

□ 기술이설 □

차세대 지능망의 서비스소프트웨어 개발기술

한국전자통신연구소 조세형*

● 목 차 ●

- | | |
|--------------------|-------------------|
| 1. 서 론 | 3.2 ITU-T IN의 문제점 |
| 2. 차세대 지능망과 서비스 개발 | 4. 서비스개발환경의 구성 |
| 3. 서비스로직의 정의 및 생성 | 5. 현황 및 향후 방향 |
| 3.1 기본개념 | |

1. 서 론

1984년 미국 AT&T의 분할로 시작된 통신 시장의 경쟁체제 하에서 통신사업자들은 고객이 원하는 새로운 서비스를 찾아 발빠르게 이를 제공해 주는 일에 많은 힘을 기울이게 되었다. 이러한 추세는 현대의 정보통신 어느 분야에서나 공통적으로 볼 수 있는 현상이지만 특히 통신 소프트웨어에서 더욱 중요하게 되었다.

새로운 서비스의 개발에 가장 문제가 되는 것은 특수한 형태의 컴퓨터인 전자교환기이다. 이 때문에 통신사업자들은 서비스에 관한 한 교환기 업체에 대한 의존도가 대단히 높을 수밖에 없다. 뿐만 아니라 일반적으로 하나의 공중망에는 한 종류의 교환기가 있는 것이 아니고 여러 종류의 교환기가 있기 때문에 특히 광역 서비스를 제공하기란 대단히 어려운 실정이다.

기술적으로는 공통적 신호방식의 발전으로 호설정, 해제 등에 필요한 제어정보를 별도의 경로를 통해서 주로 받을 수 있게 됨으로써 항시 호를 제어할 수 있게 되었고, 또한 융통성 있게 많은 양의 정보를 제한없이 고속으로 전달할 수 있게 됨으로써 전달망과 신호망을 분리하게 되었고 이것이 결국은 지능층까지 분리하

여 지능망이라는 개념이 탄생하게 하였다[2].

그림 1은 지능망의 물리적 구조를 나타내 보이고 있다. 지능망에서의 서비스 개발과 비지능망(교환기기반)에서의 서비스 개발을 그림으로 나타내면 그림2-a, 2-b와 같이 나타낼 수 있다. 지능망의 경우는 서비스제어를 범용컴퓨터가 하기때문에 새로운 소프트웨어의 개발이 용이하다. 또한 범용컴퓨터는 용량의 확장등에 있어서도 용이한 잇점을 가지고 있다. 즉, 두 가지의 가장 큰 차이점은 플랫폼의 차이라고 할 수 있다. 그러나 지능망에서의 서비스 개발도 여전히 큰 한계를 가지고 있는데 이것은 바로 개발자들이 통신 네트워크의 세부 사항에 관하여 알아야 할 것이 아직도 너무 많다는 것이다. 즉 일반 컴퓨터 프로그래머가 플랫폼이 제공하는 각종 라이브러리와 API(Application Programming Interface)를 잘 알아야 함은 물론 서비스를 설계하기 위해서는 통신망의 각 구성요소에 대해서도 잘 알아야 하고 특정 프로토콜(Common Channel Signaling No. 7-TCAP 등)에 대해서도 많은 지식을 요구하는 것이 사실이다. 통신사업자들이 서비스 개발에 관해서 바라는 것은 궁극적으로 Operator Programmability, third-party Programmability 그리고 Customer Programmability(통신망 운용자, 제 3의 사업자, 그리고 가입자가 각기 서비

*비 회 원

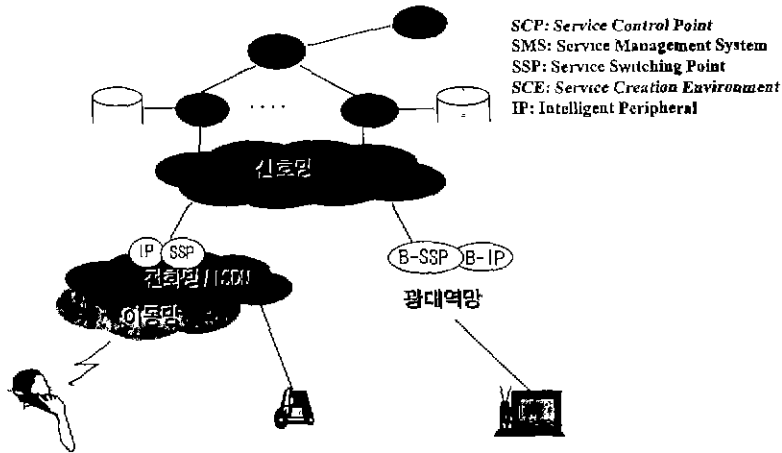


그림 1 지능망의 물리적 구조

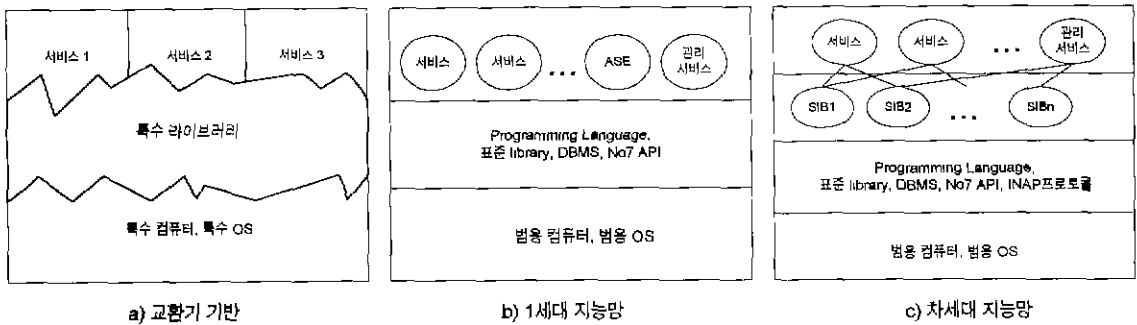


그림 2 서비스 개발 플랫폼

스를 직접 설계, 개발할 수 있게 하는 것) 이 세 가지로 집약될 수 있다. 차세대 지능망은 이를 위한 초석을 제공하고 있다. 이어서 차세대 지능망의 개념을 알아보고 이것이 서비스 개발과 어떻게 관계가 있는지를 예를 통해서 알아보도록 한다.

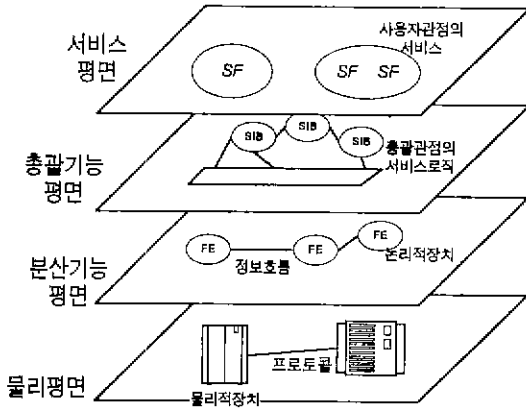
2. 차세대 지능망과 서비스 개발

차세대 지능망의 가장 큰 특징은 표준화와 재사용이라고 할 수 있다. 이 중에서도 서비스 개발과 직접 관련된 것은 재사용의 개념이다. 먼저 재사용을 가능하게 한 배경을 알아보자. 국제표준화 그룹인 ITU-T(이전 CCITT)의 지능망을 보면 소위 4평면 모델을[1] 이용하여 지능망 개념모델을 규격화해 나간다. 우선 대상

이될 기본 서비스 집단을 정의한다(서비스 평면). 다음에는 이 서비스들을 구현하기 위해서 필요한 총괄적인(즉 망을 하나의 점 개체처럼 단순화한) 망기능들을 나열한다(총괄기능평면). 다음은 분산된 기능개체들을 중심으로 상호 정보를 교환하여 서비스를 제공하는 모양으로 모델링한다(분산기능평면). 마지막으로 실물평면은 이러한 논리적 기능들을 물리적인 장치들에서의 처리와 이 장치들간에 주고받는 프로토콜로 모델링을 하며 이러한 과정은 반복적으로 개선해 나가며 진화해 간다. 그림 3은 ITU-T의 4평면 구조를 나타낸 것이다. 그림에 보인 바와 같이 총괄기능평면에서는 SIB이라고 하는 재사용 가능 블록들을 이용하여 GSL(Global Service Logic)이라는 형태로 서비스를 기술한다. 분산기능평면에는 분산기능체(Distributed Fun-

표 1 각 평면의 특징

| | 주관실자 | 구성요소 | 재사용단위 |
|--------|-----------|--------------|---------|
| 서비스평면 | 망은용자, 사용자 | 서비스, feature | feature |
| 총괄기능평면 | 서비스설계자 | SIB, GSL | SIB |
| 분산기능평면 | 서비스/망 설계자 | 분산기능체, 정보흐름 | 정보흐름 |
| 물리평면 | 구현자 | 물리장치, 프로토콜 | 프로토콜 |



지능망의 4 평면

그림 3 지능망의 4평면 구조개념

ctional Entities)가 있으며 이들간에는 서비스를 구현하기 위하여 주고받는 정보흐름이 정의된다. 실물평면은 분산기능평면에서의 하나 또는 그 이상의 기능 단위가 모인 물리실체와 정보흐름을 실현해 주게 되는 프로토콜이 정의되며 실제 서비스는 이러한 장치와 프로토콜에 의해서 수행되게 된다. 표 1은 각 평면의 특징을 요약한 것이다. 다음에는 총괄기능평면과 분산기능 평면을 위주로 어떠한 방식으로 재사용이 일어나고 어떻게 서비스의 설계가 서비스로직으로 변화될 수 있는가에 대한 기본 개념을 설명하고자 한다.

3. 서비스로직의 정의 및 생성

3.1 기본 개념

총괄기능평면에서의 기본적인 재사용 단위는 SIB(Service-independent Bulding Block)이다. 하나의 SIB는 논리적으로 독립된 기능의

블럭이며 특정 서비스에만 종속되지 않고 여러 종류의 서비스에 보편적으로 사용되는 기능을 분산개념을 무시하고 정의한 단위이다. ITU-T에서 정의한 SIB들을 보면 과금, 차단, 기본호 등 14개가 현재 정의되어 있으며 계속해서 다른 SIB들이 정의되고 있다[3][4]. 이 SIB들은 순차적으로 연결되어 GSL(Gloval Service Logic)을 형성하는데 하나의 GSL은 하나의 서비스 또는 feature를 나타내게 된다. 그림 4에 있는 예는 “Automatic Alternative Billing” 이라고 하는, 일종의 신용통화 서비스의 단순화된 형태를 나타낸 GSL이다.

아래의 큰 사각형은 ‘기본호처리’ SIB로서 교환기(즉 호제어 및 서비스 교환기능)에서 수행되는 기본적인 호처리 과정을 나타내는 부분이다. 나머지 SIB들에 대한 내용은 다음과 같이 요약할 수 있다. BCP에서 나오는 점은 시작점(POI: Point of Initiation)이라고 부르며 이 시작점은 곧 교환기의 기본호 처리과정 중 지능망 호처리의 요구가 발생하여 지능망 서비스를 위한 특수한 처리를 시작하는 점으로서 이점의 종류에 따라서 기본호 처리중 어느 시점에서 지능망 서비스 트리거가 발생되는가가 결정된다. 예를 들어 그림 4에 있는 “Address Analyzed”라는 POI는 사용자가 다이얼한 디지털이 수집되어 그의 주소특성이 분석된 시점에서 지능망서비스를 위한 활동이 시작됨을 나타낸다. 이 경우 의도된 서비스의 모양은 다음과 같이 묘사할 수 있다. 기본호의 처리도중 사용자가 다이얼한 디지털의 분석이 끝났을 때 디지털의 앞 부분이 AAB서비스의 서비스코드이면 지능망 서비스 처리를 시작한다. UI(User Interaction) SIB에서는 과금번호와 비밀번호를 물어보고 정해진 자릿수의 디지털을 받는다. Screen (차단)SIB에서는 이 결과를 이용하여

사용자가 다이얼링한 계좌번호와 비밀번호가 정해진 데이터베이스에 있는가를 보고 있으면 번역 SIB를, 없으면 UI SIB를 수행시킨다. 번역 SIB는 계좌번호를 받아 실제 과금할 전화번호를 번역하고 나서 과금 SIB에게 이를 알려주고 과금을 하게 된다. 이러한 일련의 일이 끝나

면 기본호처리 SIB로 돌아와 호처리를 계속 진행한다. 여기에서 밑줄 친 부분은 그림에서는 나타나지 않는 패러미터들인데 실제로 GSL을 작성할 때는 위와 같은 그림 뿐만 아니라 이러한 패러미터(서비스 지원 데이터라고 함) 들도 명시를 해주어야 한다. 이 과정을 분산평면의

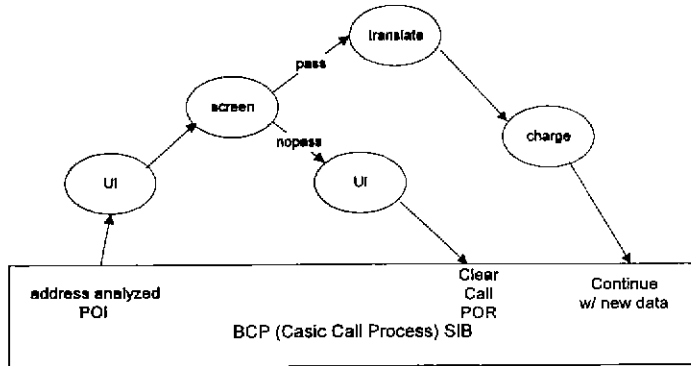


그림 4 신용통화 서비스의 GSL

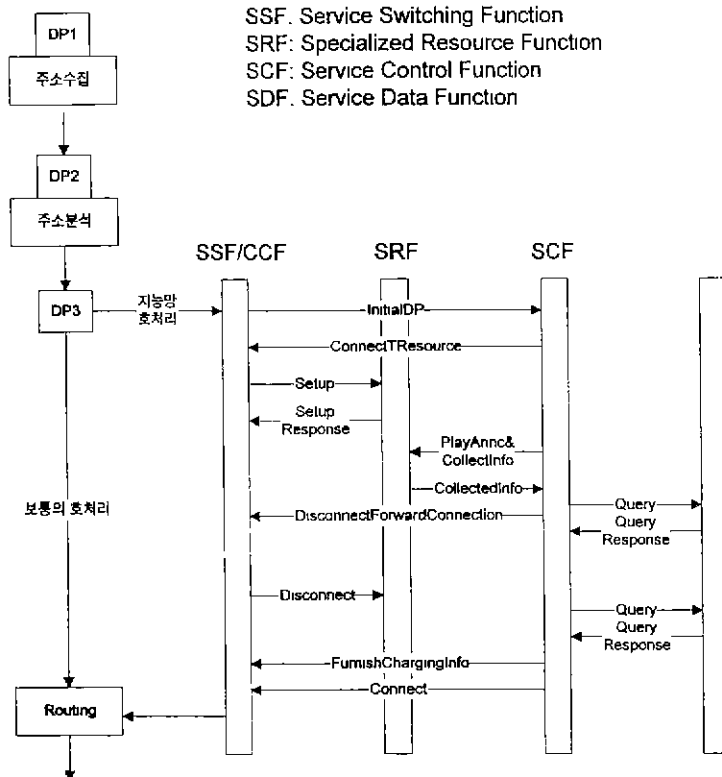


그림 5 AAB 서비스의 분산관점

SSF: Service Switching Function
 SRF: Specialized Resource Function
 SCF: Service Control Function
 SDF: Service Data Function

입장에서 보면 그림 5와 같이 그려낼 수 있다.

여기에서 왼쪽의 작은 사각형들은 기본호처리의 각 단계들을, SSF/CCF, SRF, SCF, SDF는 각 분산기능체들을 나타낸 것이다. 이들간의 화살표들은 각 기능체 간의 정보흐름을 나타내며 수평축은 공간적 변화를 수직축은 시간적 변화들(즉, 시간에 따라 위에서 아래로)나타낸다. 그림 5의 처리과정을 각 SIB 해당부분별로 구분을 하면 그림 6과 같이 된다. 다시 말하면 각 SIB와 그림 6의 각 해당 구별별로의 매핑은 곧 각 SIB를 분산평면에서의 해당 기능으로 번역하는 함수가 된다. 따라서 이 매핑을 이용하여 GSL을 분산관점으로 펼치면 그 서비스를 이루기 위해 분산기능 평면에서 해야하는 일들의 순서가 되는 것이다. 이러한 자동 번역이 가능하다면 적절한 사용자인터페이스와 번역함수를 구현하면 좁은 의미의 서비스 자동개발을

할 수 있다.

3.2 ITU-T IN의 문제점

총괄관점과 분산관점을 분리함으로써 서비스의 규격화 및 개발이 한층 용이하게 된 것이 사실이다. 서비스 설계자는 실제 망에서 제공하는 기능들이 어떻게 분산되지는 이들이 어떠한 방식으로 통신을 하여 하나의 기능을 제공하는지에 대한 세부사항을 할 필요가 없게 된다. 그러나 이를 실현하기 위해서는 GSP와 DFP간의 대응관계가 명확히 정의되어야 하는데 ITU-T의 권고안은 이러한 점에서 아직 많은 문제를 안고 있다[5][6].

그중 한가지는 파라미터의 대응관계이다. DFP의 모든 정보가 GFP에 나타날 필요도 없고 또 다 나타나서도 안되겠지만 최소한 추론을 통해서라도 알 수 있도록 의미가 명확히 표

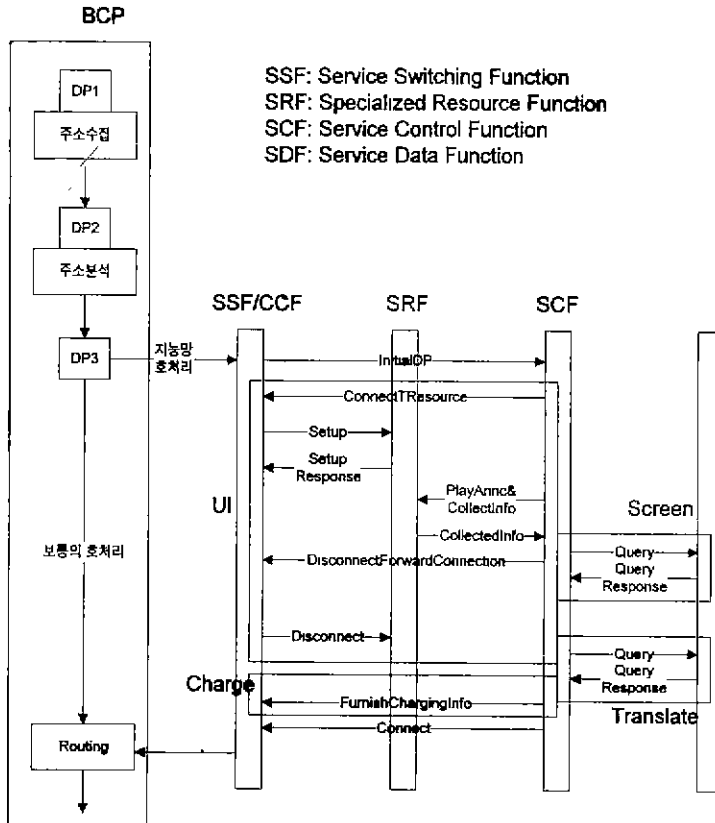


그림 6 AAB 분산관점의 SIB로의 대응관계

현되어야 한다. 예를 들어 end of call POI는 호의 종료시에 서비스로직이 구동됨을 의미하는데, 이 것이 착신자가 끊었을 때와 발신자가 끊었을 때를 구분할 수는 있어야 한다. 그러나 관리차원에서 나중에 채워질 파라미터들은 GFP에서 나타나지 않는 것이 당연하다.

GSL을 서비스생성에 사용하기 위해서는 GSL은 일종의 형식언어가 되어야 하는데 일부 SIB들은 현재 형식성이 일부 결여되어 있다. 특히 데이터를 다루는 translate, service data management 등의 SIB에서 데이터 구조에 대한 형식성이 결여되어 있다.

DFP는 실행평면에 대해서 독립적인 것으로 되어 있으나 실상은 그렇지 않은 점이 있다. 예를 들어 Specialized Resource Function(SRF)의 경우 정보흐름 자체가 물리적 망 형상에 의해 좌우되는 경우가 있다. 이는 평면을 분할한 본래의 취지를 무색하게 하는 것이다.

User interaction SIB의 경우 최적화에 관한 문제가 발생한다. 즉 UI SIB의 규격을 보면 Connect-interact-disconnect의 일반적인 순서만 정의되어 있는데 현실적으로 사용자 상호작용이 여러번 발생할 경우에 중간의 disconnect-connect는 불필요하며 이는 자원의 낭비를 초래한다. 단순한 조합의 경우는 Compiler

이론에서의 peephole optimization을 적용할 수 있으나 다른 SIB 조합과 혼용될 경우 좀더 복잡한 경우의 최적화 방법은 아직 연구된 바가 없다[7].

서비스로직 프로그램의 자동생성에 있어서 가장 걸림돌이 되는 것은 SIB가 그 실행에 있어서 단일성이 없는 것이다. 다시 말해서 하나의 SIB기능을 수행하기위해 실제로 분산된 환경에서 일이 이루어짐에 있어서 SIB의 순서와 분산기능의 순서가 일치하지 않는 것이다. 이것은 총괄 관점이 단순히 분산관점 기능들의 매크로와 같은 형태가 아니라 분산 관점을 무시하고 만들어진 개념이기때문에 일어나는 현상이다. 그림 7의 a는 단일성이 있는 SIB의 예이고 b는 단일성이 없는 예이다.

이러한 문제의 해결은 각 SIB를 정의함에 있어서 단일성을 위배하는 정도를 형식화하여 자동번역이 가능한 형태로 만들 수 있도록 SIB의 규격화시에 단일성 위배도가 어느 범주안에 들게 해야 할 것으로 생각되고 있다[11].

4. 서비스개발환경의 구성

서비스개발환경(SCE : service creation environment)은 넓은 의미로 보면 일종의 CASE

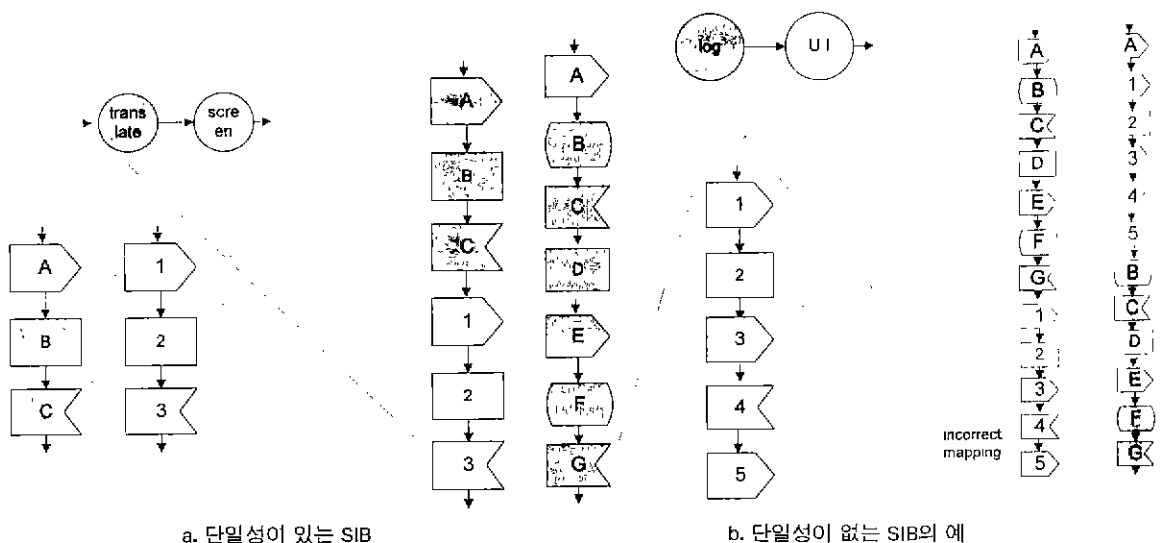


그림 7 SIB와 단일성 관계를 SDL로 스킷치 한 예제

(Computer Aided Software Engineering)도 구라고도 할 수 있다. 다른 점이라면 대상이 통신 서비스 소프트웨어라는 점이다. 앞서 말한 바와 같이 좁은 의미에서의 SCE는 그래픽 인터페이스를 통하여 서비스로직을 설계하고 이로부터 코드를 생성하여 시험할 수 있는 최소한의 환경일 것이다. 이는 툴 위주의 1세대 CASE, 특히 하위 CASE 툴과 유사한 성격을 지닌다. 이 경우에도 역시 대상 소프트웨어가 통신소프트웨어라는 점만이 다르다고 할 수 있다. 그림 8은 이러한 SCE의 구조를 나타낸다.

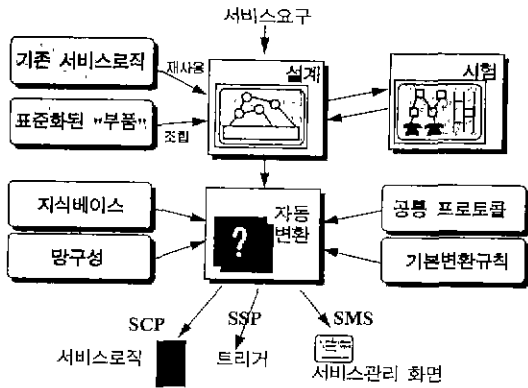


그림 8 기본적인 SCE의 구성

넓은 의미에서의 SCE는 마치 요즈음의 통합 CASE(I-CASE)와 유사하다. 이 경우 SCE는 단지 코드의 생성 및 시험만을 도와주는 것이 아니라 서비스 수명주기에서의 여러단계를 지원해 주어야 한다. 여기에는 서비스에 대한 분석으로부터 시작하여 서비스의 초기배치 및 제공에 대한 지원이 가능해야 하며 서비스 및 가입자 관리를 위한 인터페이스도 제공해주어야 한다. 이러한 SCE는 일반적인 CASE와 약간의 차이를 가지고 있다. 첫째로는 CASE의 사용자층이 컴퓨터 프로그래머인 반면 SCE의 사용자층은 비전문가이다. 둘째, 높은 재사용도이다. 이로 인하여 SCE는 일면 CASE라기 보다는 자동 프로그래밍에 가까운 패러다임을 제공한다. 반면 CASE는 그 범용성이 높는데 비해 SCE는 특정 응용범주에 종속되어 있다(Application-specific). 셋째, 표준화에 의해 망 기능

을 유한 갯수로 제한한 것이다. 이는 바로 재사용을 가능하게 하는 주 인자이다. 또한 SCE는 목적시스템과 분리되어 있지 않고 전체 망 요소장치의 일부로서 존재하며 소프트웨어의 생산 뿐 아니라 통신망에 이 서비스를 배치하고 또 각 가입자에 관련된 여러가지 활동(provisioning & service management)에까지 관여한다는 특징이 있다.

5. 현황 및 향후 방향

SCE의 개념이 처음 제시된 것은 80년대 말경이나 실제로는 최근에 와서야 초보적인 단계의 SCE들이 나타나기 시작했다. 현재 HP, MPR 텔레테크, 필립스 등에서 협의의 SCE를 선보인 바 있고 몇몇 통신사업자 및 컴퓨터 업체들이 프로토타입 수준의 SCE를 개발하고 있으나 현재의 수준은 그래픽 인터페이스를 이용한 초보적인 매크로 프로그래밍의 수준으로 볼 수 있다. 뿐만 아니라 현재로서는 사업자가 원하는 즉 서비스 수명주기를 지원하는 넓은 의미의 SCE는 개념제시 만이 되고 있는 실정이며 실제로 이에 필요한 고도의 기술, 즉 서비스로직 검증이나 상호작용 해결에 관한 연구는 아직 실용화 단계에는 못 미치는 수준에 머물러 있다. 사용 가능 대상에 있어서도 AT&T, 벨 아틀랜틱 등 선진 통신사업자들도 아직 비전문가가 사용할 수 있는 SCE는 발표하지 못하고 있다. 국내에서는 한국 전자통신연구소에서 협의의 SCE에 대한 초보적인 프로토타입을 제작한 바 있으며[8, 9] 한국과학기술원과 협력하여 서비스 상호작용에 관한 기초 연구를 수행하고 있다[10].

SCE가 본래의 목적(가입자, 사업자, 제3자의 서비스개발 가능성)에 맞게 역할을 하려면 현재의 SIB 단위보다 훨씬 복잡도가 낮은 사용자 인터페이스가 필요하다. 이를 위해서는 비교적 적은 숫자의 프리미티브가 요구되며 이는 재사용이 가능한 것이어야 한다. 둘째는 일반인이 쉽게 알 수 있는 프리미티브의 의미론이 주어져야 한다. 셋째는 사용자 인터페이스 자체가 사용자에게 친근한 것이어야 한다. 그러나 아무리 합리적으로 설계를 하여도 결국 지능이 없

는 인터페이스에서의 일반인의 사용은 한계가 있으며 궁극적으로는 인공지능 기법의 도입이 필수적인 것으로 지적되고 있다. 예를 들면 일상언어에 의한 서비스 규격 기술, 또는 사례기반의 추론(Case-based reasoning)에 의한 서비스로직 재사용 등을 들 수 있다. 지능망에서의 서비스개발환경 개념은 통신망 서비스의 개발에 있어서 중요한 이정표가 되고 있다. 우선 이는 개방, 경쟁환경 하에서 살아남기 위한 서비스 개발순기 최단화의 노력이며 이는 통신서비스의 성격 자체를 다량 소품종에서부터 소량 다품종으로 바꾸는 기폭제 역할을 하게 될 것이다. 또한 통신사업에 컴퓨터 업제가 좀 더 깊숙히 침투하게 되는 계기가 될 것으로 예상된다. 지능망에서 시작된 이 개념은 향후 좀더 일반화되어 BISDN에서의 멀티미디어 통신서비스 개발, TMN에서의 관리서비스 개발 등으로 계속 확산되어 갈 전망이다.

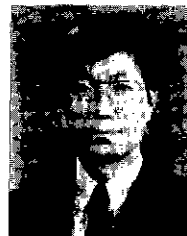
참 고 문 헌

[1] ITU-T Recommendation Q.1201, 1995.
 [2] James J. Garrahan, Peter A. Russo, Kenichi Kitami, and Roberto Kung, Intelligent Network Overview, IEEE Communications magazine, Vol. 31, No. 3, pp. 30-36, March 1993.
 [3] ITU-T Recommendation Q.1213, May 1995.
 [4] ITU-T Recommendation Q.1214, May 1995.
 [5] Ty Chang, Discussing the Weaknesses of the Standard Service Independent Building Blocks, 3rd International Conference on Intelligence in Networks, Bordeaux, France, pp. 67-72, Oct. 11-13, 1994.
 [6] Martin H. Petruk, Experiences in applying the ITU CS-1 Intelligent Network Conceptual Model(INCM) to service Creation, 3rd International Conference on Intelligence in Networks, Bordeaux, France, pp. 134-139, Oct. 11-13, 1994.
 [7] Aho A. et al, Compilers-principles, techniques, and tools, Addison Wesley, 1986
 [8] Sehyeong Cho, A Rapid Prototype of IN Using a Production System, Intelligent

Network 94 Workshop, May, 1994, Heidelberg, Germany.

[9] JungHoon Choi, KyungHyu Lee, Sehyeong Cho, PyeongDong Cho, A Prototype of a Conceptual Model for Intelligent Network, in Proceedings of the 3rd International Conference on Intelligence in Networks, 1994, Bordeaux, France.
 [10] Hyunsoo Kim, WooJin Lee, YongRae Kwon, JungHoon Choi, Sehyeong Cho, Modeling and Detecting Feature Interactions using Petri-Nets, in Proceedings of the 3rd International Conference on Intelligence in Networks, 1994, Bordeaux, France.
 [11] Sehyeong Cho, ChungJae Ihm, JungHoon Choi, ITU-T CS-1 in view of service creation: experiments and evaluations, to appear in the Proceedings of IFIP Intelligent Network Conference, August, 1995, Denmark.

조 세 형



1981 서울대학교 공학사(컴퓨터공학/계산통계학)
 1983 서울대학교 이학사(계산통계학)
 1992 펜실베니아 주립대 박사(전산학)
 1983 주식회사 금성통신 연구소 근무
 1984~현재 전자통신연구소 근무. 현 지능망 서비스연구실장

1994~현재 충남대학교 컴퓨터과학과 겸임교수
 관심분야: 자연어처리, 지식표현, 인공지능 응용, 통신망 SW engineering