

《主 題》

# 지능망 서비스의 정의와 검증

최 정 훈, 권 용 래

(한국전자통신연구원, 한국과학기술원)

■ 차 례 ■

I. 서론	IV. 서비스 개발 환경
II. 지능망 개념모델과 서비스 개발	V. 결론
III. 서비스 생성 과정	

## I. 서론

현대 사회가 점차 고도화되고 다양화됨에 따라 일반적인 단순한 전화 거비스에서 보다 편리하고 다양한 통신 서비스를 제공받으려는 전화 사용자들의 욕구가 증대하고 있다.

새로운 통신 서비스에 대한 욕구는 가입자마다 다양한 형태로 나타날 수 있고 계속적으로 변화하기 때문에 기존 전화망의 교환기가 가입자의 요구를 수용하기는 힘들며 요구 자체가 변화할 때마다 그것을 즉시로 변경시키기는 더욱 어렵다. 이에 따라 기존의 전화망을 이용하면서 새로운 서비스를 쉽게 제공하기 위해 지능망(Intelligent Network)이라는 개념이 출현하였다.

초기 지능망의 기본 개념은 가입자의 다양한 정보를 중앙 집중화된 데이터 베이스 시스템에 수용하여 이러한 정보가 요구되는 호에 대하여 이 시스템에 연결된 특수 교환기가 처리하자는 것이다. 따라서 지능망은 서비스의 변화를 교환기에서의 변경, 즉 하드웨어의 변경이 아니라 집중화된 데이터베이스 및 서비스 제공 소프트웨어 수정으로 수용할 수 있게 되었다.

초기 지능망 구조가 서비스 개발 차원에서 가지는 의미는 종래의 결합되었던 기본 호 처리와 서비스 처리를 분리하여 각각 교환 노드와 서비스 제어 노드에서 처리할 수 있으므로, 서비스 제어 노드에서 기본

호 처리 내부와 접촉하지 않고 서비스 처리를 수행할 수 있고 서비스 관리 시스템에서 수행되는 서비스 처리 프로그램이 비교적 용이하게 만들어질 수 있게 되었다는 점이다.

그러나 이러한 지능망 개념의 도입으로 가입자를 위한 서비스 제공이 용이해지기는 했으나 초기 지능망 구조하에서의 서비스 개발은 다음과 같은 문제점들을 포함하고 있다[1].

첫째, 서비스의 망 의존성이다. 초기 지능망에서 운용하고 있는 서비스들은 특정한 망을 목표로 개발 및 시험한 것들이기 때문에 그 서비스들이 다른 망으로 옮겨질 경우에는 서비스의 일부가 아닌 거의 전부를 수정하거나 새롭게 개발을 시작해야 하는 이식성의 문제가 있다. 또한 새로운 서비스를 제공할 때도 목표로 하고 있는 망의 특성을 개발의 초기 단계에서부터 고려해야하므로 서비스 개발자에게 망에 대한 지식을 요구하고, 개발된 서비스에 대해서도 수정을 어렵게 한다.

둘째, 서비스 개발 기간의 비효율성이다. 일반적인 시스템 개발 체계를 따르는 서비스 개발 방식은 서비스의 정의에서부터 개발이 완료되어 실제 가동될때까지의 기간이 너무 길다는 문제가 있다. 다양한 서비스에 대한 수용의 확산, 잘 개발된 서비스의 편리성 및 수익성을 감안할 때, 개발 기간을 단축시키기 위한 새로운 지능망 개념 및 개발 환경의 지원이 필요하다.

셋째, 서비스의 고정성이다. 지능망의 초기 단계에는 실제 지원되는 서비스의 종류가 그리 많지 않았으나 서비스의 다양화는 지능망의 필연적인 발전 방향이라고 할 수 있다. 따라서 서비스가 완성된 형태로 제공될 경우, 사용자의 요구를 반영할 수 있는 여지가 적고, 수정 및 새로운 서비스로의 조합이 어려운 고정성을 가지게 된다. 즉 사용자의 개별적인 기호에 부응할 수 있는 맞춤형의 서비스 제공과 기존의 서로 다른 서비스 기능들을 조합하여 새로운 서비스를 제공할 수 있는 능력이 필요하다.

따라서 다양한 서비스의 신속한 개발(Development) 및 도입(Deployment) 기능, 서비스의 고객화(Customization), 망 장치에 독립적인 개방형 구조(Open Architecture)등을 만족하는 차세대 지능망이라는 새로운 개념이 등장하게 되었다.

차세대 지능망 구조는 일반적인 데이터베이스를 사용하는 서비스 집합이라는 제한된 관점에서 벗어나 신속한 서비스의 생성 및 도입을 위한 서비스 개발환경(Service Creation Environment)과 개발된 서비스가 수행되도록 지원하는 강력한 플랫폼(Platform)으로 이루어져 있다.

초기 지능망과 차세대 지능망 개념의 중요한 차이점 중 하나는 서비스 개발자의 변화이다. 즉 초기 지능망에서는 망, 통신 서비스, 프로그램 등에 전문적인 지식을 가진 지능망 시스템 전문가가 지능망 서비스를 개발하였으나, 차세대 지능망에서는 이러한 전문 지식이 없는 망 운용자나 고객이 직접 서비스를 개발할 수 있게 한다는 점이다.

지능망 개념의 전환은 새로운 통신 서비스에 대한 일반 고객의 요구가 폭발적으로 증가하는 현상에 기인하는 것으로 해석할 수 있다. R. B. Robrock은 이러한 상황을 자동차의 증가와 비교하여 적절하게 설명하고 있다[2]. 즉 1990년대 초기의 자동차 회사들은 자동차 시장에 대한 수요 예측 연구에서 미국의 자동차 수요는 매년 20만대를 초과하지 않을 것으로 결론을 내렸다.

그러나 예측과 달리 수요는 백만대를 넘어서 예상치의 25배까지 증가했다고 한다. 중요한 것은 이러한 엄청난 차이가 나게 된 원인이다. 이 연구의 자동차 수요 추정 근거는 그 당시 자동차는 전문 기사가 운전하는 것으로 인식되어 전문기사의 증가치에 맞추었으나, 실제로는 일반인들의 자가 운전이 정착됨으로서 수요 예측이 완전히 빗나가게 되었다는 것이다. 이것은 물론 자동차 기술의 발달로 일반인도 수월하

게 운전을 할 수 있는 환경이 제공되었기에 가능해질 수 있다.

이러한 예에서 알수 있듯이 서비스 개발 환경은 복잡하고 다양한 지능망 서비스를 신속하게 생성해야 함은 물론, 전문 지식이 없는 망 운용자나 일반 가입자가 쉽게 서비스를 정의할 수 있어야 한다는 어려운 요구 사항들을 만족해야 하는 차세대 지능망의 핵심 기술이다.

본고에서는 지금까지 설명한 지능망 개념의 진화에 맞추어 서비스 정의와 검증에 관련한 서비스 개발 사항들을 기술하였다. 먼저 지능망 개념 모델하에서 서비스 개발과정을 알아보기 위해 세계적으로 지능망 개발의 표준어 되고 있는 CCITT의 지능망 개념 모델과 이를 바탕으로 한 서비스 개발 방법을 II장에서 기술하였다.

지능망 서비스 개발은 일반적인 소프트웨어 공학의 소프트웨어 순기 모델과 유사하게 서비스의 정의에서부터 검증 단계를 거쳐 지능망에 도입되는 단계를 통하여 이루어진다. 이러한 과정을 좀 더 자세히 살펴보기 위해 III장에서 서비스 생성 과정을 설명하였다.

지능망 서비스 개발은 서비스 생성 과정을 지원하는 자동화된 도구들의 집합인 서비스 개발 환경을 통하여 구체화된다. IV장에서는 서비스 개발 환경에 관하여 좀 더 자세히 정리하고 통신 선진국인 미국과 일본의 지능망 서비스 개발 환경에 관한 사례를 살펴 보았으며, V장에서 결론을 맺었다.

## II. 지능망 개념 모델과 서비스 개발

지능망 서비스 개발은 목표로 하는 지능망의 구조에 따라 서비스의 정의 및 검증 방법이 달라지게 된다. 예를 들어 초기 지능망 구조를 기반을 서비스를 개발한다면 서비스 개발자는 시스템에 대한 전문가가 주체가 되며, 서비스 개발도 통상의 소프트웨어 개발 체계와 유사한 과정을 거치게 되고 개발 기간도 오래 걸리게 된다.

그러나 차세대 지능망을 기반으로 서비스를 개발한다면 망 운용자나 나아가 고객이 직접 신속하게 서비스를 정의할 수 있으나 비전문가가 설계한 서비스이므로 엄격하고 철저한 검증 과정을 거쳐야 한다. 또한 지능망을 제공하는 국가나 서비스 제공자, 그리고 장치들의 특성도 서비스 개발에 영향을 미치게 된다.

따라서 지능망 서비스 개발 방법을 연구하기 위해

서는 바탕이 되는 지능망 구조에 대한 이해가 선행되어야 한다. 이 장에서는 세계적으로 지능망 개발의 표준이 되고 있는 CCITT의 지능망 개념 모델과 이를 바탕으로한 서비스 개발 방법에 관한 권고안을 중심으로 지능망 개념과 이에 따른 서비스 개발 방법[3]을 알아 본다.

2.1 지능망 개념 모델

지능망 개념 모델(INCM: Intelligent Network Conceptual Model)은 지능망의 구조를 설계하기 위한 기본골격(Framework)이다. 지능망 개념모델은 서비스 평면(Service Plane), 총괄 기능 평면(Global Functional Plane), 분산 기능 평면(Distributed Functional Plane), 물리 평면(Physical Plane)으로 이루어진 4개의 평면으로 구성되어 있는데, 각 평면은 지능망 구조를 가진 망에서 제공되는 능력에 대한 서로 다른 이론적 관점을 나타내고 있다(그림 1) 참고).

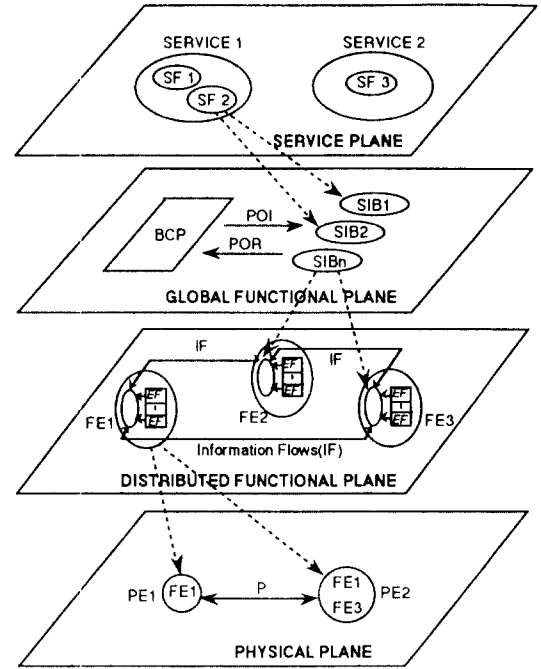
서비스 평면은 서비스 사용자의 입장에서 본 서비스와 관련된 관점에서 망의 행위를 표현하며 서비스의 구현에 관한 정보는 표시하지 않는다. 서비스 평면에서는 "서비스 특징(Service Feature)"이라 불리는 총괄 기능을 이용하여 end-user 또는 가입자 관점에서 지능망이 지원하는 서비스들을 나타낸다. 서비스는 하나 이상의 핵심 서비스 특징으로 분할할 수 있으며 다른 서비스 특징으로 보강할 수 있다.

총괄 기능 평면은 지능망을 하나의 실체(Entity)로 모형화한다. 이 평면은 가입자 관점의 서비스를 SIB(Service Independent Building Block)들을 연결한 체인으로 표현하게 된다. SIB는 여러 서비스에 공통으로 사용되지만 서비스에 독립적인 모놀을 지칭한다. 따라서 총괄 기능 평면에서는 이러한 SIB들의 조합으로 새로운 서비스를 표현할 수 있다.

분산기능 평면은 지능망을 분산개념으로 모형화한다. 각 기능 실체(FE: Functional Entity)는 다양한 기능 실체 동작(FEAs: Functional Entity Actions)을 수행할 수 있다. 각 기능 실체내에서는 다양한 기능 실체 동작들이 하나 또는 그 이상의 요소 기능들(EFs: Elementary Functions)에 의해 수행된다. SIB는 분산기능 평면에서 기능 실체내에 수행되는 특징 기능 실체 동작들의 연결에 의해 실현된다. 이들 기능 실체 동작들중의 일부는 기능 실체사이에 정보흐름을 발생시킨다.

물리 평면은 지능망에 대한 물리적 특성을 모형화한다. 이 평면은 지능망에 존재하는 물리 실체(PE:

Physical Entity)와 프로토콜을 규정한다. 또한 물리 평면은 어느 기능 실체가 어느 물리 실체에 구현되어야 하는가를 나타낸다.



- SIB : Service Independent Building Block
- FE : Functional Entity
- SF : Service Feature
- IF : Information Flow
- POI : Point of Initiation
- ▶ : Pointer
- FEA : Functional Entity Action
- PE : Physical Entity
- EF : Elementary Function
- P : Protocol
- POR : Point of Return
- BCP : Basic Call Process SIB

그림 1. 지능망 개념모델

2.2 서비스의 표현

지능망 서비스는 여러 종류의 서비스 script에 의해 표현될 수 있다. 각 script는 용도나 특성에 따라 서비스 로직 프로그램(SLP: Service Logic Program), 서비스 관리 프로그램(SMP: Service Management Program), 데이터 모형 프로그램(DTP: Data Template Program) 등으로 분류할 수 있다.

서비스 로직 프로그램은 서비스 실행 로직, 통계 로직, 데이터베이스 로직, 과금 로직 등으로 구성되는 실시간 프로그램이다. 서비스 관리 프로그램은 서비스 로직 프로그램에서 생성된 정보로부터 과금 및 서비스 통계 보고서 등을 작성한다. 데이터 모형 프로그

램은 가입자와 상호 작용하며 서비스 및 사용자 데이터를 갱신한다.

한편 본고와 관련한 프로그램은 서비스 로직 프로그램이므로 서비스 로직의 표현에 주된 관심을 두고 설명을 하고자 한다. 지능망 개념 모델의 각 평면은 2.1절에서 설명한 바와 같이 서로 다른 관점을 가지고 있으므로 서비스 로직도 각 평면마다 서로 다르게 표현된다.

먼저 서비스 생성 과정에서 서비스 평면내의 서비스 특징들은 총괄 기능 평면에서 GSL(Global Service Logic)과 기본 호 처리(Basic Call Processing) SIB 그리고 나머지 SIB와 기본호 처리 SIB의 결합으로 표현된다. 분산 기능 평면에서는 총괄 기능 평면의 서비스 로직 집합을 SCF(Service Control Function)에 배치되는 분산 서비스 로직(DSL: Distributed Service Logic) 프로그램의 집합으로 표현한다.

총괄 기능 평면에서 정의되는 SIB는 최소한 분산 기능 평면내의 하나 이상의 기능 실체에서 존재해야 한다. SIB는 한개 이상의 기능 실체에 실재할 수 있으며 따라서 몇가지 기능 실체간의 합동이 필요하다. 분산 기능 평면에서 규정한 기능 실체들은 그들이 대응되는 물리 평면의 물리 실체(Physical Entity)의 행위를 결정한다. 분산 기능 평면에서 정의된 기능 실체간의 관련성은 물리 평면에서 프로토콜로써 규정된다. 분산 서비스 로직이 물리 실체에 실장되면 실제 지능망에 수행가능한 서비스 로직이 실장된다는 의미이다.

지능망 개념 모델을 바탕으로 한 서비스 표현에서 가장 중요한 개념 중 하나는 서비스에 독립적이며 재사용이 가능한 모듈로 존재하는 SIB이다. 즉 서비스는 순차적 순서로 수행되는 사건(event)이나 모듈(module)로 나뉘어지며, 같은 모듈이 많은 다른 서비스에 나타난다는 것이다. 따라서 서비스는 모듈화가 가능하며, SIB가 모듈이 된다. 이러한 SIB들은 결합하여 새로운 서비스를 생성하는데 사용된다. 서비스를 모듈들로 나누는 원리는 CCITT뿐만 아니라 ETSI(European Telecommunications Standards Institute)에서도 채택하였다[4].

### 2.3 서비스 기술 방법론

이 절에서는 앞에서 설명한 지능망 개념 모델을 바탕으로한 서비스를 어떻게 기술한 것인가에 대해 살펴 보도록 하겠다. 지능망을 위한 서비스 기술 방법론(Service Description Methodology)으로는 CCITT 권고

안 중에서 일반적인 통신 서비스 기술을 위한 3단계(Stage) 기술 방법론(I.130)과 지능망 서비스의 생성 과정을 설명한 I.328/Q.1202가 기본적으로 사용된다.

먼저 I.130에 나타난 3단계 방법론을 간략히 정리해보면 다음과 같다.

단계 1: 사용자의 관점에서 서비스를 기술한다. 예를 들어 "사용자는 요구되는 번호를 다이얼한 후 경보음을 받는다"는 형태의 문장으로 기술한다. 이때 이 서비스의 어느 부분들이 망이나 단말기에 의해 지원을 받는지에 관한 사항은 포함하지 않는다는 점이 중요하다.

단계 2: 서비스를 기능 실체들을 중심으로 그들 사이의 기능 실체 동작들과 정보 흐름(Information Flow)들을 기술한다. 이 단계에서는 기능 실체를 물리 실체로 상세하게 할당하는 부분이나, 정보 전달을 지원하는 프로토콜에 관한 내용은 기술하지 않는다. 즉 이 단계는 순수하게 기능 명세를 기술하는 것이다.

단계 3: 서비스를 제공하는데 사용된 프로토콜을 기술한다. 이 단계에서는 모든 메시지를 상세하게 기술하고, 기능 실체를 물리 실체로 할당하며, 특정 메시지를 수신했을 때의 물리 실체의 반응에 관하여 기술해야 한다. 서비스를 구현하기 위해 필요한 모든 사항에 대해서 완벽하게 기술해야 한다.

그러나 CCITT I.130 권고에 근거한 3단계 방법론을 지능망 개념 모델에 적용하기 위해서는 서로의 관련성을 분석하고 확장할 필요가 있다. 지능망의 4개 평면과 3단계 방법론과의 관련성을 분석해보면 다음과 같다.

- 1단계 기술 방법론은 서비스 평면의 서비스와 서비스 특징의 정의 그리고 총괄 기능 평면의 SIB를 정의하는데 이용될 수 있다.
- 2단계 기술 방법론은 분산기능 평면에서 SIB의 실현을 정의하는데 이용될 수 있다.
- 3단계 기술 방법론을 사용하여 정의한 프로토콜은 물리평면에 적용하는데 이용될 수 있다.

한편 이상과 같은 3단계 기술 방법론과 지능망 개념 모델과의 관련성은 바탕으로 CCITT 권고안 I.328/Q.1202에서는 서비스의 특성을 분류하고 서비스 능력을 정의하기 위한 구조적 접근 방법의 필요성을 지적하고 있다.

구조적 접근 서비스를 분석하여 서비스를 SIB들로 분할하는 방법론을 의미한다. 이러한 SIB는 총괄 기능 평면과 분산 기능 평면에서의 모델링을 위한 기본 입력을 형성하는 것으로 4단계의 접근 방법을 정의하

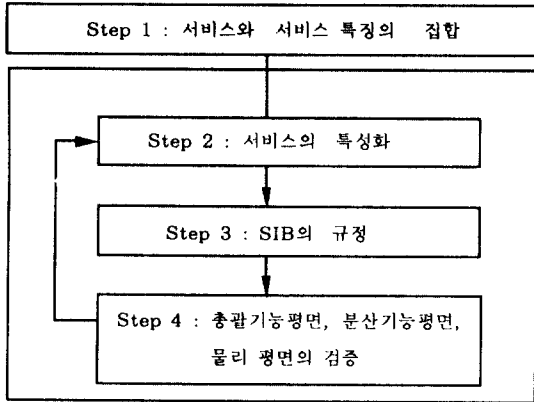


그림 2. SIB의 규정에 대한 방법론

3.1 서비스 생성 순기 모델

서비스의 생성 과정은 크게 서비스의 규격화, 개발, 검증 단계로 이루어지나, 좀 더 자세히 살펴보면 <그림 3>과 같은 7단계로 나눌 수 있다[5].

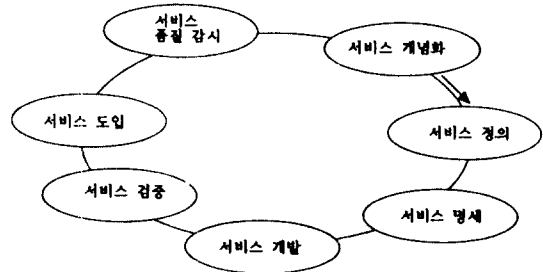


그림 3. 서비스 생성 순기 모델

고 있다.

- Step 1 : 새로운 서비스는 물론 기존 서비스들에 대한 기술도 분석대상이 될 수 있다.
- Step 2 : 서비스를 기능으로 분해하는 원리를 이용하여 서비스를 특성화한다.
- Step 3 : Step 2에서의 분석을 통하여 SIB들로 이루어진 요구 사항이 작성된다. 이 요구 사항은 총괄 기능 평면, 분산 기능 평면, 물리 평면에서의 모델링을 위한 입력으로 사용한다.
- Step 4 : 총괄 기능 평면, 분산 기능 평면, 물리 평면에서의 모델링에 대한 검증을 통하여 SIB를 개선할 경우에는 Step 2로 피드백(feedback)하여 과정을 반복한다.

III. 서비스 생성 과정

일반적으로 서비스 생성 과정은 CCITT에서 정의한 것과 같이 서비스의 규격화, 개발, 검증의 단계로 설명될 수 있다. 이러한 서비스의 생성 과정을 나타내는 서비스의 생성 순기 모델은 서비스 개발 플랫폼의 기본이 되는 중요한 개념이며 이를 중심으로 각종 방법론과 기술이 존재하게 되고 자동화된 도구들이 필요하게 된다.

따라서 본장에서는 먼저 서비스 생성 순기 모델을 중심으로 각 과정을 살펴보고, 서비스 개발자와 서비스의 정의와 검증 과정의 관계에 관해 언급하고자 한다.

서비스 개념화(Service Conceptualization)는 서비스 생성 과정의 첫단계로서 고객과의 상호 의견 교환을 통하여 텍스트나 다이어그램을 이용하여 서비스를 기술(description)한다. 서비스 개념화는 서비스에 관한 개념을 구체화시켜줌으로써 고객이 제안한 서비스가 자신의 기대를 만족하는지를 알 수 있게 한다.

서비스 정의(Service Definition) 단계는 개념적인 정보를 그 개념을 지원하는 기능의 논리적인 정의로 변환하는 과정이다. 이 단계는 요구하는 서비스를 지원하기 위해서 기존에 존재하는 재사용이 가능한 기능과 새로이 구현해야 하는 기능의 분류작업이 이루어지며, 사업적인 측면과 기술적인 측면을 고려하여 타당성에 관한 상세한 분석을 통하여 서비스 생성 비용을 평가한다.

서비스 규격화(Service Specification) 단계에서는 논리적으로 정의된 서비스를 물리적으로 존재하는 프로그램이나 데이터베이스로 매핑시키는 과정이다. 이 단계의 주요한 출력은 서비스 개발과 검증 단계를 수행하기에 충분한 서비스 규격과 상위 레벨 구조 설계를 포함한다.

서비스 개발(Service Development) 단계는 상위 레벨 구조 설계를 상세구조 소프트웨어 설계로 변환하고 설계를 실현하기 위해 요구되는 소프트웨어 성분, 데이터 정의등을 개발하는 단계이다. 이 단계의 주요 출력은 서비스 소프트웨어와 엄격한 서비스 검증 시험을 위한 문서이다.

서비스 검증(Service Verification) 단계는 개발된 서

비스 소프트웨어가 서비스 응용에 대한 규격을 완전히 만족하는지를 시험하기 위한 과정이다. 이 단계의 주요 출력은 검증된 서비스 소프트웨어와 서비스 도입에 필요한 문서이다.

서비스 도입(Service Deployment) 단계는 해당 서비스를 지원하는 모든 소프트웨어와 하드웨어를 포함한 요소들을 망에 설치하는 과정이며, 서비스 감시 단계는 고객에게 제공되고 있는 서비스의 운용 상황을 지속적으로 감시하는 과정이다.

이상과 같은 서비스 생성 모델을 지원하는데 필요한 방법론 및 도구들을 결합한 환경을 구축하기 위해서는 Computer Aided Systems Engineering을 이용한 자동화 기술, 사용자 인터페이스 기술, 객체 지향 기술, 전문가 시스템 등과 같은 소프트웨어 공학 및 인공 지능을 용한 첨단 기술을 도입해야 할 필요성이 강조되고 있으며 연구결과가 발표되고 있다([3],[5],[6],[7]).

### 3.2 서비스 개발자에 따른 서비스의 정의와 검증

지능망의 능력이 고객이 정의하는 서비스를 지원할 수 있는 구조로 발전하면서 여러 수준의 서비스 개발자가 존재하게 되었다. 즉 서비스 제공자와 같이 일반적인 서비스 생성의 전 과정을 책임져야 하는 개발자가 있는 반면, 큰 사업체나 대규모 사용자들은 서비스를 정의 및 개발후 제어할 수 있는 정도를 원하며, 소규모 사업자나 일반 사용자는 원하는 서비스를 정의하고 관련 데이터를 변경하는 정도의 기능을 필요로 한다.

이렇게 지능망이 점차 고객 자신이 서비스를 스스로 생성할 수 있는 맞춤형 서비스(Customized Service)를 제공하는 방향으로 진화함으로써 해결해야 하는 중요한 문제는 첫째 망이나 프로그램에 대한 지식이 없는 고객이 수월하게 서비스를 정의할 수 있는 방법, 둘째 지능망의 신뢰도와 안전을 보장하기 위한 서비스의 검증 방법에 관한 연구이다.

현재 고객이 쉽게 서비스를 정의할 수 있는 방법으로 각국의 차세대 지능망 시스템에서는 아이콘 기반 언어(Icon-based Language)를 설계하여 그래픽 사용자 인터페이스를 통한 시각적 프로그래밍 자동화 도구를 제공하는 것이 일반적 추세이다. Bellcore[8], NTT([9],[10],[11]), Ericsson[4]의 차세대 지능망 시스템들이 이러한 예이다.

한편 서비스 개발자가 정의한 서비스는 실제로 지능망에 도입하기 전에 필수적으로 거쳐야 할 단계가

검증 시험이다. 서비스 검증 시험의 목적은 망의 구성 요소들의 기능들과 주어진 지능망 서비스의 모든 특성들이 요구 사항서에 명세한 바와 동일하게 동작하는지를 확인하기 위한 것이다.

Richlien Hoang은 정의한 지능망 서비스를 검증 방법으로 첫째, 주어진 지능망 서비스 구조 및 고객 환경과 가능한 유사한 시험 망을 사용하도록 하고, 둘째, 자동화된 시험 시스템을 이용하여 체계적으로 모든 특성들을 시험해 보는 것을 제안하고 있다[12].

그러나 일반적으로 지능망 서비스의 검증 시험은 기본적인 교환 서비스에 비해 훨씬 복잡하고 어려운 과정이다. 왜냐하면 분산된 지능망 호 처리, multi-vendor 환경, 다른 신호 프로토콜과의 연동, 많은 서비스 특성 집합 등이 고려되는 상황에서 서비스 검증이 이루어져야 하기 때문이다.

특히 망에 관한 지식이나 프로그램에 대해 친숙하지 않은 고객의 서비스 명세인 경우에도 고객의 단말기를 통하여 서비스 관리 시스템(Service Management System)으로 보내서므로 고객이 정의한 서비스가 수행될때 망에 대한 신뢰도를 보장하기 위해서 명세서는 수행전에 철저한 검증 과정을 거쳐야 한다.

고객이 정의한 서비스에 대한 검증 방법은 아직은 연구 논문에서 발표되고 있는 수준이다. 이러한 검증 방법은 서비스 개발 환경에 관한 논문에서 부분적으로 소개되고 있으며[13], 최근에는 NTT의 Yukari Nagano와 Naoki Uchida가 고객이 정의한 서비스에 서비스 설계자가 검증 로직을 명시해서 서비스 관리 시스템에서 수행하는 정적 검증 방법을 제안한 바 있다[9].

## IV. 서비스 개발 환경

지금까지 기술한 서비스 정의와 검증을 포함한 서비스 개발 과정은 서비스 개발 환경을 통하여 구체화된다. 이 장에서는 서비스 개발 환경에 대한 기본적인 개념을 먼저 설명한 뒤, 서비스 개발 환경에 관한 연구를 활발히 수행하고 있는 미국과 일본의 사례를 살펴보기로 한다.

### 4.1 서비스 개발 환경의 개념

지능망 서비스를 요구 분석, 설계, 코드, 시험 등으로 이어지는 통상적인 소프트웨어 개발 과정에 따라 개발할 경우의 개발 기간은 몇달에서 몇년 정도가 걸린다. 따라서 서비스 개발 기간을 획기적으로 줄일 수

있는 방안은 서비스 개발 환경을 이용하는 것이다.

서비스 개발 환경의 기본 개념은 <그림 4>와 같이 새로운 서비스에 대한 아이디어를 서비스 개발용 워크스테이션을 이용하여 서비스를 표현하고, 검증 시험을 거쳐 지능망에 도입하므로써, 서비스를 신속히 제공할 수 있다는 것이다[14].

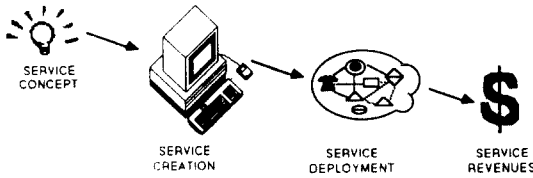


그림 4. 서비스 개발 환경을 이용한 서비스 생성 과정

이와 같은 개념을 지원하기 위해 서비스 개발 환경은 III장에서 설명한 서비스의 개발 과정과 관련한 방법론과 도구들을 통합하여 지원한다. 즉 서비스 개발 환경은 사용자의 요구를 수집하고 정형화한 명세서를 작성하는 작업에서부터, 개발을 마치고 시험을 거쳐 망 위에서 실제 운용되도록 전개하는 작업에 이르기까지의 작업을 서비스 개발환경이 지원해야 한다.

한편 CCITT에서는 서비스 개발 환경에 대한 권고안은 아직 제시하지 못하고 향후 연구 과제로 남겨두고 있는 상황이며, 현재 초기 차세대 지능망 개념 모델인 CS-1(Capability Set-1)에서는 서비스 개발 환경에 대하여 다음과 같이 규정하고 있다.

“서비스 개발 환경은 망 구성 및 망유형과는 독립적으로 SIB를 이용하여 새로운 서비스를 손쉽게 개발하도록 하는 환경을 제공한다. 서비스 개발 환경은 서비스 script를 개발할 수 있는 서비스 개발 과정을 지원하기 위해 서비스 구현과 독립적으로 도구, 기술, 프로그래밍 언어, 절차등을 제공하여야 한다.”

서비스 개발 환경에서 지원해야 할 도구들에 대한 예로서 미국 Bellcore의 보고서에서는 분석 환경, 개발 환경, 생산 환경, 통합 및 시험 환경, 망 운용 환경등으로 나누어 각 분야별로 도구들의 집합을 정의하였다[15].

분석 환경은 서비스와 데이터의 명세를 사용자 레벨, 기능적 레벨, 물리적 레벨에서 생성, 정형화, 검증하는 도구들의 집합이며, 개발 환경은 주어진 타입의 망에 한 프로그램 코드의 설계, 개발, 시험을 돕는 도

구들로 구성된다. 생산 환경은 소스 소프트웨어를 특정한 수행 가능한 이미지와 특정한 망에 대해서 소프트웨어 구성 요소들의 형상으로 변환하는 도구들의 집합이다. 통합 및 시험 환경은 실행 가능한 이미지를 물리적인 망에서 검증하는 도구들의 환경이며 망 운용 환경은 망의 특징 위치에 수행 가능한 이미지를 설치하고, 설치 후의 시험을 돕는 도구들로 구성된다.

#### 4.2 연구 개발 사례

서비스 개발 환경이 차세대 지능망 플랫폼을 기반으로 고객이 원하는 새로운 서비스를 제공한 사례로는 최근 미국의 Bell Atlantic사가 Bellcore와 함께 개발한 SPACE 시스템을 이용하여 공식적인 서비스를 시작한 것이 세계 최초이다. 일본의 경우에는 1990년대 중반 이후에야 차세대 지능망 서비스가 가능할 것으로 보인다.

한편 Ovum의 지능망 기술 조사 보고서에 따르면 많은 서비스 개발 도구들이 발표되고 있으나 대부분 특정 플랫폼에 종속적이며, Digital이 Telesoft와 MPR과 협력하여 개발하고 있는 Unix를 기반으로 한 서비스 개발 환경이 유일하게 하드웨어에 독립적인 것으로 알려지고 있다[16]. 따라서 차세대 지능망 개념에 충실하게 망이나 장치에 완전 독립적인 서비스 개발 환경을 구현하는 것이 현실적으로는 어려움 많은 것으로 보인다.

국내의 경우에는 한국과학기술원에서 1991년부터 서비스 개발 환경에 관한 기초 연구를 수행하고 있으며[1], 최근 한국전자동신연구소에서 총괄 기능평면을 중심으로 서비스 로직을 표현하는 프로토타입을 개발하고 있다.

#### ·미국의 SPACE 시스템

SPACE 시스템은 <그림 5>에서와 같이 Bellcore가 Bell Atlantic 및 IBM과 함께 개발한 서비스 플랫폼이 ISCP(Integrated Service Control Point)의 일부로서 통합 서비스 생성 관리 시스템이다([6],[8]).

특히 서비스의 표현은 서비스 기능 요소를 표현하는 그래픽 아이콘들을 연결하는 순서도(flow-chart) 형태로 구성하는 시각적 프로그래밍을 이용한다.

서비스 개발 과정은 서비스의 생성, 검증, 시험, 활성화 단계로 이루어진다. SPACE 시스템에서의 서비스 개발 과정을 그림으로 나타내면 <그림 6>과 같다[14].

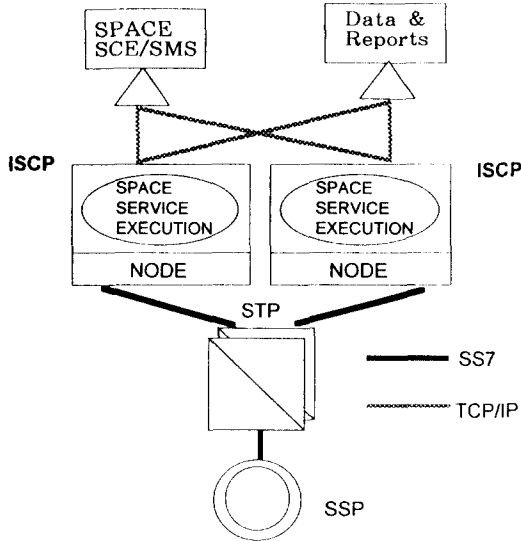


그림 5. ISCP 구조도

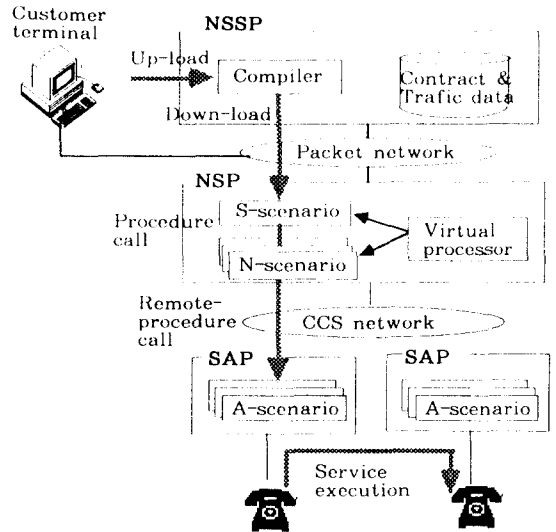


그림 7. 일본의 지능형 플랫폼과 시나리오

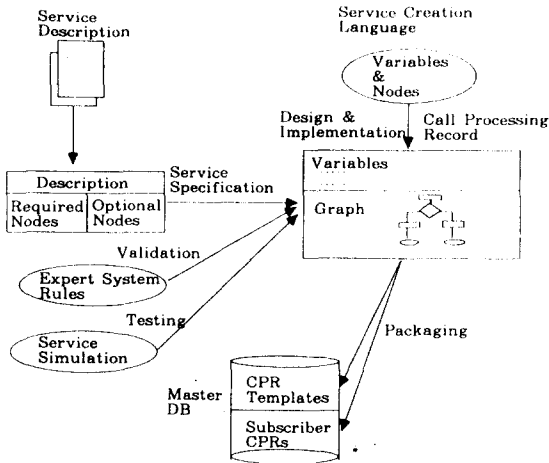


그림 6. SPACE 시스템의 서비스 개발 과정

·일본의 서비스 개발 환경

일본의 NTT는 1992년 현재 free dial, dial Q2 서비스를 제공하고 있으며, 차세대 지능망 전단계의 서비스로 mass calling, VPN 서비스를 제공할 예정이다. NTT는 1990년 중반에 차세대 지능망의 실현을 목표로 <그림 7>과 같은 구조를 플랫폼인 NSP(Network Service Control Point), NSSP(Network Service Support Point) 및 SAP(Service Action Point)를 개발하고 있다 [10].

NSSP는 서비스 생성 및 관리 시스템으로서, NSSP의 컴파일러는 시나리오(서비스 script)를 NSP가 처리할 수 있는 형태로 변환하여 NSP로 다운 로드시킨다. NSP는 서비스 제어에 위한 실시간 서비스 제어기이다. SAP은 SSP(Service Switching Point) 기능과 IP(Intelligent Peripheral) 기능을 제공한다.

NTT의 차세대 지능망의 특징은 물리적인 망과 고객이 운용할 수 있는 논리적인 망사이의 갭을 연결하기 위해 서비스를 정의하는 주체인 시스템 설계자, 서비스 설계자, 고객이 서비스를 개층적으로 정의하는 망 구조를 가지고 있다는 것이다([9],[10],[11]).

즉, 시스템 설계자는 지능망 시스템의 망 요소들에 대해 전문적인 지식을 가지고 있는 계층이며, 서비스 설계자는 지능망 서비스에 대한 논리적인 표현을 할 수 있는 계층이며, 고객은 망이나 프로그램에 전문 지식이 없으므로 자기에게 필요한 서비스의 특징만을 표현할 수 있는 계층이다.

따라서 각 계층별로 설계자는 해당하는 시나리오를 작성하게 되어 있다. 고객은 자신의 요구사항을 정의하는 S-scenario(Service Scenario)를 작성하고, 서비스 설계자는 서비스 요소 프로그램을 정의하는 N-scenario(Network Control Scenario)를 작성하며, 시스템 설계자는 망 제어 요소 프로그램을 정의하는 A-scenario(Action scenario)을 작성한다. 정의한 시나리오는 NSSP에서 NSP의 시나리오 수행 명령어로 컴파일되어 패킷망을 통해서 다운 로드된다.



## V. 결 론

지능망 서비스는 서비스 제공자가 완전한 서비스를 제공하던 방식에서 점차망 운용자나 고객이 서비스를 정의하고 제어하는 방식으로 진화하고 있다. 따라서 다양한 서비스를 신속하게 개발할 수 있는 방안으로, 재사용이 가능하고 서비스에 독립적인 SIB의 조합으로 새로운 서비스를 생성하는 서비스 모듈화의 개념이 도입되었다.

이러한 지능망 서비스 개념의 진화에 따라 서비스의 정의 및 검증을 포함한 서비스 개발 과정에 소프트웨어 공학과 인공 지능을 포함한 각종 첨단 기술을 이용한 서비스 개발 환경이 등장하게 되었다.

그러나 서비스 개발 환경에 관한 연구는 CCITT의 차세대 지능망 개념 모델인 CS-1에서 구체적인 연구가 수행되지 못하고 향후 연구 과제로 미루어진 상황이며, 또한 이를 지원하는 서비스 플랫폼의 개발을 포함하여 해결해야 할 과제가 많이 남아 있다.

차세대 지능망의 성공을 가름하는 관건이 될 서비스 개발 환경의 개발을 위해서는 서비스 정의 및 검증 방법, 서비스 플랫폼과의 인터페이스, 운용 관리 시스템과의 관계 정립 등에 관한 체계적이고 지속적인 연구가 필요하다.

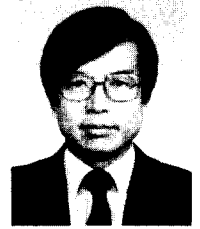
## 참 고 문 헌

1. 한국과학기술원, 서비스 개발 환경에 관한 기초 연구, 위탁과제 최종보고서, 1991.
2. R. B. Robrock, "Putting the Telephone User in the Driver's Seat," International Council for Computer Communication Intelligent Networks Conference, Tampa, May 4-6, 1992.
3. CCITT, Recommendation Q.120x, Q.121x, 1.130, 1992.
4. Mats Wennerberg, Paul van Hal, "SIBs-A Creative way to Design Services," 2nd International Conference on Intelligence in Networks, 1992.
5. Dilip Singh and Donald R. Garrison, "Requirements for Service Creation Platforms in Open Systems Environment," International Council for Computer Communication Intelligent Networks Conference, Tampa, May 4-6, 1992.
6. Martin Krupp, "Engineering Service Negotiation and Management Functionality for the Advanced Intelligent Network: An Application of Object-Oriented Technology," GLOBECOM '91, 1991.
7. S. Montesi, T. Di Stefano, N. Gatti, A. Pelaggi, "Object-Oriented Architecture of IN Service Control," Proceedings TINA '92, 1992.
8. Bellcore exchange, "The SPACE System: A New Dimension in Service Creation," January/February 1992.
9. Yukari Nagano, Naoki Uchida, "Verification of Customer-Defined Services in the Intelligent Network," International Council for Computer Communication Intelligent Networks Conference, Tampa, May 4-6, 1992.
10. Naoki Uchida and Akira Miura, "Customer-Defined Service Model and Definition Method for Intelligent Networks," ICC '91, 1991.
11. Shigehiko Suzuki, "IN Rollout in Japan," IEEE Communications Magazine, March 1993.
12. Richlien Hoang and Amir Atai, "Service Verification Testing for Intelligent Network Services," GLOBECOM '90, 1990.
13. M. W. Beckner and R. L. Fike, "Network Service Creation-a Key to the Future," ISS '90, 1990.
14. Richard A. Orriss, "Creation-Making the Promise a Reality," ICC '93, 1993.
15. Bellcore, Advanced Intelligent Network Release 1 Update(1992), Special Report, SR-NWT-002247, December 1992.
16. Ovum, ETRI Quarterly Report: Intelligent Network, February 1993.



최 정 훈

- 1985년 2월 : 경북대학교 전자공학과 졸업
- 1987년 2월 : 한국과학기술원 전산학과 졸업(석사)
- 1987년 2월 ~ 현재 : 한국전자통신연구소 통신망지능 연구실 선임 연구원
- 1992년 9월 ~ 현재 : 한국과학기술원 박사과정 재학중
- 관심분야 : 소프트웨어 개발 환경, CASE, 그래픽 사용자 인터페이스, 차세대 지능망 기술중 특히 서비스 개발 환경



권 용 래

- 1969년 2월 : 서울대학교 물리과 졸업
- 1971년 2월 : 서울대학교 대학원 이학석사(물리학)
- 1971년 9월 ~ 1974년 9월 : 육군사관학교 전임강사
- 1978년 12월 : 미국 피츠버그 대학 이학 박사(물리학)
- 1978년 8월 ~ 1983년 7월 : 미국 Computer Sciences Cooperation 연구원
- 1983년 8월 ~ 현재 : 한국과학기술원 전산학과 부교수
- 관심분야 : Software Specification, Development Tools, CASE, Program Verification, Human Computer Interface, Groupware.