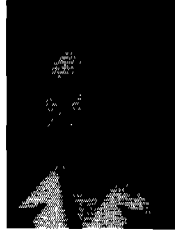
[9] 오병호, *자료 흐름의 변칙성을 고려한 프로토콜의 테스트 스위트 생성*, 충북대학교 박사학위 청구논문, 1996.

[10] CCITT, "Functional Specification and Description Language(SDL)," *Recommendation Z.100 and Annexes A, B, C, D, and E, Blue Book*, 1998.

[11] 우성희, *LTS 기반의 시험목적을 고려한 프로토콜 테스트 스위트 생성*, 충북대학교 박사학위 청구논문, 1999.

이 현 정(Hyun-jeong Lee)

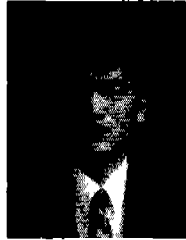
정회원



1997년 2월 : 충북대학교 컴퓨터 과학과 졸업(학사)
 1999년 2월 : 충북대학교대학원 전자계산학과 졸업(석사)
 1999년 4월~현재 : 한국전자통신연구원 연구원
 <주관심 분야> 개방형 네트워크, 개방형 통신 시스템

최 영 일(Young-il Choi)

정회원

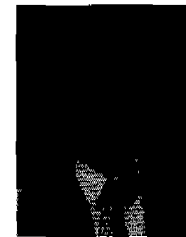


1983년 2월 : 서울대학교 전자 공학과 졸업 (학사)
 1997년 2월 : 충남대학교대학원 컴퓨터과학과 졸업 (석사)
 1984년 12월~1986년 12월 : AT&T Bell연구소
 객원연구원

1983년~현재 : 한국전자통신연구원 책임연구원
 <주관심 분야> 개방형 네트워크, 개방형 통신시스템, 통신 프로토콜

이 병 선(Byung-sun Lee)

정회원



1980년 : 성균관대학교 수학과 졸업(학사)
 1982년 : 동국대학교 전자계산 학과(공학석사)
 1982년~현재 : 한국전자통신연구원
 개방형서비스팀장

<주관심 분야> Fault-tolerant Computing, Component-based Software Engineering, Real-time Systems