
 <主 題>

CTI 기술의 표준화 동향

정 담, 김대중, 김숙영, 강우식
(삼성전자 정보통신연구소)

 □ 차 례 □

- | | |
|--|--|
| I. 서 론
II. CTI 개념
III. CTI 링크
IV. SCSA/MVIP | V. CT API
VI. CTI 통합 모델
VII. 결 론 |
|--|--|

I. 서 론

우리 생활에서 컴퓨터와 전화는 빼놓을 수 없는 가장 대표적인 통신 수단이다. 전화는 서로 떨어져 있는 두 사람간 가장 편리하고 신속하게 음성 정보를 전달할 수 있는 수단이 되었으며, 컴퓨터는 자체적으로 강력한 정보처리 능력을 가졌을 뿐만 아니라 인터넷에 연결되어 임의의 정보 제공자로부터 문자, 음성, 화상, 영상 등 여러 가지 형태의 복잡한 멀티미디어 정보를 손쉽게 얻을 수 있도록 해준다. 이제까지 전화는 그 기능이 크게 변하지 않은 가운데 널리 퍼져 핵심적인 통신수단으로 자리잡은 반면에 컴퓨터는 급속도로 발전하는 정보 처리 능력 및 네트워킹 기술로 서비스 제공 능력이 커지고 있지만 아직까지도 전화만큼 널리 보급되지는 않았다.

그러나 서비스가 복잡하고 다양해지면서 음성 사서함, 무인자동교환안내, 팩스정보시스템, 통합메시지시스템 등과 같이 컴퓨터와 전화가 결합되어 서비스를 제공할 필요가 생겼다. 이와 같은 복합 서비스를 위하여 전화를 위한 교환기의 호 제어 기능과 정보처리를 위한 컴퓨터의 기능을 링크 시키는 방법이 CTI(Computer Telephone Integration) 기술이다. CTI 기술은 공중망 및 사설 교환망에 적용되어 텔레마케팅, 고객지원 서비

스 등 다양한 서비스를 제공하게 된다.

초창기에 사설 통신망 분야에서는 이미 사설 교환기 업체들과 컴퓨터 시스템 업체들이 상호 협조하여 사설 교환기와 밀접합된 음성 사서함 등을 개발하였다. 그러나 사설 교환기 업체나 컴퓨터 시스템 업체들마다 고유의 프로토콜을 사용하였으므로 대상 교환기나 음성 시스템이 변경될 때마다 비슷한 작업을 반복해야 하는 문제가 있었으며 따라서 그 응용 범위로 제한될 수밖에 없었다. 이러한 문제를 극복하고 시장 영역을 확대하기 위해서는 교환기와 컴퓨터 시스템 사이의 인터페이스 표준화가 절실히 요구되었다. 그 결과 국가별, 지역별, 국제적으로 몇 개의 그룹이 형성되어 표준화 작업이 진행되고 있다.

컴퓨터와 전화를 결합함으로써 얻어지는 경제적인 효과가 증명되면서 기업 전산망 담당자들은 수익 증대, 비용 절감, 고객 서비스 향상, 보다 쉬운 전산망 관리 등 각종 장점을 열거하며 전화망과 컴퓨터 통신망의 통합을 주장하고 있다.

또한, 최근 들어 컴퓨터 시스템 업체들은 전화 통신 장치, 미디어 처리 장치 등을 개발하면서 UNPBX의 시대를 예고하고 있는 등 CTI 개념이 단순히 전화망에 정보 처리 기능을 강화하기 위해 컴퓨터를 연결하는 수준에서 벗어나 이제는 컴퓨터 통신망과 전화망의 통합 개

념으로까지 확대되고 있다.

CTI와 관련된 표준은 크게 세부분에 걸쳐 진행되고 첫번째는 교환기로 대표되는 스위칭 영역의 기능과 컴퓨터에서 주로 일어나는 컴퓨팅 영역의 기능을 링크시키기 위해 인터페이스를 정의하는 부분으로 CTI 링크라고 하는 부분이다. 이 부분은 이미 오래전부터 유럽의 컴퓨터 제조업체들이 주축이 된 ECMA (European Computer Manufactures Association)에서 CSTA(Computer Supported Telecommunications Applications) 분과를 통해 표준화를 진행하였는데 그 결과 대표적인 Call 모델은 이후 다른 CTI 연구에 적지 않은 영향을 미쳤다. CTI 링크에 관한 표준은 사실 교환기와 컴퓨터 서버간의 연동을 위해 주로 사용된다.

두번째는 컴퓨터 제조업체 및 통신 미디어 처리 장치를 개발하는 업체들에 의해 활발히 표준화가 진행되고 있는 하드웨어 버스 방식에 관한 부분인데 SCSA (Signal Computing System Architecture), MVIP (Multi-Vendor Integration Protocol) 등이 대표적이다. 기존의 컴퓨터를 이용한 전화 통신 처리를 전문으로 하는 사업자들의 시스템은 같은 기능을 수행하지만 서로 다른 로컬 버스를 가지고 있고 기타 정보처리 방식도 달라서 호환이 되지 않는 것이 사실이었다. 이러한 문제점을 극복하기 위해 각각의 시스템간 정보 처리를 위한 표준의 고속 전송 방식이 필요하게 되었고 이에 따라 개방형 구조를 갖는 하드웨어 및 소프트웨어가 개발되고 있으며 최근의 컴퓨터, 전화 환경에서 경제적이며 호환성이 뛰어난 시스템의 출현을 가능하게 하고 있다. 이와 같은 구조를 바탕으로 현재 ECTF(Enterprise Computer Telephony Forum)에서는 소프트웨어 전반적인 구조에 관하여 인터페이스를 정의하는 등 산업 표준화를 진행하고 있다. SCSA, MVIP를 기반으로 전화망을 연결하는 컴퓨터 시스템 업체들은 기존의 사설 교환기 기능을 상용 컴퓨터에 구현함으로써 UNPBX 개념을 실현하고 있다.

세번째는 컴퓨터에서 전화망을 제어하기 위한 API(Application Programming Interface)부분인데 Microsoft와 Intel에서 주도하는 TAPI(Telephone Application Programming Interface), Novell과 AT&T에서 주도하는 TSAPI(Telephony Services Application Programming Interface) 등이 대표적이

다. 초창기에 TAPI는 PC에서 모뎀으로 연결된 전화를 제어하기 위해 시작되었으며 TSAPI는 PBX(Private Branch eXchange)와 LAN(Local Area Network)이 텔리포니 서버로 연결된 환경에서 3rd-Party Call 제어 방식을 지원하기 위해 시작되었다.

본 고에서는 CTI를 컴퓨터와 전화 통신의 결합이라는데 초점을 맞추어 CTI와 관련된 표준화 동향에 대하여 포괄적으로 다루도록 한다. 특히, 최근 들어 CTI와 관련된 여러 가지 기술이 두서없이 소개되고 있는데 본 고에서는 앞에서 언급한 바와 같이 CTI 링크, 컴퓨터에서 통신 처리 장치를 위한 버스 방식, API 그리고 CT(Computer Telephony) 통합 모델 등으로 분류하여 기술하고자 한다.

II. CTI 개념

CTI란 CT 개념의 일부인데 컴퓨터와 전화망이 상호 협조하여 서비스를 제공하는 것을 말한다. 예를 들어 전화 벨이 울리고 있는 중에 PC 화면에는 발신자 정보가 자동적으로 디스플레이되어 전화를 받기 전에 발신자를 확인할 수 있다면 이것은 CTI의 결과라고 할 수 있다.

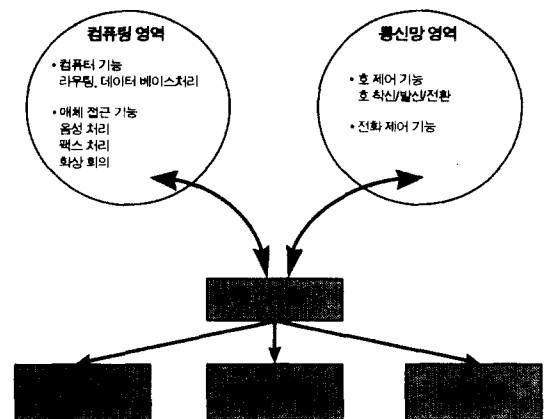


그림 1. CTI의 개념

CTI를 위해서는 컴퓨터와 교환기 또는 전화기간 물리적인 연결이 필요하며 컴퓨터 기반의 애플리케이션이 전화 서비스를 요청하고 전화 호를 추적하는 등 전화 서

비스 관련 동작을 수행할 수 있어야 한다. CTI라는 말은 1980년대말 컴퓨터와 교환기간 링크 제품이 발표되면서 일반화되었다. 초기의 CTI 링크는 단순히 하나의 교환기와 하나의 컴퓨터를 연결하는 것이었으며 컴퓨터에서는 단일 애플리케이션이 운용되었다. 그러나 이제 많은 컴퓨터가 상호 작용하는 클라이언트 서버 컴퓨팅 환경이 도입되면서 CTI 구조도 많이 변하게 되었다.

그림 1은 CTI의 개념을 설명하고 있다. 컴퓨팅 영역과 스위칭 영역의 기능을 CTI로 통합하고 이를 기반으로 스크린 팝업(Screen Pop-up) 서비스나 텔레마케팅 서비스, 안내대(Help Desk) 서비스 등을 제공할 수 있다.

일반적으로 CTI를 구현하는 방법에는 그림 2의 1st-Party Call 제어 방식과 그림 3의 3rd-Party Call 제어 방식이 있다. 1st-Party Call 제어 방식에서는 전

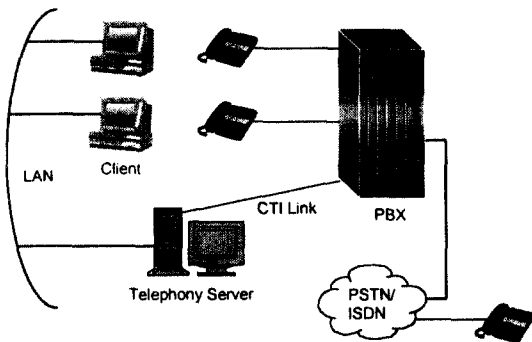


그림 2. 3rd-Party Call 제어 방식

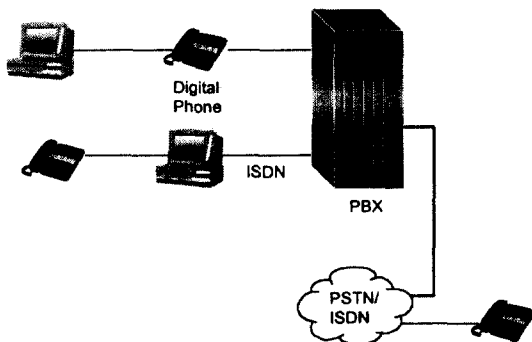


그림 3. 1st-Party Call 제어 방식

화기 인터페이스 카드를 PC에 삽입해서 항상 컴퓨터가

전화기 라인을 모니터링할 수 있도록 하는 것으로 점차 많은 호응을 얻고 있는 방법이다. 1st-Party Call 제어 방식은 모뎀을 접속한 PC 또는 ISDN에 접속한 PC 등에 적합한 방식이다.

3rd-Party Call 제어 방식은 전화 교환기를 컴퓨터 시스템에 CTI 링크로 연결하는 방법이다. CTI 링크를 통해서 호 이벤트 및 관련 정보가 전달된다. 이러한 방법은 서버를 기반으로 하는 것으로서 특히 CTI 링크를 통해 모든 전화 제어 정보가 전달되기 때문에 비교적 대용량의 콜 센터(Call Center) 등 그룹 사용자를 위한 서비스 제공에 적합하다. 1980년대 말부터 몇몇 스위치 및 컴퓨터 회사들이 CTI 링크를 제공하고 있으나 큰 용량의 콜 센터 운용 시스템을 제외하고는 거의 사용되고 있지 않았으나 요즘 링크당 가격이 하락되면서 상황이 많이 변하고 있다. CTI 링크 제품에는 IBM의 Call-Path, Lucent의 PassageWay, Nortel의 Meridian Link, Siemens의 CallBridge, Dialogic의 CT-Connect 등이 있다.

3rd-Party Call 제어 방식에서는 단일 CTI 링크를 통해서 모든 전화기를 제어할 수 있는 반면에, 1st-Party Call 제어 방식에서는 단일 사용자의 전화기 라인만을 제어할 수 있다. 따라서 1st-Party Call 제어 방식에 적용되는 기능은 상대적으로 적으며 호 제어 범위도 하나의 전화로 제한된다.

III. CTI 링크

CTI 링크는 3rd-Party Call 제어 방식에서 텔레포니 서버와 전화 교환기 사이에 전화 회선 모니터링 정보 및 호 정보, 제어 정보 등을 주고 받을 수 있도록 해주는 부분이다. CTI 링크는 물리적 연결 측면과 메시지 프로토콜 측면에서 살펴볼 수 있다. 물리적 연결 측면에서 볼 때 간단하게는 RS-232C로 연결되기도 하고 Ethernet과 같은 LAN으로 연결하기도 한다. 텔레포니 서버와 PBX를 연결할 때 기본 통신 프로토콜은 주로 X.25 또는 TCP/IP를 사용하고 있으며 국선 교환기의 경우 D채널 프로토콜이나 No.7 프로토콜을 사용하기도 한다.

전화 교환기와 컴퓨터 시스템간 회선 모니터링, 호

제어 등의 정보를 전달하기 위하여 교환기 업체들은 지금까지 고유의 CTI 링크 프로토콜을 만들어 사용해왔다. 심지어는 동일한 교환기 업체에서도 교환기에 따라 여러 가지의 상이한 프로토콜을 만들어 사용하기도 했다. 이것은 컴퓨터 시스템과 교환기를 링크시키고자 할 때 일일이 인터페이스를 맞추어 주어야 한다는 것의 미한다.

따라서 이러한 문제점을 극복하기 위하여 ECMA에서 CSTA 표준안을 만들었다. CSTA 표준안은 Ericsson, Siemens 등 여러 교환기 제조회사에서 채택하여 사용하였으며 TSAPI와 같은 API도 CSTA 표준안을 기반으로 만들어졌다. ANSI(American National Standards Institute)에서도 SCAI(Switch-Computer Applications Interface) 표준을 만들었으나 CSTA 표준안만큼 널리 이용되고 있지는 않다.

한편, 일본에서는 SCAI를 근간으로 한 TPCI(TTC PBX-Computer Applications Interface) 표준을 만들었다. CSTA와 SCAI가 상호 호환성이 없게 되자 ITU-T에서는 국제적 표준으로 TASC(Telecommunications Applications for Switches and Computers) 권고안을 만들었으나 관심을 끌지 못하고 1994년에 연구가 중단된 상태다.

3.1 CSTA

유럽의 정보, 통신, 멀티미디어 표준을 제정하기 위해 1961년에 설립된 ECMA에서 기존의 전화망과 컴퓨터를 결합하여 다양한 응용 서비스를 제공하려는 목적으로 발표한 대표적인 CTI 권고안이 CSTA이다.

3.1.1 CSTA 구성

현재 ECMA의 구성은 각종 표준화를 담당하는 12개의 TC(Technical Committee)와 표준안을 제정하기 위해 각 TC에 속한 다수의 TG(Task Group)로 구성되어 있고, 이들 그룹에서 226 개의 표준안과 69개의 TR(Technical Report)을 30년 넘도록 전 세계에 널리 배포하고 있다.

CSTA는 ECMA TC32-TG11에 속하는 CTI 표준화 단체로서 1988년 6개의 통신업체(British Telecom 등), 컴퓨터 제조 업체들(Digital 등)이 모인 것이 시발점이 되었다. 이들의 목적은 기존의 전기통신망과 컴퓨

터 망에서 스위치와 컴퓨터를 연결하고 통합하는 데 소요되는 비용과 시간을 절약하자는 것이었다. 즉, 스위치와 컴퓨터 사이의 링크 표준을 제정하자는 것이었다. 또한, 이들의 공통 관심사는 사실 통신망에서의 텔레포니 응용 프로그램을 갖는 범용 컴퓨터의 응용 및 표준화에 있었다.

TC32-TG11 CSTA에서는 컴퓨터 망과 전기 통신망 간의 CTI의 표준을 정의하며 이에 해당하는 TR로는 TR/52와 TR/68이 있다. TR/52에서는 CSTA의 구조 프레임워크를 정의하고 있고, CSTA Phase I의 구조와 서비스를 기술한 ECMA-179와 CSTA의 응용 계층 프로토콜을 기술한 ECMA-180이 있다. 기존의 CSTA를 확장한 CSTA Phase II는 TR/68 CSTA Scenarios에서 정의되고 있다. CSTA Phase II의 구조와 서비스는 ECMA-217에서 기술되고 응용 계층 프로토콜은 ECMA-218에서 기술되어 있다. CSTA Phase II를 확장한 Phase III의 작업은 현재 진행 중이다. 그림 4는 현재 CSTA의 구성을 보여준다.

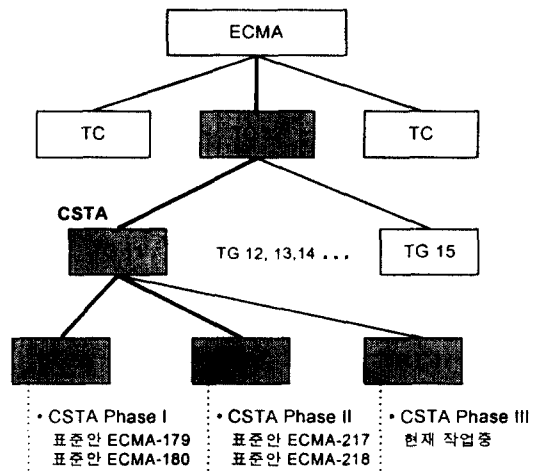


그림 4. CSTA 구성도

TR/52에서는 CSTA의 구조 프레임워크를 제시하고, 세부적으로는 클라이언트/서버 구조 모델과 스위칭 기능의 동작 모델, 스위칭 기능과 컴퓨팅 기능사이의 애플리케이션 인터페이스를 정의한다. 그리고, TR68에서는 호 제어와 모니터링에 따른 CSTA 서비스의 실제 응용을 다루고 있다.

현재 CSTA 구성원으로는 컴퓨터, 스위치 제조업체, 망 운영 사업자, 소프트웨어 제조업체 등으로 구성되어 있고, 주요 구성원으로는 Philips, Ericsson, Novell, Tandem, BT, Nortel, Microsoft, IBM, Dialogic, Alcatel, Callscan, Lucent 등이 있다.

3.1.2 기능 구조

CSTA 기능 구조를 정의하는 목적은 물리적인 구현과는 관계없이 컴퓨팅 기능과 스위칭 기능사이의 서비스 경계를 통하여 CSTA 애플리케이션 인터페이스를 정의하는데 있으며, CSTA 애플리케이션은 컴퓨팅 망을 기반으로 하는 컴퓨팅 요소와 전기통신망을 기반으로 하는 스위칭 요소에 의해서 지원된다.

일반적으로 컴퓨팅 기능은 컴퓨터망에서 여러 컴퓨터에 의해서 구현되고, 스위칭 기능은 전기통신망의 스위치에서 구현된다. 그러나, CSTA 애플리케이션 기능은 분산된 자원을 사용하므로 분산된 엔티티간에 상호작용이 가능해야 한다.

그림 5에서는 ODP(Open Distributed Processing) 환경에서 CSTA 요소간의 상호관계를 보여주고 있다.

- 프로세싱 요소(Processing Component) : CSTA 에서 정의된 기능을 지원하는 애플리케이션
- 통신 요소(Communication Component) : 메시지 교환을 지원하는 CSTA 서비스
- 네트워킹 지원(Networking Support) : 하위 계층에서 CSTA 서비스를 지원

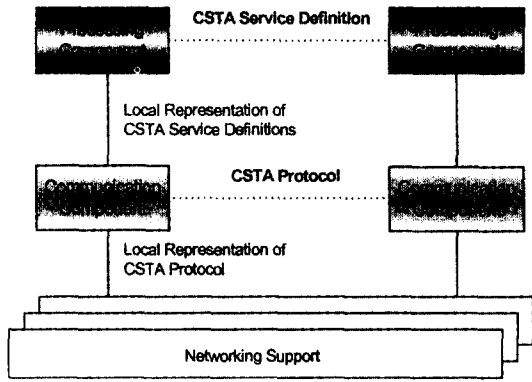


그림 5. CSTA 요소들의 상관관계

OSI 관점에서 볼 때 서비스는 하위 계층에서 인접한

상위 계층에 제공하는 기능으로 정의되지만 CSTA에서는 동급(Peer-to-Peer) 애플리케이션간에 주고 받는 기능들을 서비스라고 정의한다. CSTA 서비스는 스위칭 영역에서 실제로 전기통신 가입자들에게 제공하는 서비스에 대해 독립적으로 동작하도록 설계된다. 따라서 전기통신망과 사용자의 인터페이스에 관계없이 CSTA 서비스를 요구할 수 있고 스위칭 영역에서는 요구된 서비스의 지원 여부를 결정하게 된다. 예를 들어 호 설정과 같은 CSTA 서비스에 대해 아날로그 망과 ISDN 망에서는 각각의 고유한 프로토콜을 통해 호 설정 서비스를 구현하게 된다. CSTA에서는 요청한 CSTA 서비스를 스위칭 영역에서 어떠한 방법으로 수행하는가에 대한 세부사항에 대해 정의하지 않는다.

CSTA 애플리케이션을 지원하기 위한 통신 메카니즘은 클라이언트/서버 관계로 모델링된다. 프로세싱 컴포넌트가 서비스를 요청하면 통신을 담당하는 로컬 컴포넌트인 클라이언트가 서버에게 관련 서비스를 요청하게 된다. 애플리케이션 수준에서의 클라이언트 서버간 통신은 OSI 7 계층 모델에서 애플리케이션 계층에 해당한다.

CSTA 구조가 양방향 서비스를 지원하므로 클라이언트/서버의 관계는 그림 6과 같다. 그림 6에서 a)는 클라이언트가 컴퓨팅 기능, 서버가 스위칭 기능을 하는데 이 때 "MakeCall Service"와 같은 스위칭 기능의 서비스가 정의된다. 그림 6에서 b)는 클라이언트가 스위칭 기능, 서버가 컴퓨팅 기능을 하는데 이 때 "Route Request Service"와 같은 컴퓨팅 기능의 서비스가 정의된다.

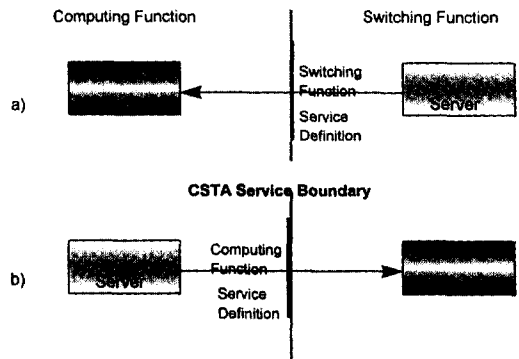


그림 6. 양방향 서비스 정의

3.1.3 CSTA 동작 모델

CSTA 동작 모델(Operational Model)은 서비스 경계를 기준으로 스위칭 영역과 컴퓨팅 영역으로 구분된다. 또한, 스위칭 영역과 컴퓨팅 영역은 각각 스위칭 기능과 컴퓨팅 기능을 CSTA 애플리케이션에 제공한다.

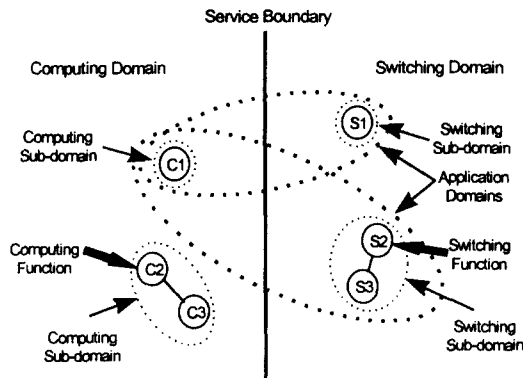


그림 7. 스위칭 영역과 컴퓨팅 영역

CSTA 애플리케이션은 적어도 하나의 스위칭 부영역과 하나의 컴퓨팅 부영역을 포함한다. 그림 7은 스위칭 영역과 컴퓨팅 영역 그리고 각각의 부영역을 나타내고 있다.

일반적으로 스위칭 영역은 컴퓨팅 영역에서 접근되는 전화 자원을 의미하며 국설 교환기, 사설 교환기, 각종 전화기, 키폰 시스템 등에서 구현된다. 컴퓨팅 영역은 스위칭 영역에서 접근되는 컴퓨터 자원을 의미하며, 하나 이상의 소프트웨어 구성요소를 가지는 프로그램을 의미한다.

스위칭 기능에 대한 추상적인 관점을 제공하기 위하여 Device, Call, Connection 등에 대한 부영역 모델을 정의하는데 이들에 대한 상호 관계는 그림 8과 같다.



그림 8. Device, Call, Connection

- Device : 전화 기능을 수행하는 다양한 형태의 물리적, 논리적인 종단점

- Call : Device간의 통신관계를 나타내는 추상적인 개념

- Connection : 스위칭 영역에서 주어진 Call과 Device의 관계를 나타내는 추상적인 개념

(1) Device

Device는 텔리포니 서비스에 접근할 수 있는 추상적인 형태로 스위칭 영역의 다양한 텔리포니 종단점을 의미한다. CSTA에서 정의하는 Device의 범위는 단일 종단점에서 종단점 집합인 그룹까지 다양하며 각각의 Device는 자신의 속성(Attributes), 특징(Features), 서비스(Services)를 갖는다.

Device는 크게 물리적인 Device와 논리적인 Device로 분류된다. 물리적인 Device는 물리적인 성분의 속성과 사용자 인터페이스를 구성하는 특징 및 서비스를 갖는데 예를 들면 전화 단말의 여러 구성 요소들 즉, 음성 수신 장치, 혹은 스위치, 버튼, 램프, 링어(Ringer), 디스플레이 등에 해당된다. 이러한 물리적인 Device는 스위칭 영역에 의해서 Device 식별자를 사용하여 모니터링되고 제어된다. 논리적인 Device는 Call과 상호작용을 하거나 Call을 관리하는데 사용되는 부분을 일컫는다. 예를 들면 매체 스트림 채널이나 신호 처리 기능 등이 이에 해당하고 물리적 요소와도 상호 작용하게 된다.

CSTA Devices에 관련된 주요한 속성으로는 Device 타입, Device 클래스, 정적 또는 동적(Static or Dynamic) Device 식별자, Device 상태 등이 있다. Device 타입은 그 목적에 따라 다양한 형태를 가질 수 있는데 전화 단말, 망 인터페이스 장치(예를 들면, Trunk, CO Line 등), ACD(Automatic Call Distributor), ACD Group, Hunt Group, Park, Park Group 등이 해당된다. Device 클래스는 데이터, 이미지, 음성 등의 다양한 형태이며, 정적 또는 동적 Device 식별자는 Call에서 유일한 값으로 스위칭 영역 내에서 고정적인 값을 갖는다. 그리고 Device는 자체의 상태를 가지고 있지 않지만 Device와 관련된 성분, 요소 및 관련된 Call은 상태 정보를 갖는다. 이러한 상태들에는 연결 상태(Connection State), 물리적, 논리적 성분의 상태 등이 있다.

(2) Call

Call은 여러 Device들의 통신 관계를 나타내는 추상적인 개념이며 Call의 동작(예를 들어, Call 설정, 해제)은 서비스 경계를 통해 여러 Device에서 모니터링되거나 제어될 수 있다. 진행 단계에 있는 Call은 완전하게 형성된 것이 아니므로 Call이 속한 Device는 다른 Device로 대체되거나 두 개의 Call이 하나의 Call로 될 수 있다.

CSTA Call의 속성으로는 Call 식별자, Call 상태가 있다. Call 식별자는 서비스 경계에서 처음 관측 될 때 스위칭 기능에 의해서 각 Call에 할당된다. Call 상태는 하나의 Call을 형성하는 일련의 Connection 상태들로 구성된다.

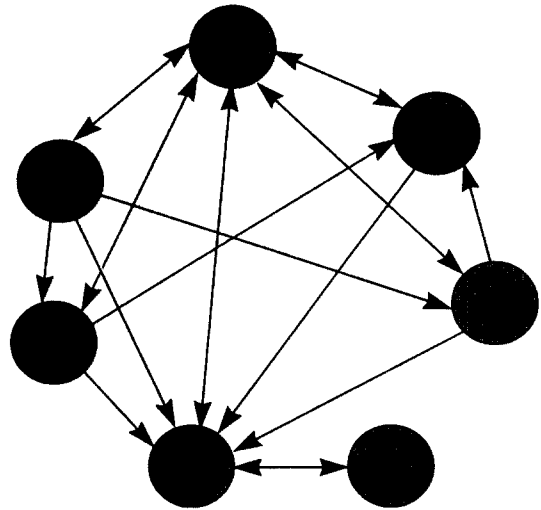


그림 9. Connection 상태 모델

(3) Connection

Connection은 스위칭 영역 내에서 주어진 Call 과 Device의 관계를 나타내는 추상적인 개념을 말한다. Call과 Device의 관계는 컴퓨팅 영역에서 모니터링되고 제어될 수 있다.

Connection의 속성으로는 Connection 식별자, Connection 상태가 있다. Connection 식별자는 스위칭 영역 내에서 Call ID와 Device ID의 쌍으로 이루어지고 하나의 Call과 관련된 Device들에 따라서 여러 개의 Connection ID를 가질 수 있다. 이 식별자는 하나의 영역 내에서는 유일하며 새로운 Call이 발생하거나 또는 이미 존재하는 Call에 새로운 Device가 포함될 때 할당된다. 또한 동작(예를 들면 회의 또는 전송과 같은 동작)에 따라 진행 중에 ID가 변할 수도 있는데 이때는 컴퓨팅 영역으로 보고되어야 한다.

Connection 상태는 하나의 Call 과 Device간 연결 상태를 나타내는 속성으로 단계가 변화될 때마다 다른 값을 갖는다. CSTA에서 정의한 Connection 상태는 다음과 같은 7가지 상태로 기술되고 이러한 Connection 상태의 변화는 모니터에 의해서 보고된다. 그림 9는 Connection 상태 모델이다.

- Null : Call 과 Device 사이에 어떠한 관계도 없는 상태.
- Initiated : Device가 서비스를 요구하는 상태로 이 때 Call을 생성한다. "Dialing" 상

태라고도 한다.

- Alerting : Device의 링이 울리고 있는 상태
- Connected : Device가 Call에 참여하여 다른 Device와 통신하는 상태이며, 논리적인 Call의 참여뿐만 아니라 물리적인 Call 참여도 포함한다.
- Held : Connected 상태의 Device가 비활성화된 상태. 논리적으로는 Call에 참여하고 있지만 물리적으로는 Call에 참여하고 있지 않은 상태.
- Queued : 정상적인 상태의 Call이 그 다음 상태로 천이하기 위하여 대기하고 있는 상태.
- Failed : 정상적 상태의 진행이 중지된 상태로 Call을 요청한 Device나 Call을 받은 Device에 문제가 발생한 경우 또는 Call 생성이 실패한 경우에 발생한다.

3.1.4 CSTA 서비스

CSTA 서비스는 스위칭 영역과 컴퓨팅 영역 사이의 서비스 경계를 정의한 것인데 스위칭 기능 서비스와 컴퓨팅 기능 서비스로 분류할 수 있다. 스위칭 기능 서비스는 스위칭 영역내의 모든 호의 제어, 관리 등을 다루는 것으로서 예를 들면 Make Call, Hold Call,

Retrieve Call, Alternate Call, Answer Call, Call Completion, Clear Call, Clear Connection, Conference Call, Consultation Call, Divert Call, Make Predictive Call, Query Device, Reconnect Call, Set Feature, Transfer Call 서비스 등이 있다.

컴퓨팅 기능 서비스에는 라우팅 서비스가 대표적인 데 예를 들면 Route Request, Route Select, Route Used, Re-Route, Route End 등이 제공된다.

그 밖에도 상태 관리를 위한 서비스로 Status Reporting 서비스가 있다. 스위칭 영역내의 상태 변화를 감지하기 위해 서버에서 클라이언트에게 메시지를 보내는 Event Report 서비스와, 각각의 Call이나 Device에 대해 Event Report 서비스가 이루어지도록 하는 Monitor Start/Stop 서비스가 있다.

CSTA 서비스는 ITU-T 권고안 X.219의 원격 동작 (Remote Operation)으로 모델화되었다. 하나의 개체 (Entity)가 특정 동작의 수행을 요구하게 되면, 다른 개체가 그 동작을 시도하고 응답을 보내게 되는데 결과적으로 CSTA 프로토콜의 동작은 OSI 애플리케이션 계층에서 지원하는 기본적인 요구/응답의 상호작용으로 구성되며 응용 프로세스간의 연관 관계에 따라 기능이 수행된다. CSTA에서는 서비스에 따라 서비스의 요구를 받은 개체가 응답을 생성하는 경우 다음과 같이 세가지 방식이 있다.

- 항상 요구에 대한 응답을 보내주는 비동기 방식 : ITU-T X.219의 Class 2 동작
- 요구에 대한 결과가 실패했을 때만 응답을 보내주는 비동기 방식 : ITU-T X.219의 Class 3 동작
- 요구에 대한 응답을 보내지 않는 비동기 방식 : ITU-T X.219의 Class 5 동작

CSTA 프로토콜은 주로 3rd-Party Call 제어 방식에 적합하도록 고안되었고 이의 구현은 OSI 응용 서비스 개체로서 정의되었다. OSI 응용 계층을 다루기 위한 기법은 일반적으로 OSI의 ACSE(Application Control Service Element)를 사용하고 서비스 요구, 확인 응답 및 이벤트를 전송하기 위한 프로토콜로는 ROSE (Remote Operation Service Element)를 사용한다.

3.2 Versit CTI

Versit는 1994년에 Apple Computer, AT&T, IBM, Siemens 등이 주도하여 다양한 장치 개발 및 자유롭게 장치를 선택할 수 있도록 상호 연동성을 증진시키고자 하는 목적에서 결성되었다. Versit는 그 동안 기존 표준안을 기반으로 좀 더 널리 응용될 수 있도록 플랫폼간 상호 연동성을 보장하는 개방형 사양을 개발하였다.

1996년에 Versit는 CTI 분야에서 관련 용어 정의, 구성 방법, CSTA 기능, 프로토콜, API 등을 포괄적으로 정의하는 Versit CTI Encyclopedia를 발표하였는데

〈표 1〉 Versit와 CSTA 권고안 비교

versit CTI Encyclopedia	ECMA/CSTA Standards & Technical Reports	기타 차이점
Volume 1 : CTI Concepts & Terminology	ECMA Technical Report/52 : CSTA	CSTA : - 3rd Party Call 제어방식 - 스위치와 서버간 연결 고려 Versit : - 1st and 3rd Party Call 제어 방식 - 스위치와 서버간 연결 고려 - 클라이언트/서버, 직접 연결 고려
Volume 2 : CTI Configurations & Landscapes		
Volume 3 : Telephony Feature Set	ECMA-217 : Services for CSTA Phase II	
Volume 4 : Call Flow Scenarios	ECMA Technical Report/68 : Scenarios for CSTA Phase II	
Volume 5 : CTI Protocols	ECMA-218 : Protocol for CSTA Phase II	
Volume 6 : versit TSAPI	API 권고안 없음	

이는 ECMA에서 권고한 CSTA 표준안과 Novell의 TSAIBM의 CallPath 등을 참조하여 만들어진 것이다.

기존의 CTI 링크 표준이 스위치 서버간 단순 연결 또는 클라이언트/서버로 이루어지는 분산 환경 등 특정 구성 환경에 적합하도록 만들어진 반면에 Versit는 이러한 CTI 구성 환경에 구애받지 않고 응용 프로그램이 개발될 수 있도록 만들어졌다.

Call 모델을 포함한 기본 개념은 CSTA를 기반으로 하여 큰 차이가 없으나 그 동안 발전해 온 클라이언트/서버 컴퓨팅 환경과 키폰, ISDN 폰 등 다양해진 전화망 형태에 응용될 수 있도록 기능이 확장되었다. 표 1은 Versit 표준안과 CSTA 표준안을 비교한 것이며 다음은 Versit 내용 중 CSTA에 포함되지 않은 기능들이다.

- 컴퓨팅 영역에서 Capabilities Exchange 서비스를 통해 특정 디바이스에 대한 스위칭 영역 기능 및 특징을 결정할 수 있다.
- 브리징 및 여러 가지 형상을 구성 관리할 수 있는 이벤트 및 모델링 기능을 지원한다.
- 디바이스의 물리적 특징뿐만 아니라 논리적 특징을 지원할 수 있도록 물리적이고 논리적인 서비스/이벤트 등을 포함하여 디바이스를 모델링한다.
- 기존의 디바이스 주소 형식을 지원할 뿐만 아니라 숫자 스트링으로도 다이얼링할 수 있다.
- 클라이언트/서버 환경을 지원하기 위하여 Routing, System Status, Escape 서비스 등을 등록할 수 있다.
- 복수 단계의 다이얼링이 가능하다.
- 서비스 및 이벤트에 사용자 정보를 함께 실어 보낼 수 있다.
- 전화기 돈을 검출하고 송신할 수 있다.
- 호에 미디어 서비스를 지정할 수 있다.

다음은 CSTA의 내용 중 Versit에 포함되지 않은 기능이다.

- 서비스 및 이벤트에 보안 정보를 실어 보낼 수 있다.
- I/O 서비스를 지원한다.
- 호 상태를 알 수 있다.
- 에이전트 상태 모델을 지원한다.

- 라우팅 서비스를 위하여 Route Enable 메커니즘이 제공된다.

IV. SCSA/MVIP

본 장에서는 컴퓨터 기반 트렁크 접속, 음성 처리, 팩스 처리, 음성 인식 등을 위한 모듈간의 상호 연동을 위하여 정의된 버스 방식 표준에 대하여 설명한다. 이 부분에서는 SCSA와 MVIP가 주요 산업표준으로 자리잡고 있다.

음성보드 시장에서 절대적인 우위를 갖고 있는 Dialogic은 1993년 3월 70여개의 다른 CT 회사들과 SCSA(Signal Computing System Architecture)를 발표하였다. SCSA는 개방된 환경에서 CTI를 통한 정보처리를 목표로 소프트웨어, 하드웨어, 그리고, 인터페이스 등을 개방된 플랫폼에서 실현하려는 시스템이다. 특히, 개방된 환경에서 CTI를 이루기 위해서는 미디어 처리, 스위칭, 텔리포니 인터페이스를 위한 텔리포니 서버의 공유와 여러 업체들이 만들어 왔던 기존의 버스 구조와 연동 가능한 텔리포니 플랫폼이 필요하다. 오늘날 약 300개 회사가 SCSA를 지지하고 있다.

SCSA는 개방형 CT 개념을 실현하기 위하여 크게 두 가지를 추구하고 있다. 하나는 소프트웨어에 관계없이 동작할 수 있는 개방형 하드웨어 구조로서 여러 업체에서 만든 컴포넌트가 표준형 PC 또는 VME 컴퓨터에서 상호 연동되어 복잡하고 다양한 특징을 갖는 CT 플랫폼을 가능하게 하는 것이다. 다른 하나는 하드웨어에 관계없이 동작할 수 있는 개방형 소프트웨어 구조로서 미디어 처리, 스위칭, 텔리포니 인터페이스 등을 위해 여러 업체에서 개발한 애플리케이션들이 동일한 CT 서버를 공유할 수 있도록 하는 것이다. 구체적으로 보면 하드웨어 부분에서는 SCbus 버스 방식을 정의하고 있으며 소프트웨어 부분에서는 play, record, voice recognition, text-to-speech 등의 미디어 서비스를 위해 TAO(Telephony Application Objects) 구조를 정의하고 있다. 그러나 소프트웨어 구조 및 인터페이스 사양 정의는 대부분 뒤에서 설명할 ECTF의 기반이 되고 있다.

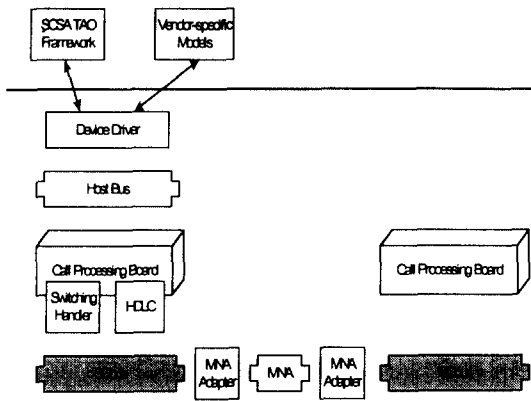


그림 10. SCSA 하드웨어 모델

MVIP(Multi-Vendor Integration Protocol)는 Dialogic과 경쟁관계에 있는 PC기반 음성 처리, 팩스 처리 보드 업체들이 결속하여 만든 GO-MVIP(Global Organization for MVIP)에서 개발한 산업표준으로 CT 분야에서 광범위하게 쓰이는 표준중의 하나다. MVIP의 주요 목적은 서로 다른 업체들이 제공하는 보드 사이의 상호 운용성을 가능하게 하여 하드웨어 컴포넌트의 이식성(Portability)을 높이고자 하는데 있다. MVIP는 트렁크 접속, 음성, 팩스, 음성 합성, 음성 인식 등의 기능 모듈 사이에 상호 연동을 위한 버스 구조이다.

4.1 SCSA 하드웨어 모델

SCbus로 알려진 SCSA 하드웨어 구조는 1993년에 발표되었는데 현재는 ANSI 표준안으로 채택되었다. 최대 2048 타임 슬롯까지 제공할 수 있는 SCbus는 보드와 보드간을 연결하는 리본 케이블 또는 VME I/O 버스로서 구현될 수 있다. 최초의 SCbus 표준안은 기존의 저속 버스와도 호환성을 유지하도록 만들어졌는데 예를 들면 Dialogic의 PEB(PCM 확장 버스)와 MVIP의 1990년 MVIP 90 버스와도 호환이 가능하다. 1995년 ANSI는 SCbus를 ANSI/VITA-6 1994라는 이름으로 VME(VersaModule-Europe) 표준안으로 채택하였는데 그 후 이것이 CT 분야의 유일한 I/O 버스 표준안이 되었다. SCbus의 대역폭을 확장하고 안정성을 높이기 위한 연구가 VITA(VME Industry Trade Association)에서 계속되고 있다.

SCbus는 보드들 사이에 데이터를 송수신하기 위하여 16개의 데이터 라인을 이용하는 동기식 자원 공유 TDM(Time Division Multiplexing) 버스이다. SCbus는 OUT-OF-BAND 신호 방식을 이용한다. SCbus 타임 슬롯의 최대 수는 2048개이며 프로그램으로 그 갯수를 512, 1024로 바꿀 수 있다. SCbus는 단일 분산 TDM 스위칭 모델에 기반을 두며, 대용량 분산 시스템에 적합한 MNA(Multinode Network Architecture) 구조로 되어있으므로 많은 시스템을 상호 접속할 수 있으며 대용량 스위칭 기능을 갖는 시스템 개발에 적합하다. 그림 10은 SCSA의 하드웨어 기본 모델이다.

또한 SCSA 하드웨어 모델에서는 시스템의 확장성을 위하여 여러 개의 노드를 연결할 수 있는 SCxbus를 제안하고 있다. SCxbus는 서로 다른 OS와 하드웨어 플랫폼을 갖는 CT 시스템을 연결하는 확장 버스로서 SCbus로 구성된 노드를 최대 16개까지 연결할 수 있다. 버스 연결을 위해서는 타임슬롯 스위칭과 메시지 전송 기능을 갖는 어댑터가 필요하다. 특히, SCxbus는 MVIP의 노드간 연결 방식인 MC1(Multi-Chassis 1) 버스 방식과 호환 가능하다.

4.2 MVIP

MVIP는 1989년 Natural MicroSystems, Mitel, GammaLink 등에 의해 제안되었다. 당시 Natural MicroSystems에서는 PC기반의 전화 시스템에서 멀티-라인 시스템의 필요성을 느끼고 T1 인터페이스 보드를 개발했으나, 보드 사이를 연결하는 호 라우팅의 문제점을 발견했다. 이와 같은 문제 해결을 위해 GammaLink의 팩스 보드와 연동된 것이 MVIP 표준의 시초가 되었다. 1990년에는 Natural MicroSystems의 DSP를 기본으로 한 음성보드와 Mitel의 T1 인터페이스 보드가 결합되기도 했으며, 이것이 MVIP 90 표준의 기초가 되었다. 1991년 개방형과 많은 업체들의 참여라는 성격을 띠고 표준화 단체로 출범하였다. 1992년에는 140개의 업체가 가입했고 Steering, Educational, Technical, Standards Committee 등 4개의 분과 위원회가 생기게 되었다. 또한 1993년 Steering, Educational Committee에서는 기존의 MVIP와 독립적으로 MVIP 프로토콜을 표준화하여 GO-MVIP라는 단체를 만들었다. 이후 지금까지 200여개 이상의 MVIP 호환 보드 제품들

이 CTI 시장에 선보이고 있다.

MVIP는 초기 표준안 MVIP 90에서부터 H-MVIP (High Capacity MVIP), MC(Multi-Chassis) MVIP에 이르기까지 많은 표준안을 발표하고 있다. 각 표준안을 살펴보면 다음과 같다.

(1) MVIP 90

MVIP 최초의 표준안이며 단일 샷시 시스템에 이용된다. 단일 샷시 시스템에서는 표준형 IBM PC, Micro-Channel-based 컴퓨터, ISO VME bus 표준을 따르는 컴퓨터 등이 있다. 1994년 GO-MVIP에 채택되었으며, 512 x 64 kbps (256 Full Duplex Path) 용량을 갖는다.

(2) H-MVIP

H-MVIP는 MVIP 90의 모든 규격을 포함하고 전송 용량을 512에서 3,072 타임슬롯으로 획기적으로 확장하였다. 따라서 전체 용량은 3,072 X 64kbps (1,536 Full Duplex Path)가 되었다. MVIP 90과 마찬가지로 단일 샷시 시스템이며, 1995년 GO-MVIP에 채택되었다.

(3) MC(Multi-Chassis) MVIP

CPU가 하나 이상 적용되는 시스템의 경우에 해당하며, 멀티-샷시 MVIP의 최초 표준안은 1993년 발표되었고, 초기 제품은 1994년 Amtelco의 MC1 어댑터 카드와 1995년 Natural MicroSystems의 MC1 어댑터이다. MC-MVIP는 MC1(TDM on Twisted - Pair Copper), MC2(FDDI-II on Fiber or Copper), MC3(SDH/SONET), MC4(ATM Technology) 등의 서로 다른 물리적 매체에서 적용되는 소프트웨어 인터페이스로 구분되어 진다.

(4) Connection Control API

1994년 Mitel의 Connection Master 소프트웨어로 업체에 관계없이 적용할 수 있는 드라이버 인터페이스와 소프트웨어 사양인 MVIP Connection Control 표준안이 채택되었다.

V. CT API

앞에서 언급한 바와 같이 컴퓨터와 전화의 결합인 CTI에 대한 관심과 연구가 활발히 진행됨에 따라서 CTI 응용 프로그램인 텔리포니 응용 프로그램의 개발이 중요하게 되었다. 텔리포니 응용 프로그램 개발이 CT 하드웨어 및 CT 시스템에 대하여 독립적으로 이루어지도록 하기 위해서 API의 표준화가 필요하다. 본 장에서는 텔리포니 응용 프로그램을 위한 최근의 API 업계 표준화에 대하여 살펴보도록 한다. 텔리포니 API 분야에서는 3rd-Party Call 제어 방식을 중심으로 하는 TSAPI(Telephony Services Application Programming Interface)와, 1st-Party Call 제어 방식으로 시작되었던 TAPI(Telephony Application Programming Interface) 그리고 최근 JAVA의 등장과 함께 나타난 JTAPI(Java Telephony API) 등에 관심이 집중되고 있다.

5.1 TSAPI

TSAPI는 CTI 텔리포니 서비스를 위한 국제 표준을 기반으로 하고 있는 API이다. TSAPI는 앞서 언급한 ECMA/CSTA의 CTI 링크의 서비스와 프로토콜 정의에 기반을 두고 있다. TSAPI는 이러한 CSTA 표준안을 기반으로 하고 있기 때문에 보편적이고 스위치에 독립적인 API의 특징을 가지고 있다. 또한 Versit는 CTI Encyclopedia를 발표하면서 TSAPI 확장 버전을 포함시켰다.

5.1.1 TSAPI를 이용한 텔리포니 서버 구성

Novell과 AT&T는 TSAPI를 기반으로 Novell 네트워크에 접속된 클라이언트에서 교환기를 제어할 수 있도록 서비스 프로그램(NetWare Telephony Service)과 인터페이스를 발표하였다. Novell사의 NetWare는 텔리포니 서비스와 텔리포니 응용 프로그램사이의 동작과, 클라이언트와 서버간의 통신을 지원하는 O/S이다. 이러한 클라이언트/서버 환경을 텔리포니 서비스에 적용할 경우 클라이언트 플랫폼은 사용자 인터페이스와 사용자 전화 제어 기능을 제공하고, 서버 플랫폼은 호출 라우트하기 위한 호 정보를 제공하거나 또는 발신자 정보를 이용하여 스위치가 호출 사용자 전화에 보내도록

해주는 기능을 포함할 수 있다. Novell의 텔리포니 서비스는 TSAPI를 통해 클라이언트와 서버에서 텔리포니 응용 프로그램을 개발할 수 있도록 해준다. 그림 11은 NetWare 텔리포니 서비스가 구현되기 위한 NetWare서버와 클라이언트의 구성요소를 나타내고 있다. 각 구성 요소를 살펴보면 다음과 같다.

- CTI 링크
 - 이 링크는 서버 내의 특정 하드웨어와 스위치를 연결해준다. 즉, 서버 내의 TSAPI 응용 프로그램과 스위치 내의 호 처리 소프트웨어 사이의 선택을 제공하고 있다.
- CTI 링크 하드웨어
 - 이것은 서버내에 포함되는 것으로서 PBX 스위치 제작 업체 고유의 하드웨어이다.
- 스위치 드라이버
 - 소프트웨어 모듈(NLM)로 CTI 프로토콜과 TSAPI를 매핑해 준다. 이것은 CTI 링크에서 드라이버 인터페이스를 지원해준다.
- 스위치 드라이버 인터페이스
 - 스위치 드라이버와 텔리포니 서비스 모듈 사이에 메시지를 교환하는 소프트웨어 인터페이스이다. 교환되는 메시지 정보는 CSTA 서비스 요구, 응답 메시지, 관리와 유지에 관한 요구, 응답 메시지 등이 있다.
- 텔리포니 서비스 NLM(Network Loadable Module)
 - 이 소프트웨어 모듈은 클라이언트 응용 프로그램과 스위치 드라이버 사이의 통신을 제공하고 있다. 즉, 클라이언트 응용 프로그램과 PBX 스위치 드라이버 사이의 모든 메시지들은 이 모듈을 통해서 전달된다.
- 텔리포니 서버
 - NetWare 텔리포니 서비스 NLM이 실행되고 있는 시스템이다.
- 텔리포니 서비스 API(TSAPI)
 - 텔리포니 응용 프로그램이 텔리포니 서버에 접근하기 위해서 사용되는 C 언어 함수, 데이터 타입, 이벤트 메시지들이다. 이것은 CSTA를 기반으로 하고 있다.
- 텔리포니 서버 라이브러리

서버의 응용 프로그램들은 이 소프트웨어 모듈을 이용하여 TSAPI 함수에 접근한다. 이 라이브러리는 TSAPI 서비스 요구를 받아서 응답을 서버 응용 프로그램에 넘겨준다.

- 텔리포니 클라이언트 라이브러리
 - 클라이언트의 응용 프로그램들은 이 소프트웨어 모듈을 이용하여 TSAPI 함수에 접근한다. 이 라이브러리 역할은 텔리포니 서버 라이브러리와 같다.

5.2 TAPI

TAPI는 Microsoft사와 Intel에서 개발한 텔리포니 API이다. 1993년 별도의 SDK로 발표된 TAPI 1.3 버전은 PBX에서 지원하는 것과 같은 호 제어 응용 프로그램에 초점을 맞추었고 16비트 응용 프로그램만을 처리하였다. TAPI 1.4버전으로 발전되면서 윈도우즈95의 일부분으로 발표되었고 제공되는 기능도 확장되었다. 그러나 두 버전은 데스크탑에서 호를 제어하는 1st-Party Call 제어 방식만을 지원하였다. 윈도우즈 NT 서버 4.0과 윈도우즈 워크스테이션 4.0의 개발과 더불어 TAPI는 TAPI 2.0으로 발전되었고 개발자들을 위하여 TAPI SDK는 MSDN(Microsoft Developer Network)의 일부분인 Win32 SDK 개발자 플랫폼으로 통합되었다. TAPI 2.0은 이전의 두 버전보다 발전된 많은 기능들을 제공하게 되었다. 주요 특징으로는 다음과 같다.

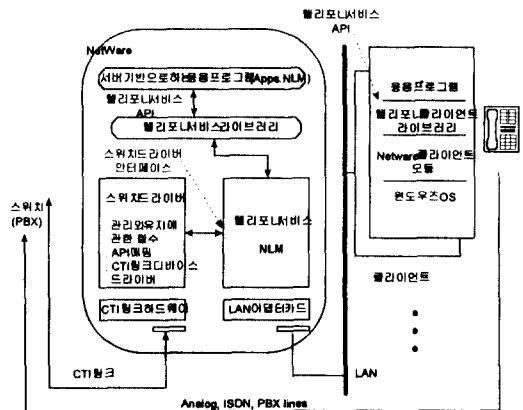


그림 11. TSAPI 기반 텔리포니 서버 구성도

- 전체적으로 32비트 구조를 가지며 16비트, 32비트를 모두 제공한다.
- 응용 프로그램이 QoS 성능 파라미터를 요구, 협상할 수 있도록 해준다.
- 응용 프로그램들이 디바이스를 공유하는 기능이 확장되었다.
- 개발자는 TAPI 2.0을 가지고 3rd-Party Call을 이용한 콜 서버 프로그램을 개발할 수 있다.
- PSTN, ISDN, PBX, IP 등 다양한 네트워크에서 이용될 수 있다.
- TAPI 2.0은 윈도우즈NT 환경에서 하드웨어 추상화와 네트워크에 독립적으로 설계할 수 있기 때문에 개방성과 유연성 있는 CT 플랫폼을 제공할 수 있다.

Windows Telephony SDK는 크게 TAPI와 TSPI로 나뉜다. TAPI는 전화기능을 제공하는 응용프로그램을 개발하기 위한 인터페이스이고 TSPI는 응용 프로그램의 요구를 처리하는 디바이스의 서비스 개발을 위한 인터페이스이다. Windows Telephony에서는 응용 프로그램(TAPI)과 서비스 제공자(TSPI)간의 인터페이스를 위해 TAPI.DLL을 제공한다. 다음 절에서는 최근 TSAPI의 표준 규약 경쟁이 되고 있는 TAPI 2.0의 구성을 살펴보겠다.

5.2.1 TAPI 2.0 구성

그림 12는 유니모뎀(Universal Modem) 드라이버 (UNIMODEM.TSP)를 서비스 제공자로 한 TAPI 2.0의 일반적인 구성이다.

- 16비트 프로세스
 - 윈도우즈3.1 과 윈도우즈95에서 Tapi.dll은 윈도우즈 텔리포니의 중심이 되었다. 즉, 16 비트 TAPI 응용 프로그램들은 Tapi.dll을 통해서 TAPI 서비스를 제공 받았다. 그러나 윈도우즈 NT에서 Tapi.dll은 16 비트 주소로 32 비트 주소로 매핑하여 필요한 요청을 Tapi32.dll에 보내는 역할을 한다.
- 32비트 프로세스
 - 32 비트 TAPI 응용 프로그램은 Tapi32.dll을 통해 TAPI 서비스를 제공 받는다. Windows NT에서 Tapi32.dll은 함수 요청을 Tapisrv.exe에 전

송하고 필요한 서비스 제공자로 하여금 적절한 서비스를 수행하도록 한다.

- 32 비트 TAPI 서비스 프로세스
 - Tapisrv.exe는 TAPI의 중심이 되는 부분으로 윈도우즈3.1과 윈도우즈95에서 Tapiexe.exe와 Tapi.dll에서 수행했던 역할을 포함하고 있고 독립된 서비스 프로세스로 수행한다.

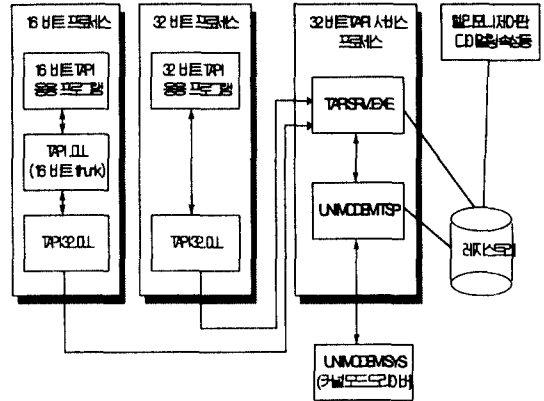


그림 12. TAPI 2.0 구성도

모든 텔리포니 서비스 제공자는 Tapisrv.exe 실행 과정 내에서 수행한다. 이렇게 함으로써 Windows Telephony 버전에서 발생했던 문제점을 제거할 수 있었다. 즉, 이전 버전에서 서비스 제공자는 응용 프로그램 컨텍스트내에서 수행되었기 때문에 이 응용 프로그램의 문제가 발생할 경우 서비스 제공자에 의해서 생성된 자원에 직접적인 영향을 줄 수 있었다. TAPI 2.0 환경에서 서비스 제공자는 작업 수행 중 필요하면 Tapisrv의 컨텍스트내에서 스레드를 생성한다. 그러므로 서비스 제공자에 의해서 생성된 자원들은 응용 프로그램의 영향을 받지 않는다.

서비스 제공자는 CreateFile, DeviceIOControl 을 포함한 시스템 함수나 필요한 다른 컴포넌트를 사용하여 하드웨어 업체에서 설계한 커널 모드 컴포넌트, 서비스, 표준 장치를 이용할 수 있다. 서비스 제공자는 또한 클라이언트/서버를 위해서 RPC, WinSockets, named pipes와 같은 네트워크 서비스에 접근할 수 있다. Windows NT를 위한 Windows Telephony는 KMDDSP라고 하는 커널 모드 TSPI(Telephony Service Pro-

vider Interface) 매퍼(mapper)를 포함하고 있다. 이것은 서비스 제공자가 커널 모드의 디바이스 드라이버로 구현되도록 한다. 윈도우즈 NT에서 서비스 제공자 프로세스는 다른 프로세스로부터 보호되도록 커널 모드 디바이스 드라이버로 구현할 수 있다. 이것은 KMDDSP라고 알려진 커널 모드 서비스 제공자 인터페이스 매퍼를 통해서 가능하다. TAPI의 컴포넌트들은 다양한 프로세스간 통신 메커니즘을 통해 응용 프로그램과 서비스 제공자들 사이의 함수 요청과 메시지를 교환한다.

5.2.2 TAPI 2.0 서비스

텔리포니 응용 프로그램을 개발하기 위해서 윈도우즈 NT와 윈도우즈95에서는 4가지 수준의 서비스를 제공하고 있다.

○ 보조 텔리포니 서비스

텔리포니 기능이 없는 응용 프로그램에 기본적인 텔리포니 기능을 지원하기 위해서 제공하는 서비스이다. 예를 들면, 마이크로소프트 워드 프로세서와 같은 응용 프로그램에 tapiRequest MakeCall과 같은 보조 텔리포니 함수를 첨가하면 간단한 다이얼링을 할 수 있다.

○ 기본 텔리포니 서비스

단순히 호를 설정하거나 수신하는 것 이상의 기능을 제공하는 서비스로 이 서비스의 기능들은 POTS(Plain Old Telephone Service)의 기능들과 일치한다. 그러나 기본적인 텔리포니 함수는 디바이스 드라이버에서 지원해야 한다. 기본적인 텔리포니에서 제공하고 있는 기능들로는 호 설정 및 해제, 호 응답, 호 상태 보고와 이벤트, 호 조작, 그리고 보조 텔리포니 서버의 기능들을 가지고 있다.

○ 추가적인 텔리포니 서비스

텔리포니 응용 프로그램에 보다 강력한 기능을 지원하기 위해서 제공되는 서비스이다. 텔리포니 응용 프로그램을 개발하는 개발자와 서비스 제공자는 이 서비스 내의 기능들을 이용하여 고객의 요구에 맞는 상품을 개발한다. 이 서비스에서 제공하는 기능으로는 호 대기, 호 전송, 호 거절 및 매체 스트림 라우팅에 관한 기능들이 있다.

○ 확장된 텔리포니 서비스

이 서비스는 서비스 제공자 업체들이 장치의 특별한 기능을 확장할 수 있도록 API 확장 메커니즘을 제공하고 있다. 즉, TAPI의 확장 메커니즘을 통해서 서비스 제공자는 TAPI에서 직접 정의되지 않은 함수들을 생성할 수 있다.

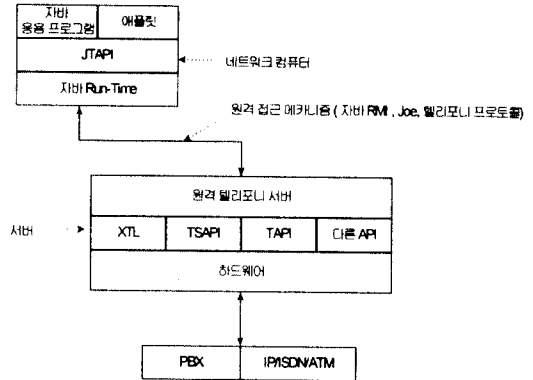


그림 13. 네트워크 컴퓨터 구성

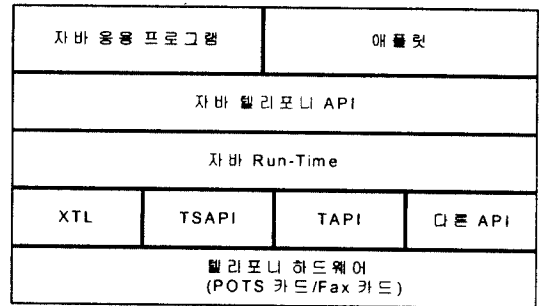


그림 14. 데스크탑 컴퓨터 구성

5.3 JTAPI

1996년 11월 JTAPI(Java Telephony API) 1.0의 시작으로 1997년 2월 1.1로 발전된 JTAPI는 자바를 기반으로 컴퓨터 텔리포니 응용 프로그램을 위한 객체 지향 응용 프로그램 인터페이스가 되었다. JTAPI는 자바 소프트웨어와 함께 활동하는 Sun Microsystems, Lucent, Nortel, Intel, IBM 등의 공동노력으로 이루어졌다.

JTAPI는 1st, 3rd-Party Call 제어 방식 모델을 모두 지원하고 데스크탑 응용 프로그램에서 분산 콜 센터

텔리포니 응용 프로그램까지 다양하게 제공할 뿐만 아니라 웹 환경에서 텔리포니 기술을 이용한 응용 프로그램을 개발하도록 해준다. 또한 JTAPI는 SunXTL, TAPI, TSAPI 등과 같이 기존의 API 상에서 구현될 수 있으므로 컴퓨터 통합기능으로써 이용할 수 있다. 현재 JTAPI는 Portable API Task Group을 관리하는 ECTF의 Call Control Interoperability Working Group에서 C.100으로 표준화되고 있다.

5.3.1 JTAPI 구성

JTAPI의 구성은 네트워크 컴퓨터(NC), 데스크탑 컴퓨터(PC)에 따라서 두가지로 구성된다.

(1) 네트워크 컴퓨터 구성

네트워크용 JTAPI 응용 프로그램이나 자바 애플릿은 원격 워크스테이션에서 실행된다. 원격 워크스테이션은 디스플레이, 키보드, 프로세서와 메모리를 가진 네트워크 컴퓨터이다. 네트워크 컴퓨터는 텔리포니 자원을 관리하는 서버를 이용하여 네트워크 자원에 접근한다. 이 때 JTAPI는 Java's Remote Method Invocation(RMI), JOE 또는 텔리포니 프로토콜과 같은 원격 통신 메카니즘을 통해 서버와 통신한다. 그림 13은 네트워크 컴퓨터로 이루어지는 네트워크 상에서의 텔리포니 서버 구성을 나타내고 있다.

(2) 데스크탑 컴퓨터 구성

이 구성에서 JTAPI와 자바 애플릿은 텔리포니 자원이 존재하는 데스크탑 시스템에서 실행되고 JTAPI가 TSAPI, TAPI 등 이미 존재하는 API위에서 수행된다. 자바 응용 프로그램은 JTAPI를 기반으로 기존의 텔리포니 API에 독립적으로 수행할 수 있다. 그림 14는 데스크탑 환경에서의 텔리포니 구성을 나타내고 있다.

5.3.2 자바 텔리포니 서비스

JTAPI는 자바 언어 패키지로 구성되었고 각 패키지는 컴퓨터 텔리포니 응용 프로그램을 위한 특별한 기능들을 제공한다. 개발자는 이러한 패키지들을 이용하여 텔리포니 서버를 구현 할 수 있다.

JTAPI 패키지는 코어 패키지, 확장 패키지로 구성되어 있다. 코어 패키지는 전화 호 설정, 전화 호 응답, 전

화 호 해제와 같은 기본적인 텔리포니 기능들만을 제공한다. 이러한 기본적인 기능외에 부가적인 기능을 제공하는 패키지가 확장 패키지이다. 확장 패키지들에 대해서 자세히 살펴보면 다음과 같다.

○ 호 제어 패키지(Call Control Package)

java.telephony.callcontrol 패키지로 존재하며 이 패키지는 호 대기 설정 기능, 전송 기능, 회의 기능 등과 같은 발전된 호 제어의 특징들을 제공하고 있다.

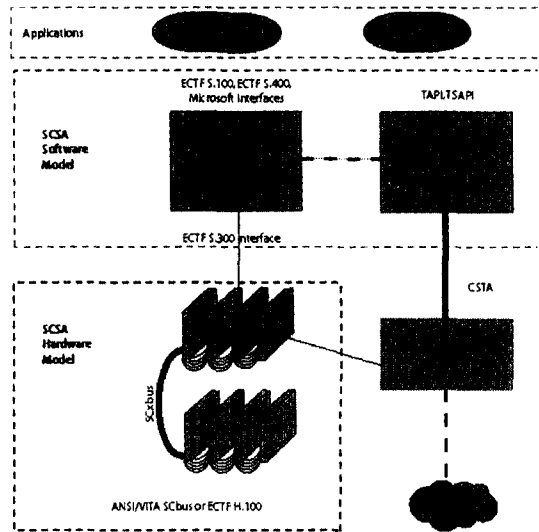


그림 15. CT 구성 요소

○ 콜 센터 패키지(Call Center Package)

java.telephony.callcenter 패키지로 존재하며 라우팅 자동 호 분산(ACD), 예약 전화(Pre-Active Calling)와 같은 대규모 콜 센터를 관리하기 위해서 필요한 특징들을 제공하고 있다.

○ 매체 패키지(Media Package)

java.telephony.media 패키지로 존재하며 응용 프로그램이 전화 호와 관련된 매체 스트림에 접근할 수 있도록 한다.

○ 폰 패키지(Phone Package)

java.telephony.phone 패키지로 존재하며 하드웨어 전화 장치의 물리적인 특징들을 제어한다.

○ 기능 조회 패키지(Capabilities Package)

- java.telephony.capabilities 패키지로 존재하며 이 패키지를 이용하여 어떠한 동작이 수행될 수 있는지 없는지를 조회할 수 있다.
- 프라이빗 데이터 패키지(Private Data Package) java.telephony.privatedata 패키지로 존재하며 이 패키지를 이용하여 응용 프로그램은 하드웨어 스위치와 직접 데이터를 통신할 수 있다. 이 때의 데이터는 주로 스위치에 특별한 동작을 주문하는 데이터이다.

5.4 Tmap

TAPI나 TSAPI는 서로 다른 인터페이스를 요구하므로 소프트웨어 개발자들에게는 무엇을 선택할 것인가 하는 혼돈을 일으키게 할 수도 있다. 그래서 Nortel과 같은 회사에서는 개발자의 혼돈을 없애고 성장하고 있는 CTI시장의 장벽을 없애기 위해 윈도우즈 환경에서 TAPI명령과 TSAPI명령을 변환시키는 Tmap이라는 소프트웨어를 Microsoft, Novell, Intel과 함께 공동 개발하여 이를 무상으로 제공하고 있다. Tmap은 TAPI 응용 프로그램을 Novell TSAPI를 사용하는 클라이언트/서버 환경에서 동작되도록 하는 프로그램이다.

VI. CTI 통합 모델

앞에서 CTI 링크에 관한 표준화 동향과 CT 시스템의 하드웨어 컴포넌트간 데이터 전달을 위한 버스 구조 표준화 동향 그리고 텔리포니 응용 프로그램의 독자적 발전을 촉진하기 위한 API 관련 표준화 동향 등을 살펴 보았다. 이러한 표준안들은 CT 시스템 실현을 위해 상호 보완적 관계를 가지는데 본 장에서는 CT 전체 시스템에 대한 통합 모델을 제시하는 표준화 동향에 대하여 살펴본다.

6.1 SCSA 비전

SCSA는 분산된 CT 서버를 구축하기 위한 포괄적인 구조이다. 앞서 4장에서는 SCSA의 하드웨어 모델에 대하여 살펴보았지만 SCSA에서는 이러한 하드웨어 모델을 바탕으로 상위의 소프트웨어 모델까지 제시하고 있다.

소프트웨어 모델은 1994년 여러 업체들이 6개의 Working Group을 형성하여 SCSA TAO(Telephony Application Object) Framework 버전 1.0을 발표하면서 시작되었다. SCSA TAO Framework 3.0 Call Control Working Group에서는 매체 처리를 위하여 TAPI, TSAPI, Versit와 같이 현재 주목받고 있는 콜 제어 컴포넌트와 상호 연동될 수 있는 SCSA Media Service를 정의하기도 하였다.

하드웨어 모델과 소프트웨어 모델을 통합한 전체적인 SCSA의 CT 시스템 구조는 그림 15와 같다. SCSA TAO Framework 상에서 응용 프로그램 개발자는 ECTF의 S.100, Microsoft의 SAPI(Speech API) 또는 ECTF의 S.400, Microsoft의 ActiveX 등과 같은 객체 지향 인터페이스를 선택하여 사용할 수 있다. 응용 프로그램은 물리적 CT 자원이 있는 동일한 서버에 존재할 수도 있으며 클라이언트 시스템에 분산될 수도 있다. 이러한 SCSA 구조는 ECTF 모델 정의의 기반이 되고 있다.

6.2 ECTF

ECTF(Enterprise Computer Telephony Forum)는 CTI 시장 활성화를 위한 비영리 산업 단체로서 1995년 4월에 Dialogic, Digital Equipment 등을 중심으로 결성되어, CTI 표준안의 구현과 상호운용성에 관해 연구하는 기관이다. ECTF는 표준을 만드는 단체는 아니며 CTI 시장을 위해 여러 분야의 업체가 모인 산업 협의체로서 그 성격이 ATM Forum과 유사하다. 현재 국내에서는 ETRI가 Auditing Member로 참여하고 있다.

6.2.1 ECTF 요구사항

ECTF에서 추구하는 CTI 시스템은 다음과 같은 요구조건을 만족해야 한다. 먼저, 다양한 CTI 애플리케이션을 위한 여러 가지 기능들을 포함해야 되고, 기존의 전화 애플리케이션 및 그 밖의 애플리케이션과 효과적으로 통합될 수 있어야 하며, 클라이언트/서버 환경으로 CTI 애플리케이션 개발이 가능해야 한다. 또한, 플랫폼에 독립적으로 애플리케이션이 제공되어야 하며 전체적인 구조는 모듈성을 갖도록 인터페이스가 정의되어야 한다. 추가적으로 기존의 전기통신 시스템, 기업 정보처리 시스템과 연동될 수 있어야 한다.

그림 16은 ECTF에서 추구하는 CTI 환경으로 구조적으로는 클라이언트/서버 분산 환경이 된다.

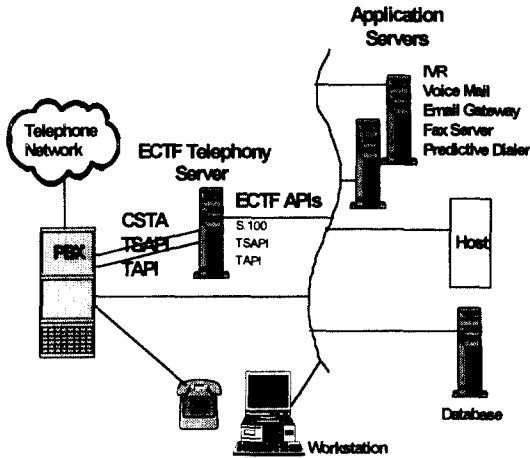


그림 16. ECTF CTI 환경

6.2.2 ECTF 구성

ECTF는 Technical Committee, Marketing Awareness and Education Committee, User Committee 등 3개의 위원회로 구성된다. 그 중 기술 위원회는 CT 기술을 가속화하고 상호 호환 가능한 시스템이 되도록 기술적인 측면에서 산업체가 협의안을 내도록 하는 역할을 하고 있다. 기술 위원회는 WG(Work Group)으로 구성되는데 각 WG은 기술 보고서를 내고 구조를 정의하며 상호 호환성과 관련된 협의안을 도출하고 관련 표준화 단체와 상호 협의의 관계를 갖게 된다. 다음은 각 WG의 역할이다.

- Architectural Working Group
CT의 전체적인 프레임워크를 정의하여 컴퓨터와 전화의 수렴 및 상호 운용성이라는 목적을 가진 ECTF의 모든 WG간 시너지 효과를 낼 수 있도록 한다.
- Call Control Interoperability Working Group
관련 표준안 및 산업 단체와 연계하여 Call 제어 및 상호 운용을 촉진한다. ECMA, IMTC(International Multimedia Teleconferencing Consortium)를 포함한 표준화 단체 및 관련 산업 단체와 협력하여 Call 모델을 정의하는 것이 주요

활동이다.

- Applications Interoperability Working Group
ECTF 텔리포니 서버에서 S.100 호환 애플리케이션에 초점을 두어 CT 애플리케이션의 효과적인 상호 운용을 정의한다. S.100 API에 대해 연구한다.
- Call Center Interoperability Working Group
콜 센터 내에서 데이터의 수집 및 보고에 관한 부분을 표준화하여 개념화하고 단순화시킨다. 콜 센터 환경에서 공통으로 사용하는 용어를 정의하고, 콜 센터 장치로부터 이용 가능한 최소한의 데이터 셋을 정의하고, 클럭 동기화에 대한 요구사항 정의, DB 접근 방법 정의 등을 연구한다.
- CT Services Platform Working Group
확장가능하고 네트워크 구성이 가능한 CT 서비스 플랫폼 구조를 정의한다. S.100, S.200, S.300 등의 권고안을 발표하였다.
- Hardware Components Interoperability Working Group
CT 미디어 트랜스포트를 위한 미디어 버스간의 상호 운용성을 연구하며, MVIP/SCSA 등에 관하여 연구한다. Geoport, USB(Universal Serial Bus), ATM, ISO Ethernet 등과 같은 외부 트랜스포트를 위한 외부 미디어 버스 연결과 내부 미디어 버스간의 상호운용성에 관해 연구한다.

6.2.3 ECTF 프레임워크

ECTF 프레임워크에는 서버와 모듈 두가지 형태의 컴포넌트가 있다. 즉, ECTF 프레임워크의 구조와 기능은 서버와 그에 속한 모듈과 모듈 사이의 인터페이스들로 구성된다. 그림 17에서는 이러한 구조를 보여주고 있다.

ECTF 프레임워크에서 정의하는 서버는 ECTF 애플리케이션 서버와 텔리포니 서버가 있다. 애플리케이션 서버는 분산 환경에서 전화 및 미디어 애플리케이션을 실행시키며 텔리포니 서버는 애플리케이션이 요구하는 전화 및 미디어 자원(line, 음성인식, 팩스 등)을 제공한다. 애플리케이션 프로그램과 텔리포니 자원이 하나의 노드에 존재하는 경우에는 별도의 애플리케이션 서버가 필요하지 않다. 이러한 구성은 텔리포니 서버가 정

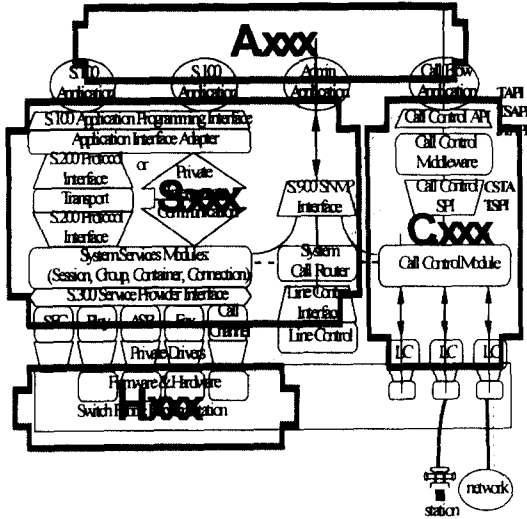


그림 17. ECTF 프레임워크

보 처리 능력을 갖고 있어서 텔리포니 자원 및 애플리케이션 프로그램을 모두 구동하는데 적합한 경우에 해당한다. ECTF 프레임워크에서 정의하는 인터페이스는 다음과 같다.

- Media and Switching Services Interface (S.100)
CTI 응용 소프트웨어 모듈간 상호운용성을 제공하기 위해서 정의된다.
- Transport Protocol Interface(S.200)
ECTF 텔리포니 서버 수준에서의 상호운용성을 제공한다. S.200에서는 TCP/IP, SPX/ IPX, ATM, Ethernet 등 하위 계층과는 독립적으로 동작한다. S.200 프로토콜은 분산환경과 신뢰성 있는 프로토콜을 요구한다.
- Service Provider Interface(S.300)
서로 다른 업체간 텔리포니 자원 및 서비스를 연동하기 위한 인터페이스를 정의한다.
- Hardware Compatibility Interface(H.100)
하드웨어 계층에서의 보드 상호간 연동을 위하여 정의된다. 예를 들면 PCI 보드에서의 TDM 데이터 버스에 대한 인터페이스를 정의한다. 차세대 CTI 하드웨어 플랫폼을 마련하기 위해 H.100에서는 기존의 모든 컴퓨터 텔리포니용 버스 구조와

상호운용이 가능한 CT Bus를 정의하였다.

- Call Control Interface(C.X00)
CSTA, TAPI, TSAPI, JTAPI 등 다양한 텔리포니 API와 연동하기 위하여 정의된다.(참고로 C.X00은 아직 발표되지 않은 권고안임)
- Administrative Services Interface(S.900)
애플리케이션간 상호 교환되는 텔리포니 관리 정보에 대해 정의한다.
- Application Interoperability(A.X.00)
ECTF 텔리포니 서버 수준에서의 상호 호환성을 정의하는 것으로서 구성, 장애, 성능, 보완 과정 등의 관리를 위한 API를 정의한다.

S.100, S.200, S.300 등 S.xxx 계열 권고안에는 텔리포니 서버 및 애플리케이션 서버의 구성형태에 따른 API를 정의하고 있으며, 기본적인 모델은 S.100에서 제공하고 있다. 또한, 프레임워크에서 정의하는 모듈은 다음과 같다. 모듈은 정의된 인터페이스에 맞게 구현된 프로그램 엔티티를 말한다.

- Application Modules
음성 배선, ACD 등 텔리포니 자원을 이용하는 응용 프로그램에 해당한다. S.100 미디어 서비스 프로그래밍 인터페이스를 이용한다.
- Application Interface Adapter Module
S.200의 서비스를 S.100 인터페이스로 맞춰주며, 응용 프로그래밍 개발을 용이하게 해준다.
- System Services Modules
서버의 하드웨어 자원에 대한 운용, 통신, 제어 등의 기능을 제공하는 모듈이다.
- Call Control Modules
기존의 PBX 또는 CO에서 제공하던 전형적인 호 제어 기능을 구현한다.
- Driver Modules
S.300 API와 텔리포니 자원 모듈간의 변환을 위한 로직부분을 구현한다.
- Telephony Resource Modules
ECTF 텔리포니 서버의 모든 하드웨어 자원을 나타낸다.

그림 18은 ECTF 텔리포니 서버 구성 방안을 보여준

다. 스위치와 결합된 텔리포니 서버에서는 S.300 인터페이스를 통하여 호 제어가 이루어지는 구조가 된다.

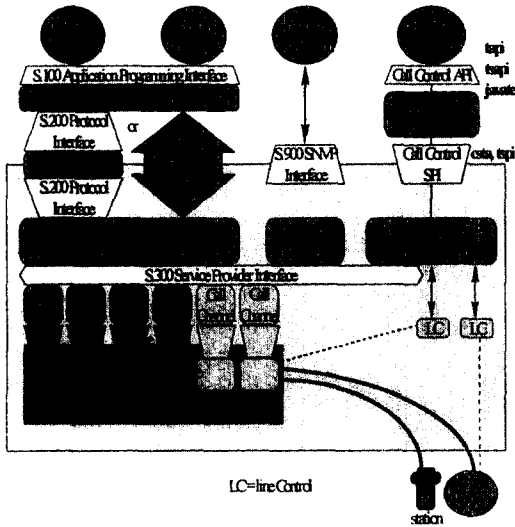


그림 18. ECTF 텔리포니 서버

VII. 결 론

본 고에서는 요즘 신문 및 전문지에 자주 오르내리는 등 관심이 고조되고 있는 CTI 전반에 대하여 표준화 동향을 살펴보았다. CTI 기술에 대해, 간단하게는 컴퓨터와 전화의 결합으로 이해되지만 자세히 살펴보면 전화 시스템과 컴퓨터 시스템, 하드웨어와 소프트웨어 전반에 걸쳐 매우 광범위하게 관련되어 있어 복잡하다. 따라서 본 고에서는 CTI 관련 표준화 동향을 CTI 링크, 버스 구조, 텔리포니 API, CT 통합 모델 등으로 나누어 설명함으로써 CTI 전반에 대한 이해를 돕고자 하였다.

CTI 기술은 ECMA의 CSTA 표준으로부터 시작되고 볼 수 있다. 시대적으로 볼 때 초창기부터 연구가 진행되었고 그 만큼 후발 연구에 많은 영향을 미쳤다. 또한 하드웨어 분야에서는 SCSA의 역할이 크다.

Dialogic 등 관련 업계에서 영향력이 큰 업체를 중심으로 만들어진 사양이 산업 표준으로 자리잡아 가고 있음을 알 수 있다. API 분야에서도 Novell, Microsoft

등 관련 업계에서 산업 표준을 선도하고 있다. CSTA는 Versit, TSAPI 등에 영향을 주었으며 SCSA 구조는 대부분 ECTF에서 수용하고 있다. 그러나 ECTF의 표준화는 아직 초기 단계에 있어 앞으로의 귀추가 주목된다.

표준화 동향 분석을 통해 볼 때 CTI 기술은 전화망 시스템을 컴퓨터에서 수용하기 위한 노력의 결과라고 보아진다. CTI 링크 부분을 제외하면 대부분의 CTI 기술은 컴퓨터 시스템 쪽에 집중되어 있음을 알 수 있다. 이것은 컴퓨터 시스템과 관련하여 하드웨어 및 소프트웨어의 새로운 시장이 펼쳐지고 있음을 예고하는 것이다. 한편, 전화에 대한 사용자 인터페이스는 이제까지 그러했듯이 앞으로도 쉽게 변하지 않을 것으로 보는 관점이 지배적이다.

CTI 기술은 폭 넓은 기술 분야를 다루고 있어 앞으로도 많은 연구가 진행되리라고 본다. 특히, 현재는 단순한 전화와 컴퓨터 시스템의 결합이 아니라 전기 통신망과 컴퓨터 통신망이 만나는 초기 단계로 진단되기 때문에 향후 복합 통신망을 위한 연구가 활발하리라고 본다.

참 고 문 헌

- (1) ECMA-179, Services for Computer Supported Telecommunications Applications (CSTA) Phase I, ECMA, June 1992
- (2) ECMA-180, Protocol for Computer Supported Telecommunications Applications (CSTA) Phase I, ECMA, June 1992
- (3) ECMA TR/68, Scenarios for Computer Supported Telecommunications Applications (CSTA) Phase II, ECMA, December 1994
- (4) ECTF, Enterprise Computer Telephony Forum Architecture Framework Rev. 1.0, 1997
- (5) ECTF, Enterprise Computer Telephony Forum S.100 Revision 1.0 Media Services C Language Application Programming Interfaces, 1996
- (6) ECTF, Enterprise Computer Telephony

- Forum ECTF Call Control Model, 1997
- [7] ECTF, Enterprise Computer Telephony Forum H.100 Revision 1.0 Hardware Compatibility Specifications: CT Bus
 - [8] versit, Computer Telephony Integration (CTI) Encyclopedia Volume 1 : CTI Concepts and Terminology 1.0, 1996
 - [9] versit, Computer Telephony Integration (CTI) Encyclopedia Volume 2 : CTI Configurations and Landscapes 1.0, 1996
 - [10] versit, Computer Telephony Integration (CTI) Encyclopedia Volume 3 : Telephony Feature Set 1.0, 1996
 - [11] versit, Versit CTI-ECMA/CSTA Differences, July, 1996
 - [12] Novell, NetWare Telephony Services Releases 2.21 Telephony Services Application Programming Interface(TSAPI), 1995
 - [13] Microsoft, The Microsoft Windows Telephony Platform : Using TAPI 2.0 and Windows to Create the Next Generation of Computer Telephony Integration, 1996
 - [14] Javasoft, The Java Telephony API, January 1997
 - [15] Edwin Margulies, Client Server Computer Telephony, 1997
 - [16] Carlton Carden, Understanding Computer Telephony, January, 1997
 - [17] Edwin Margulies, The UnPBX, July, 1997
 - [18] Edwin Margulies, The Java API, 1997

정 답

김 속 영

- 현 재 : 삼성전자 정보통신연구소 주임연구원
- 참 조 : 한국통신학회논문지 제21권 3호

- 1995년 2월 : 홍익대학교 전산학과 졸업(학사)
- 1997년 2월 : 홍익대학교 대학원 전산학과 졸업(석사)
- 1997년 2월~현재 : 삼성전자 정보통신연구소 주임연구원

김 대 중

강 우 식

- 1994년 2월 : 고려대학교 전자공학과 졸업(학사)
- 1996년 2월 : 고려대학교 대학원 전자공학과 졸업(석사)
- 1996년 2월~현재 : 삼성전자 정보통신연구소 주임연구원

- 1985년 2월 : 경북대학교 전자공학과 졸업(학사)
- 1996년 7월~현재 : 삼성전자 정보통신연구소 수석연구원