

論文93-30A-12-2

고도 지능망의 CCF 기능실체 설계 및 구현

(Design and Implementation of CCF in Advanced Intelligent Network)

劉 永 民*, 趙 顯 俊*, 盧 承 煥*, 李 榮 豪**, 金 惠 鎮*,

(Young Min Yoo, Hyun Joon Cho, Soong Hwan Ro,
Hyeong Ho Lee and Duck Jin Kim)

要 約

본 논문에서는 CCITT(International Telegraph and Telephone Consultative Committee)의 지능망 구조에 대한 단기적인 연구단계인 CS-1(Capability Set-1)에 속하는 지능망 서비스들을 제공할 수 있는 지능망 교환기의 기능실체중의 하나인 CCF(Call Control Function)을 설계 및 구현하였으며, 유한상태머신(finite state machine)에 대한 테스트 시퀀스 산출방법중의 하나인 UIO(Unique Input Output) 방법을 사용하여 구현된 기능실체의 타당성을 검토하였다.

Abstract

In this paper, CCF(Call Control Function), functional entity for basic call processing in SSP(Service Switching Point), is designed and implemented. This functional entity can provide the IN(Intelligent Network) services which are included in CCITT CS-1(Capability Set-1), near-term process for IN Architecture. UIO(Unique Input Output) method, one of the test sequence generation methods for the finite state machine, is used for the implementation test of this functional entity.

1. 서 론

정보화 사회에서 통신은 핵심적인 도구가 될 것이며 고품질의 다양한 서비스가 요구된다는 점은 주지의 사실이다. 이러한 요구를 만족시키기 위해서 통신

망은 전송방식의 고속화 및 광대역화, 개인휴대통신으로의 개인화 및 다양한 서비스를 제공하기 위한 통신망의 지능화의 세가지 방향으로 진화하고 있다. 즉 광대역화, 개인화, 지능화의 세가지 관점에서 통신망 및 통신서비스가 발전되고 있으며, 이들이 복합되어 고도의 지능망 서비스가 출현할 것이다.^{[1][4]} 지능망 서비스의 도입은 미국 AT&T 사가 1980년 No.1A 교환기에서 공통선 신호방식을 통하여 800 서비스와 신용통화 서비스를 기초로 운용하면서 비롯되었으며, 1984년부터는 독립적인 SCP(Service Control Point)를 갖춘 지능망 구조에서 advanced 800 서비스가 도입되었다.^[1] 우리나라에서도 미국의 800 서비스와 유사한 광역착신과금 서비스와 신용통화 서

*正會員, 高麗大學校 電子工學科
(Dept. of Elec. Eng., Korea Univ.)

**正會員, 韓國電子通信研究所
(Electronics Telecommunications Research Institute)

接受日字 : 1992年 12月 4日

비스를 목표로 시스템을 개발하고 있으며 1991년까지 개발을 완료하고 실용화를 거친후 1994년 상용 서비스를 수행할 계획이다.^[3]

한편, CCITT(International Telegraph and Telephone Consultative Committee)에서는 SG XI/WP 4 와 SG XVIII/WP 4 에서 주도적으로 지능망 관련한 연구활동을 수행하고 있으며, 장기적인 지능망 목적에 부합되는 지능망 개념모델 연구와 현재의 통신장비를 근간으로 한 단기적인 연구를 병행하고 있다.^{[5] [6] [7] [9]}

기존 통신망에서 상용화된 부가서비스들은 대부분의 기능이 교환기에서 수행되고, 구조자체도 모듈화되지 못한 관계로 새로운 서비스의 추가나 수정이 어려웠다. 이러한 점들을 극복하기 위하여 최근들어 고도 지능망(advanced intelligent network) 구조에 대한 연구가 미국을 비롯한 선진국에서는 활발히 전개되고 있다.

본 논문에서는 CCITT 의 지능망 구조에 대한 단기적인 연구단계인 CS-1 (Capability Set - 1)에 속하는 지능망 서비스들을 제공할 수 있는 지능망교환기의 기본호처리 기능실체인 CCF(Call Control Function)를 설계 및 구현하였으며, 유한상태머신(finite state machine)에 대한 테스트 시퀀스 산출 방법중의 하나인 UIO(Unique Input Output) 방법을 사용하여 구현된 기능실체의 타당성을 실험하였다.

본 논문의 구성은 서론에 이어, 제 2 장에서는 CCF 의 설계 및 구현방법에 대하여 기술하였다. 그리고 제 3 장에서는 구현된 CCF 기능실체에 대하여 실험 및 검토하였으며, 마지막으로 제 4 장에서 결론을 맺었다.

II. CCF 의 설계 및 구현

1. CCF 의 기능

CCITT 에서는 지능망에 대한 연구를 단기적 방안과 장기적 방안으로 나누어 진행하고 있는데, 장기적 방안으로서 자리를 잡아가고 있는 것이 지능망 개념모델^[9]이다. 지능망 개념모델은 네개의 평면들로 구성되어 있으며, 각 평면은 지능망에 대한 다른 관점들을 제공하고 있다. 네개의 평면들중 물리평면은 지능망 구성을 물리적 관점에서 보여주고 있는데, 물리평면상의 핵심적 물리실체들중의 하나가 지능망교환기(SSP : Service Switching Point)이다. 지능망교환기는 통신망 사용자들에게 지능망으로의 접근을 가능하게 하는 유일한 물리실체로서, 지능망교환기의

다양한 기능들은 지능망 개념모델상의 분산기능평면상의 존재하는 몇몇 기능실체들에 의해 수행되는데, CCITT 의 CS-1 단계에서 지능망교환기에 반드시 포함되어야 할 기능실체들로는 CCF 와 SSF 가 있다. 그럼 1 은 물리평면에서 지능망교환기와 다른 물리실체간의 연결상태를 나타내고 있다.^[9]

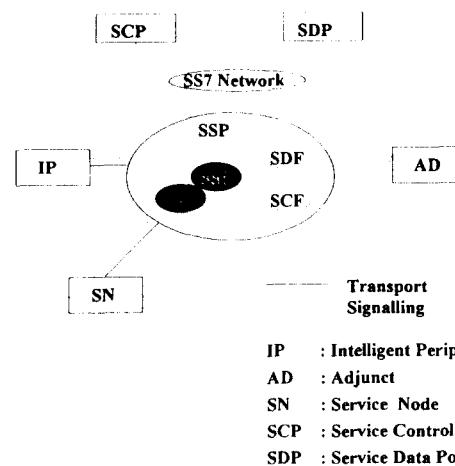


그림 1. 물리평면에서 지능망교환기와 다른 물리실체간의 인터페이스

Fig 1. Interface between SSP and other PEs in Physical Plane.

본 연구의 대상인 CCF 기능실체에서 수행되는 주요한 기능들은 다음과 같다.

- 기본호모델을 통한 기본호의 관리
- 지능망 서비스에 대한 요구를 감지하고 이를 SSF 로 통보

2. 인터페이스 규정

본 논문에서는 그림 2 에서와 같이 CCF와 직접적인 인터페이스를 갖는 기능실체인 SSF 및 지능망교환기의 Bearer 부분간의 인터페이스를 규정하였다.

1) CCF - Bearer 인터페이스

CCF 와 Bearer 간에 교환되는 메세지들은 다음과 같이 분류될 수 있다.

- 호 처리과정상 베어러에서 발생하는 메세지
Hook_Off, Collected_Info, Route_S_Failure, Called_Party_Busy, No_Answer, Answered, Connect_Failure, Connected, Disconnected, Term_Attempt, Hook_Flash, Hook_On (그림 5, 6 참조)
- 호 처리과정상 베어러를 제어하기 위해 CCF 에

서 발생하는 메세지

Connect, Route_Alert, Select_F_Present_C, T_Alert, Disconnect, Collect_Info, Orig_Dined, Invalid_Info

2) CCF - SSF 인터페이스

CCF와 SSF 간에 교환되는 메세지들은 다음과 같이 분류될 수 있다.

- 기본호모델을 제어하기 위하여 SSF에서 발생하는 메세지

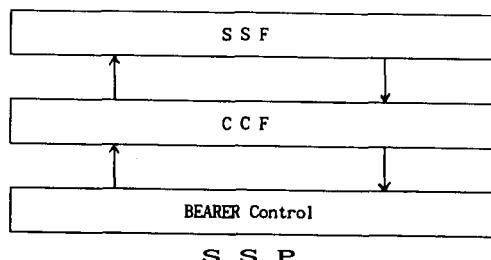


그림 2. 인터페이스 구조

Fig. 2. Structure of interface.

Collect_Info, Analyze_Info, Route_Alerting, O_Active, O_Excep, S_Faci_P_Call, T_Alerting, T_Excep (그림 5, 6 참조)

- 지능망 서비스와 관련되어 SSF에서 발생하는 메세지
Continue, Create_BCM, Destroy_BCM
- 지능망 서비스가 요구되었을 경우에 CCF에서 발생하는 메세지 Feature_Requested, SCF_Resp_Timeout

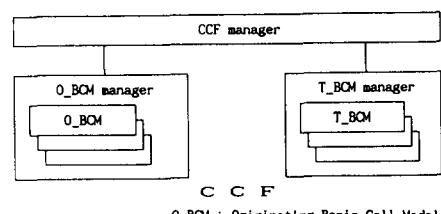
3. CCF 기능실체의 구성

CCF는 지능망교환기에서 기존의 기본호를 수용할 수 있도록 기본호에 대한 상태를 두개의 모델로 나누고 있다. 즉 기본호에 대한 상태를 일종의 유한상태머신인 발신 기본호모델과 착신 기본호모델로 모델링된다. [7] [9] 그림 3에는 CCF 능실체의 객체 관리모델을 나타내었다. 또한 그림 4에는 CCF의 기능모듈들간의 관계를 나타내고 있다. 1) CCF manager 기능모듈이 기능모듈은 CCF를 총괄적으로 관리하는 기능모듈로서, 다른 기능실체들로부터 도착하는 메세지를 읽어 들여 이것을 분석하고 해당되는 기능모듈을 동작시킨다. 또한, 필요에 따라서는 create_bcm() 함수 또는 destroy_bcm() 함수를 동작시켜 CCF의 객체인 기본호모델을 새로 생성시키거나 기존의 것을 제거한다.

CCF와 직접적인 인터페이스를 갖는 베어러부분은 지능망교환기의 물리적인 상태를 나타내는 부분이므로, 이곳으로부터 도착하는 메세지는 발신 및 착신 기본호모델에서 처리될 수 없는 형태를 갖는 경우가 대부분이다. 따라서 이 기능모듈에서 이러한 메세지들을 기본호모델이 처리할 수 있는 형태로 변형시키고, 기본호모델과 물리적인 부분을 매칭(matching)시키는 기능을 수행한다. 이를 위하여 O_BCM 또는 T_BCM 기능모듈을 동작시킬 때, 기본호모델에 대한 데이터 테이블 index 을 argument로서 제공한다. 다음은 다수의 기본호모델 인스턴스들을 제어하기 위하여 CCF에서 관리하는 데이터 테이블의 구조를 나타내고 있다.

```

struct BCM_element
{
    int scf_control:
        /* SCF 와의 제어관계 유무를 나타낸다 */
    int o_line_id: /* 발신측 line ID */
    int t_line_id: /* 착신측 line ID */
    int BCM_id:
        /* BCM ID = {1, 2, ...,
           CALLNUM} */
    int BCM_type:
        /* 기본호모델의 유형(O_BCM
           또는 T_BCM) */
    int BCM_pic: /* 기본호모델의 현재상태 */
    int EDP_N_list [NEDP]:
        /* SCF 에 의해 동적으로 arm 되어야
           할 메세지들에 대한 EDP_N 목록 */
    int EDP_R_list [NEDP]:
        /* SCF 에 의해 동적으로 arm 되어야
           메세지들에 대한 EDP_R 목록 */
}BCM_table [NBCM];
    
```



O_BCM : Originating Basic Call Model
T_BCM : Terminating Basic Call Model

그림 3. CCF의 객체 관리모델

Fig. 3. Object management model of CCF.

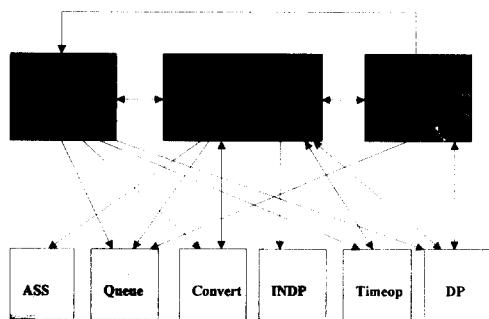


그림 4. CCF 의 기능모듈들간의 관계

Fig. 4. Relationship between CCF's function modules.

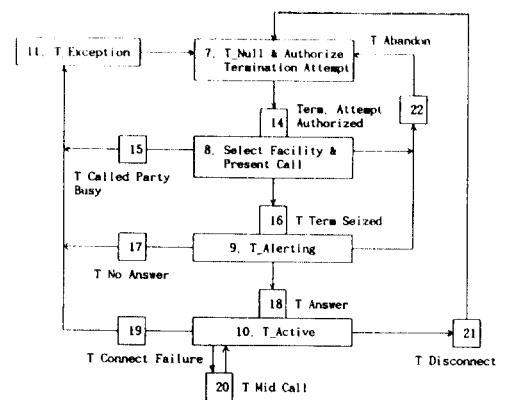


그림 6. CS-1 의 착신 기본호모델

Fig. 6. Terminating BCSM for CS-1.

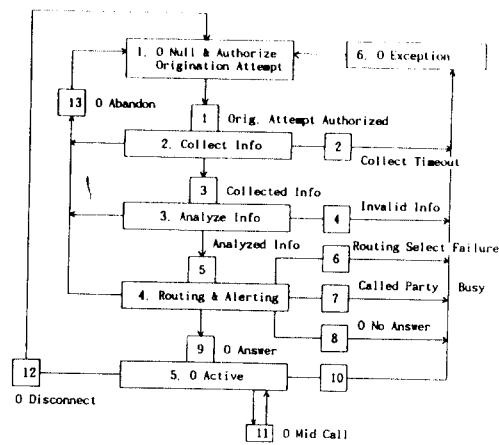


그림 5. CS-1 의 발신 기본호모델

Fig. 5. Originating BCSM for CS-1.

2) O_BCM(Originating Basic Call Model) manager 기능모듈

이 기능모듈은 CCF의 객체중의 하나인 발신 기본호모델을 관리하는 모듈로서, CCF manager 기능모듈에 의해 동작된다. CCF manager 기능모듈과 기본호모델에 관련된 데이터 테이블과 CCF에서 수신된 메세지를 공유한다. CCITT의 CS-1 단계에서 발신 기본호모델은 그림 5에서와 같이 다섯개의 상태를 갖는 유한상태머신으로 모델화되었는데, 각각의 상태에서는 기본호에 관련된 메세지들과 지능망 서비스 요구에 관련된 메세지들에 대한 사건발생을 검색하기 위하여 해당 메세지에 대한 검색점(DP : Detection Point)이 설정되었다.

CCITT의 CS-1 단계에서 발신 기본호모델은 그림 5에서와 같이 여섯개의 상태를 갖는 유한상태머신으로 모델화되었는데, 각각의 상태에서는 기본호에 관련된 메세지들과 지능망 서비스 요구에 관련된 메세지들에 대한 사건발생을 검색하기 위하여 해당 메세지에 대한 검색점(DP : Detection Point)이 설정되었다.

3) T_BCM(Terminating Basic Call Model) manager 기능모듈

이 기능모듈은 CCF의 객체중의 하나인 착신 기본호모델을 관리하는 모듈로서, CCF manager 기능모듈에 의해 동작된다. CCF manager 기능모듈과 기본호모델에 관련된 데이터 테이블과 CCF에서 수신된 메세지를 공유한다. CCITT의 CS-1 단계에서 발신 기본호모델은 그림 6에서와 같이 다섯개의 상태를 갖는 유한상태머신으로 모델화되었는데, 각각의 상태에서는 기본호에 관련된 메세지들과 지능망 서비스 요구에 관련된 메세지들에 대한 사건발생을 검색하기 위하여 해당 메세지에 대한 검색점이 설정되었다.

4) 기타 기능모듈

기타 기능모듈들의 주요한 기능들은 다음과 같다.

가) DP 기능모듈

다른 기능실체에서 도착한 메세지들에 대해 CCF에서 제공하고 있는 arming 메카니즘이 적용되도록 하는 기능들을 수행한다. 기본호모델의 각 상태에서 검색되어야 할 메세지들에 대해 호 처리과정상의 지점을 DP(Detection Point)라 칭하며, 이 DP는 다음과 같은 네가지 속성에 의해 특징지워진다.

- arming 메카니즘 : 정적 arming과 동적 arming으로 구분. 기준 : arming 조건 및 SCF로 통보되어야 하는 사건의 조건
- 제어관계 : SCF가 호 처리에 영향을 줄 수 있는가를 명시. 호 처리의 지연 : 지능망 서비스에 의해 호 처리를 지연
- 나) INDP 기능모듈

지능망교환기가 초기화될 때에 서비스 제어 시스템에 의해 주어진 자료를 기초로 하여 서비스 제어 시스템에 보고되어야 할 메세지를 기본호모델의 각 상태에 arming 한다. 즉, 기본호 모델과 정에서 고정적으로 검색되어야 하는 지능망 서비스에 관련된 메세지들에 대해 발신 및 착신 기본호모델의 각 상태에 대해 arming 을 실시한다.

다) Convert 기능모듈

베어러에서 수집한 착신측 관련정보를 분석하여 망의 번호계획에 따라 착신측의 번지를 결정한다. 착신측 관련정보는 다음의 네가지 부류로 나뉘어 통신망상의 번지로 변역된다.

- 특수번호열 : 3자리로 이루어진 특수용 전화번호.(예, 112, 114)
- 지역내 번호열 : 7자리로 이루어진 동일 지역내 전화번호(예, 925-5379)
- 지역외 번호열 : 10자리로 이루어진 다른 지역의 전화번호(예, 042-861-0784)
- 서비스 요구 번호열 : 11자리로 이루어진 동일 지역내 지능망 서비스 요구 전화번호(예, 080-925-5379)

라) Queue 기능모듈

기본호모델에서 자체적으로 메세지를 만들어 상태 천이를 수행하고자 할 경우에, 이 메세지가 마치 다른 기능실체에서 도착한 메세지인 것과 동일하게 처리할 수 있도록 한다. 즉, 기본호모델상의 한 상태에서 특정 메세지를 발생시켜 다른 상태로의 천이를 실행하고자 할 때 사용되는 기능모듈이다.

마) ASS 기능모듈

세명의 사용자가 한 통화에 속해 있으면서 선택적으로 두명의 사용자가 통화하는 형태인 associated 호에 대한 데이터를 관리하기 위한 동작들을 처리한다. 이러한 호에 대해 각각 데이터 테이블을 관리하여 two-party 호와는 다르게 호 처리를 수행하도록 한다.

바) Timeop 기능모듈

요구를 위한 메세지를 다른 기능실체로 전송한 후에 지정된 시간내에 원하는 정확한 메세지가 도착하는지를 확인하기 위한 타이머 동작들을 수행한다. 이를 위해 CCF 의 프로세스중의 하나인 CCF_Timer 프로세스와 인터페이스를 갖게 되는데, 각각의 타이머에 대해 데이터 테이블을 두어 관리한다.

4. CCF 기능실체의 구현

본 논문에서는 CCF 기능실체를 CCF_Main 및

CCF_Timer 프로세스들로 구현하였으며, multi-tasking 이 가능한 UNIX O.S 환경에서 C 언어를 사용하여 프로그램하였다. 또한, CCF 내 프로세스들 간의 메세지 교환을 위하여 프로세스간통신(IPC : Inter-Process Communication) 방법중의 하나인 socket 을 사용하였다.

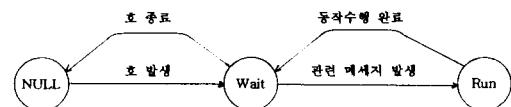


그림 7. 기본호모델 인스턴스의 상태천이도

Fig. 7. State Transition Diagram of BCM Instance.

CCF 에서 각각의 기본호모델 인스턴스(instance)들은 호에 관련된 정보들을 포함하는 데이터 테이블에 의해 관리되며, 기본호모델상의 각 상태에서 메세지 발생에 따라 관련된 데이터 테이블의 정보들이 수정된다. 기본호모델 인스턴스는 호 발생에 의해 생성된다. 해당 호에 관련된 메세지가 발생하기를 기다리는 Wait 상태로 되며, 관련 메세지가 발생하게 되면 필요한 동작을 수행하는 Run 상태로 갔다가 다시 Wait 상태로 천이한다. 그림 7 은 기본호모델 인스턴스의 상태천이도이다.

III. 실험 및 검토

1. 적합성 검토를 위한 테스터 구조

구현된 CCF 기능실체는 일종의 유한상태머신(finite state machine)으로 볼 수 있으며, 구현의 타당성을 실험하기 위해서는 먼저 테스터를 설정하

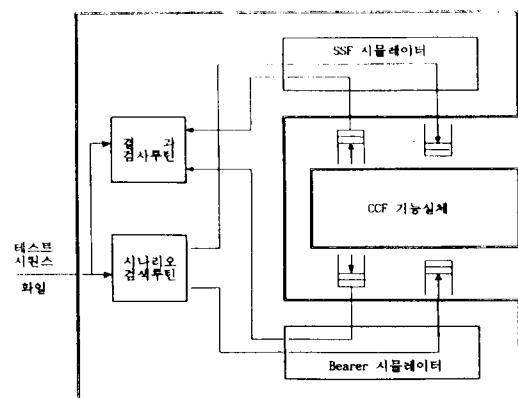


그림 8. 테스터의 구조

Fig. 8. Structure of tester.

고도 지능망의 CCF 기능실체 설계 및 구현

고, 테스트 시퀀스를 작성해야 한다. 이를 위하여, 본 논문에서는 지능망교환기의 베이어부분과 SSF 기능실체를 두개의 시뮬레이터로 설정하였다. 그림 8은 테스터의 구조를 나타내고 있다.

2. 테스트 시퀀스 산출

본 논문에서는 구현된 CCF 의 타당성을 테스트하기 위하여 유한상태머신에 대한 테스트 시퀀스 산출방법중의 하나인 UIO(Unique Input Output) 방법을 사용하였다. 한 상태에 대한 UIO 시퀀스란 일련의 입, 출력들로서 입력 시퀀스를 시스템에 인가하였을 때 시스템이 다른 상태에 있었다면 결코 나올 수 없는, 반드시 그 상태에 있었을 때만 나오는 출력 시퀀스로 구성된다. 즉 한 상태에 대한 UIO 시퀀스는 그 상태에 대한 독특한 입, 출력 특성을 가진다. [8] [10] [11]

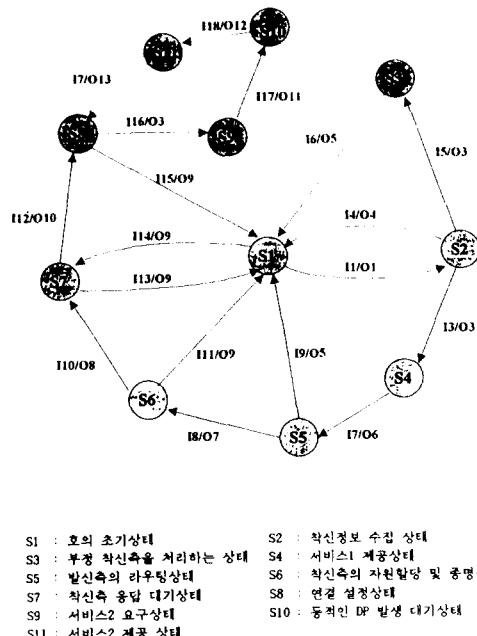


그림 9. 테스트 시나리오에 대한 상태천이도
Fig. 9. State transition diagram of the test scenario.

2) 테스트 시나리오의 작성

본 논문에서는 구현된 CCF 기능실체가 정상적으로 동작되는지를 테스트하기 위하여 다음과 같은 몇 가지 사항들을 고려하여 통합된 테스트 시나리오를 작성하였다.

- 기본호처리모델의 상태천이
- 지능망 서비스 요구에 대한 정적 armed DP의 작동

11	Org_Atempt (valid)	110	T_Term_Seized
12	Org_Atempt (invalid)	111	Called_F_Busy
13	Collect_Info (valid org & valid dest)	112	Answered
14	Collect_Info (invalid org & invalid dest)	113	Conn_Failure
15	Collect_Info (valid org & invalid dest)	114	No_Answer
16	O_Ack	115	Not_There
17	Continue	116	Hook_Flash
18	Term_Atempt	117	Collect_Data
19	Route_S_failure	118	Collected_Data
20	Collect_Info	119	T_Alert
21	Org_Denied	120	Disconnect / Disconnect_C
22	Feature_Requeste	121	Connect
23	Invalid_Info	122	Collect_Data
24	Disconnect	123	Collected_Data
25	Route_Alert	124	Send_Alarm
26	Select_F_Present_C	125	

선분	현재상태	다음상태	입 역	출 역
1	S1	S2	11	01
2	S1	S1	12	
3	S2	S4	13	03
4	S2	S1	14	04
5	S2	S3	15	03
6	S3	S1	16	05
7	S4	S5	17	06
8	S5	S6	18	07
9	S5	S1	19	05
10	S6	S7	110	08
11	S6	S1	111	09
12	S7	S8	112	010
13	S7	S1	113	09
14	S7	S1	114	09
15	S8	S1	115	09
16	S8	S9	116	03
17	S9	S10	117	011
18	S10	S11	118	012
19	S11	S8	17	013

그림 10. 테스트 시나리오에 대한 상태천이표

Fig. 10. State transition table of the test scenario.

[1] Org_Atempt > Collect_Info	[25] Term_Atempt > Select_F_Present_C
[2] Collect_Info > Feature_Requeste	[26] T_Term_Seized > T_Alert
[3] Continue > Route_Alert	[27] Answered > Conn_C
[4] Term_Atempt > Select_F_Present_C	[28] Hook_On > Disconnect / Disconnect_C
[5] T_Term_Seized > T_Alert	[29] Org_Atempt > Org_Denied
[6] No_Answer > Disconnect / Disconnect	[30] Org_Atempt > Collect_Info
[7] Org_Atempt > Collect_Info	[31] Collect_Info > Feature_Requeste
[8] Collect_Info > Feature_Requeste	[32] O_Ack > Disconnect
[9] Continue > Route_Alert	[33] Org_Atempt > Collect_Info
[10] Term_Atempt > Select_F_Present_C	[34] Collect_Info > Feature_Requeste
[11] T_Term_Seized > T_Alert	[35] Continue > Route_Alert
[12] Conn_Failure > Disconnect / Disconnect	[36] T_Term_Seized > Disconnect
[13] Org_Atempt > Org_Denied	[37] Org_Atempt > Org_Denied
[14] Org_Atempt > Collect_Info	[38] Org_Atempt > Collect_Info
[15] Collect_Info > Feature_Requeste	[39] Collect_Info > Feature_Requeste
[16] Continue > Route_Alert	[40] Collect_Info > Collected_Data
[17] Term_Atempt > Select_F_Present_C	[41] Term_Atempt > Select_F_Present_C
[18] Called_F_Busy > Disconnect / Disconnect	[42] T_Term_Seized > T_Alert
[19] Org_Atempt > Collect_Info	[43] Answered > Conn_C
[20] Collect_Info > Invalid_Info	[44] Hook_Flash > Feature_Requeste
[21] Org_Atempt > Org_Denied	[45] Collect_Data > Collected_Data
[22] Org_Atempt > Collect_Info	[46] Collected_Data > Collected_Data
[23] Continue > Route_Alert	[47] Continue > Send_Alarm
[24] Term_Atempt > Select_F_Present_C	[48] Hook_Flash > Feature_Requeste

그림 11. 테스트 시나리오에 대한 입, 출력 시퀀스
Fig. 11. Input/output sequence of the test scenario.

- 지능망 서비스 요구에 대한 동적 armed DP의 작동
- 통합된 테스트 시나리오를 기초로 CCF에 대한 상태천이도가 만들어지게 되고, 상태천이도에서 상태천

이가 발생하는 경우를 중심으로 하여 상태천이표를 작성한다. 작성된 상태천이표를 테스트 화일로 작성한 후, 이것을 시나리오 번역 프로그램에 입력으로 넣게 되면 원하는 입, 출력 테스트 시퀀스를 얻는다. 그림 9는 테스트 시나리오에 대한 상태천이도를 나타내며, 그림 10은 테스트 시나리오에 대한 상태천이표를 나타낸다.

3. 테스트 및 검토

UIO 방법을 통하여 얻은 입, 출력 시퀀스는 또 다시 한번의 번역과정을 갖는다. 즉 실제의 입력 메세지들과 출력 메세시들을 나타내는 시퀀스들로 바뀌게 되는 것이다. 이리하여 최종적으로 입, 출력 시퀀스로 구성된 테스트 화일을 테스터에서 읽어 들이게 되므로써 테스트가 수행되는 것이다. 그림 11은 통합된 테스트 시나리오에 대한 입, 출력 시퀀스이다.

실험은 테스터가 테스트 화일로부터 입력 메세지를 읽어 들인 후, CCF로 첫번째 입력을 넣은 뒤에 원하는 출력이 제대로 나오는 가를 확인함으로써 시작되는데, 만약 정확한 출력이 지정된 시간내에 올바른 시뮬레이터로 나오지 않는 경우에는 테스트가 중단된다. 입력을 인가한 후에 정확한 출력이 올바른 시뮬레이터로 나올 경우에는 또 다시 테스터가 테스트 입, 출력 시퀀스 화일로부터 다음 입력 메세지를 읽어 들여 상기한 과정과 동일한 방법으로 구현된 CCF에 대한 테스트를 반복하여 테스트 입, 출력 시퀀스 화일이 종료될 때까지 테스트를 계속하게 된다.

본 논문에서는 CCF 기능실체에 발생할 수 있는 입력 메세지들중에 일부만을 테스트 시퀀스로 작성하여 실험하였다. 통합된 테스트 시나리오는 교환기에서 발생할 수 있는 일반적인 메세지들로 구성되었으며, 테스트 시나리오에 대한 입력 시퀀스를 CCF 기능실체에 인가하였을 때, 설정된 입력 시퀀스에 대한 정확한 출력 시퀀스가 나오는 것이 확인됨으로써, 구현된 CCF 기능실체의 기능들이 정상적으로 구현되었음을 알 수 있었다.

IV. 결론

본 논문에서는 CCITT CS-1 단계에 적합한 지능망교환기의 주요한 기능실체인 CCF 기능실체를 설계 및 구현하였다. 구현된 CCF에 대한 테스트는 CCF에 발생할 수 있는 모든 경우를 테스트의 대상으로 하여야 하겠으나, 모든 경우에 대해 테스트를 수행한다는 것은 불가능한 일이므로, 본 논문에서 구현된 CCF 기능실체에 대해 교환기의 동작수행상 일반적으로

로 발생할 수 있는 사전발생 모델을 통합된 테스트 시나리오로 작성하고 이를 CCF 기능실체에 적용시켜 보았다.

작성된 테스트 시나리오를 기초로, 유한상태머신에 대한 테스트 시퀀스 산출방법중의 하나인 UIO 방법을 사용하여, 통합된 테스트 입, 출력 시퀀스를 산출하였다. 이렇게 산출된 테스트 시퀀스를 CCF에 인가하여 실험한 결과, 구현된 CCF 기능실체가 정상적으로 작동됨을 알 수 있었다.

기존의 교환기에서 새로운 서비스 도입시에 생겨나는 주요한 문제점인 기존 서비스 제공 소프트웨어 모듈에 대한 수정 및 새로운 모듈의 추가는 기존의 통신망을 지능화한다면 완전히 해소될 수 있을 것이다. 이러한 관점에서 현재 우리나라에서는 아직 초기단계에 있는 지능망에 대한 연구는 좀 더 활발히 추진되어야 할 필요성이 있다고 본다. 본 논문은 지능망의 핵심적인 물리실체인 지능망교환기에서 기본호처리를 담당하는 CCF 기능실체를 설계 및 구현하였으며, 지능망개념모델의 분산기능평면상에서 기능실체를 구현하였기 때문에 물리실체들간에 고려되어야 할 프로토콜을 고려하지는 않았다. 이러한 연구를 바탕으로, 앞으로 지능망교환기에 포함되어야 할 또 다른 주요 기능실체인 SSF에 대한 설계 및 구현을 위한 연구가 수행되어야 할 것이다. 또한 상이한 물리실체간의 정보흐름을 위하여 프로토콜에 적용시키기 위한 연구가 필요할 것으로 판단된다.

* 본 논문은 한국전자통신연구소의 위탁연구 결과입니다.

参考文献

- [1] 홍진표, “지능망 기술동향”, 전자공학회지 vol. 18 no. 1, 1991. 1.
- [2] W. D. Ambrosch, A. Maher, B. Sassecer, *The Intelligent Network*, Springer - Verlag, 1989
- [3] 이영희와 3인, “우리나라 지능망 개발 현황”, 전자공학회지 Vol. 18 No. 1, 1991. 1
- [4] 김용기, 표현명, “개인통신 서비스 구현을 위한 지능망 기술”, 전자공학회지 제 19권 제 9호, 1992. 9.
- [5] 최용원, 지능망 서비스의 표준화 조사 연구, CCITT XI 연구위원회 연구보고서, 1991. 12.
- [6] 김덕진, 광대역 ISDN과 지능망을 위한 신호

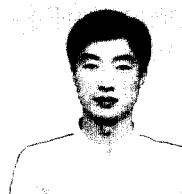
- 방식 연구, 고려대학교 부설 정보통신기술 공
동연구소 연구보고서, 1992.3.
- [7] 한국전자통신연구소 지능망연구부, 차세대지
능망 개념서, 1991.12.
- [8] 기장근, “프로토콜 적합성 검증을 위한 최적
시험절차 추출 자동화에 관한 연구”, 박사학위
논문, 고려대학교, 1991.12.
- [9] CCITT Draft Recommendations(Q.1200,
Q.1201, Q.1202, Q.1203, Q.1204, Q.
1205, Q.1208, Q.1211, Q.1213, Q.1214,
Q.1215, Q.1218, Q.1219, Q.1290, Q.

1400), Sep 27, 1991.

- [10] Krishan Sabnani, Anton Dahbura, “A
Protocol Test Generic Procedure,”
Computer Network and ISDN System
15 (1988) pp.285-297
- [11] Raymond E. Miller, Sanjoy Paul,
“Generating Minimal Length Test
Sequences for Conference Testing of
communication Protocols,” IEEE
INFOCOM '91, Vol.2, 1991, pp.970-979

著者紹介

劉永民(正会員)



1966年 1月 26日生. 1989年 3月
육군사관학교 전자공학과 졸업(학
사). 1993年 2月 고려대학교 대학
원 전자공학과 졸업(석사). 1993
年 3月 ~ 현재 육군통신학교 교
관. 주관심분야는 ATM, 지능망
등임.

盧承煥(正会員)



1962年 8月 19日生. 1987年 8月
고려대학교 전자공학과 졸업(공학
사). 1989年 8月 고려대학교 대학
원 전자공학과 졸업(공학석사).
1993年 8月 고려대학교 대학원 전
자공학과 졸업(공학박사). 주관심
분야는 정보통신, CAT(Computer Aided Test) 등
임.

金惠鎮(正会員) 第 29 卷 A 編 第 8 號 參照

현재 고려대학교 전자공학과 교수

趙顯俊(正会員)



1963年 1月 28日生. 1985年 2月
고려대학교 전자공학과 졸업(학
사). 1987年 8月 고려대학교 대학
원 전자공학과 졸업(석사). 1991
年 9月 ~ 현재 고려대학교 대학
원 전자공학과 박사과정. 1989年
1月 ~ 1991年 8月 (주) 쌍용컴퓨터 시스템연구소
연구원. 주관심분야는 프로토콜, 지능망, 광대역
ISDN 등임.

李榮豪(正会員)



1955年 4月 3日生. 1977年 2月
서울대학교 공업교육과(학사).
1979年 2月 한국과학기술원 전기
및 전자공학과(공학석사). 1983年
8月 한국과학기술원 전기 및 전자
공학과(공학박사). 1983年 9月 ~
1984年 11月 한국전자통신연구소 교육훈련실 선임연
구원. 1984年 12月 ~ 1986年 11月 미국 AT&T
Bell Laboratories 방문연구원. 1986年 12月 ~ 현
재 한국전자통신연구소 (ETRI) 신호장치개발실장 역
임. 현재 교환서비스 연구실장(책임연구원). 주관심
분야는 교환기술, 고도 지능망, 광대역 ISDN, 음성
신호처리 등임.