

CORBA를 이용한 TMN 기반의 지능망 번호 이동성 서비스 관리 시스템 설계 및 구현

(Design and Implementation of a TMN-based Intelligent Network Number Portability Service Management System using CORBA)

안 숙 경[†] 홍 원 기^{**} 김 상 기^{***}
(Suk-Kyung An) (Won-Ki Hong) (Sang-Ki Kim)

요 약 번호 이동성은 사용자가 전화 번호의 변경 없이 통신 사업자, 지역, 서비스 종류를 바꿀 수 있는 능력을 제공하는 지능망 서비스이다. 번호 이동성은 통신 시장의 경쟁을 활성화 시키기 위해 반드시 필요한 서비스로, 최근 많은 나라에서 통신 사업자간에 규율을 정하고 번호 이동성 서비스를 도입하기 위한 작업을 추진 중에 있다. 번호 이동성 서비스를 제공하기 위해서는 통신 사업자들이 모든 고객의 정보를 각각 관리해야 하므로, 공통된 고객 정보를 관리하기 위한 단일화된 방법이 필요하다. 이러한 기능을 담당하는 것이 NPAC이다. 한편, 최근 CORBA는 분산 시스템 관리를 위한 주요 기술로 대두되고 있으며, CORBA를 지능망에 적용하기 위해 지능망, TMN, CORBA의 통합에 관한 연구가 진행되고 있다. 이러한 연구의 일환으로, 본 논문에서는 번호 이동성 서비스 관리에 TMN과 CORBA를 적용하기 위한 연구를 소개한다. 우선 CORBA 기반의 번호 이동성 서비스 관리 구조를 기술하고, CORBA를 이용하여 NPAC을 구현하는 방안에 대해 설명한다.

Abstract Local Number Portability (LNP), an Intelligent Network (IN) service, provides the ability to retain existing telephone numbers when switching from one telecommunications carrier to another, from one region to another, or from one service to another. LNP is a key service for increasing competition in the local telephone marketplace. Many countries are now establishing detailed rules for ensuring competition between network operators. Because many major carriers will deploy their own LNP databases, a single access point must be provided to effectively manage and distribute updates to the common regional LNP database. This is called Number Portability Administration Center (NPAC). Recently, CORBA is being accepted as one of the key technologies for developing distributed application systems. To use CORBA for managing IN services, several standard organizations are working towards the integration of IN, TMN and CORBA. In this paper, we present our work on applying the TMN and CORBA technology for the service management of LNP. We first propose a CORBA-based LNP service management system architecture. We then present the design and implementation of a LNP service management system using CORBA.

1. 서 론

통신 시장의 성장에 따라 여러 통신 사업자들이 경쟁을 벌이고 있다. 하지만 후발 사업자들은 새로운 고객을 유치하는데 많은 어려움을 겪고 있다. 현재 가입자가 사업자를 변경하려면 사용하던 전화번호를 변경해야 하는데 이것은 새로운 사업자들이 고객을 유치하는데 걸림돌이 되고 있다. 번호 이동성[1]은 바로 이러한 문제를

[†] 비 회 원 : 데이콤 종합연구소 연구원

skan@halla.dacom.co.kr

^{**} 종 신 회 원 : 포항공과대학교 컴퓨터공학과 교수

jwkhong@postech.ac.kr

^{***} 비 회 원 : 한국전자통신연구원 연구원

sangkim@etri.re.kr

논문접수 : 1999년 1월 18일

심사완료 : 1999년 9월 8일

해결해 주는 것으로 가입자가 전화번호를 변경하지 않고 통신 사업자, 가입 지역, 사용 서비스 등을 변경할 수 있도록 해준다. 이러한 번호 이동성은 통신 시장에서 사업자간 경쟁을 활성화 시키고 가입자에게 보다 양질의 서비스를 제공하기 위해서 필수적인 것으로 세계 각국에서 활발하게 진행되고 있다[2].

번호 이동성은 여러 방법으로 구현할 수 있으나 장기적인 관점에서 지능망을 이용한 방법이 고려되고 있다. 루팅에 필요한 정보를 별도로 관리하고 질의를 통해 정보를 얻어 호를 처리한다. 이를 위해 여러 통신 사업자들이 루팅 정보를 개별적으로 관리한다. 따라서, 이러한 사업자들간 정보의 내용을 통일 시키기 위한 새로운 관리 방안이 필요하게 된다. 이러한 관리 방법 중 한 가지가 NPAC(Number Portability Administration Center)을 사용하는 방법이다[9, 10]. 통신 사업자들은 NPAC의 루팅 정보를 기준으로 이를 각각 복사하여 사용하며 이를 위해 LSOA(Local Service Order Administration), LSMS(Local Service Management System)를 구현하여 NPAC과 연동해야 한다. 이러한 NPAC에 대한 구현은 현재 TMN을 기반으로 LSMS 및 LSOA와의 연동도 CMIP을 사용하고 있다.

본 논문에서는 NPAC을 이용하여 번호 이동성 서비스를 구현하는데 있어서, TMN개념에 입각하여 OMG(Object Management Group)[13]에서 제안한 분산객체 기술인 CORBA(Common Object Request Broker Architecture)[12]를 적용하는 방안을 소개하고 구현 예를 통해 이를 설명하고자 한다. NPAC과 LSOA, LSMS는 지능망의 서비스 관리에 해당하는 부분으로 분산 객체 기술의 적용을 시작하기에 가장 적합한 모델이다. 이러한 관점에서 본 논문은 번호 이동성의 구현 뿐 아니라 지능망에 분산 객체 기술을 적용하는데 좋은 예가 될 것이다.

본 논문의 구성은 2장에서 번호 이동성에 대해 좀 더 자세히 알아본다. 3장 이하는 CORBA를 이용한 NPAC의 구현에 관한 사항으로 3장에서는 NPAC의 분석, 4장에서는 설계, 5장에서는 구현에 관한 내용을 설명한다. 6장에서는 결론을 맺고 향후 연구 계획에 관하여 기술한다.

2. 번호 이동성

번호 이동성(Number Portability)[1, 4, 5]이란 가입자가 전화번호를 변경하지 않고 통신 사업자, 가입 지역, 사용 서비스 등을 변경할 수 있는 능력 또는 그러한 능력을 제공하는 서비스를 의미한다. 번호 이동성은 크

게 통신 사업자, 통신 시장 경제와 일반 전화 이용자의 세 가지 측면의 이득을 가져오며, 결과적으로 국가 경제에 큰 이득을 가져올 것으로 분석되었다. 세계 각국의 전문 조사 기관의 조사 결과가 이러한 분석을 뒷받침해 주고 있으며, 이에 따라 많은 나라들이 번호 이동성의 구현을 추진하고 있다.

2.1 국제적 번호 이동성 추진 현황

번호 이동성은 현재 미국, 영국, 홍콩 등에서 서비스 제공을 시작했으며, 대상 지역을 확대하거나 서비스 범위를 확대하고 있다. 이 외 다른 나라들도 번호 이동성 서비스를 준비하거나 계획을 수립 중에 있다[2]. 번호 이동성을 구현하는데 가장 큰 문제는 번호 이동성을 제공할 사업자로부터 제공에 대한 동의를 얻어내는 일이다. 이 일은 사업자의 손익과 직접적으로 관계되므로 쉽게 동의를 얻어내기 어려운 문제로, 정부의 주관 기관이 주도하여 문제를 해결하고 있다. 또한 국제 표준화 기관인 ITU-T에서도 번호 이동성 관련 문제를 전담하도록 연구반을 신설하여 번호 이동성에 대한 표준을 추진 중에 있다[7, 8].

2.2 번호 이동성 구현 방법

번호 이동성을 구현하는 방법으로는 교환기 기반의 단기 해결 방법과 지능망 기반의 장기 해결 방법이 있다. 교환기 기반의 단기 해결 방법은 한 가입자에게 두 개의 번호를 부여해 호를 재전달하는 방식으로, 현재 호주와 영국에서 사용되고 있다. 그러나 궁극적인 해결 방법으로는 지능망을 이용하여 이동한 착신자의 호를 루팅하는 방법이 고려되고 있다. 호를 루팅하기 위해 사용하는 정보의 종류에 따라 CPC(Carrier Portability Code), LRN(Location Routing Number), Name &

표 1 번호 이동성 구현 방법

종 류	구현방법	장 점	단 점
RCF (Remote Call Forwarding)	변경된 번호로 새로운 호 생성	구현 용이	두 개의 번호 사용
LRN (Location Routing Number)	루팅에 사용할 일련번호를 교환기에 부여	모든 번호 이동성, 부가 서비스 제공 가능	구현 복잡
CPC (Carrier Portability Code)	지역별로 사업자마다 지역번호 할당	구현 용이, 부가 서비스 가능	사업자 이동성만 제공, 지역코드 사용
Name & Address	가입자마다 전화 번호와 위치번호 부여	모든 번호 이동성, 부가 서비스 제공 가능	구현 매우 복잡

Address 방법 등이 존재한다. 이러한 방법 중에서 미국에서는 LRN을 표준으로 채택, 번호 이동성을 구현하고 있다. 표 1에서 여러 가지 구현 방법에 대한 비교 내용을 보여주고 있다.

2.3 번호 이동성 관리 모델

지능망을 이용해 번호 이동성을 제공하기 위해 각 사업자는 호 루팅 정보를 데이터베이스를 이용해 관리한다. 그러나 기존 서비스에 대한 정보와 달리 번호 이동성을 위한 정보는 각 사업자 자신의 고객 뿐 아니라 다른 사업자의 고객에 대한 정보를 포함하며, 서비스 제공 사업자들이 모두 동일한 내용의 정보를 가지고 있어야 한다. 이를 위해서 각 사업자들의 정보를 일률적으로 관리할 수 있는 방안이 필요하게 된다. 이러한 방법으로 사업자와 별도로 마스터 시스템을 두어 루팅 정보를 관리하는 방법과, 각 사업자들간에 서로 연동을 통해 통일된 정보를 유지하도록 하는 두 방법을 생각해 볼 수 있다.

첫 번째 방법은 가입자 정보에 대한 마스터 시스템을 두는 방법으로, 현재 미국에서 사용하고 있으며, NPAC이 이에 해당된다. 모든 가입자 정보의 변경은 NPAC에서 일어나며, 각 사업자는 가입자 정보 변경을 통보하는 기능을 포함한 LSOA를 운영하며, NPAC으로부터 변경된 정보를 받아가는 기능을 포함한 LSMS를 구현하여 NPAC과 연동해야 한다. 그림 1은 NPAC을 기반으로 한 번호 이동성 구조를 보여주고 있다. 미국에서는 NANC(North America Numbering Council)를 구성하여 NPAC에 대한 스펙을 제작하고 있다[9, 10]. 다음 방법은 각 사업자가 다른 사업자들과 데이터베이스의 내용을 주기적으로 확인하여 일관된 정보를 유지하는 방법이다. 그림 2는 사업자 데이터베이스를 기반으로 한

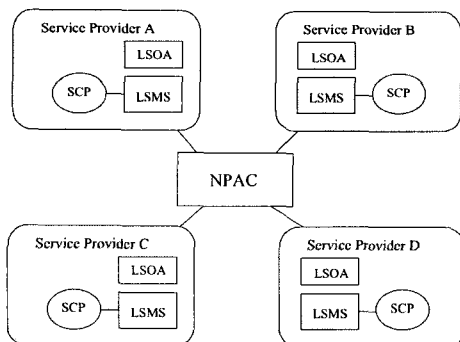


그림 1 중앙 집중 방식의 번호 이동성 관리

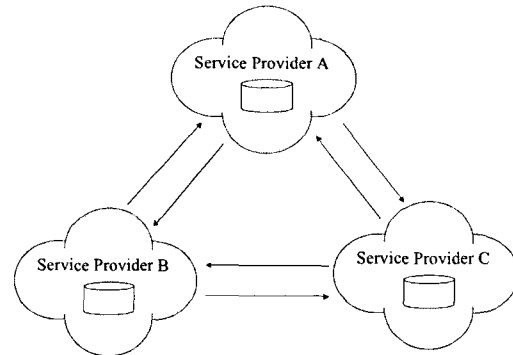


그림 2 분산 방식의 번호 이동성 관리

분산 관리 방법을 보여주고 있다. 현재 이 방법은 홍콩에서 사용되고 있으며, OFTA(Office of the Telecommunications Authority)에서 각 사업자들이 관리해야 할 정보의 내용과 다른 사업자와 확인을 위한 프로토콜 등을 정의하고 있다[6].

2.4 번호 이동성에 CORBA의 적용 방안

번호 이동성 서비스 시스템도 분산 시스템이므로 CORBA를 이용해 여러 장점을 얻을 수 있다. 이 장에서는 이상에서 언급한 지능망을 이용한 번호 이동성 서비스에 CORBA를 적용하는 방법에 대하여 고찰해 보고자 한다. CORBA는 OMG에서 제안한 분산 객체 구조로 분산 시스템 개발에 여러 가지 장점을 제공한다.

새로운 서비스가 추가 됨에 따라 지능망 시스템은 추가된 서비스를 수용해야 한다. 기존 서비스가 수정된 경우에도 지능망 시스템은 이를 반영하기 위해 변경해야 한다. 이렇게 서비스에 따라 지능망 시스템은 끊임없이 추가, 변경을 반복하게 된다. 그러므로 지능망은 이러한 변화를 수용하기 위해 확장이 용이하며 구성이 유연해야 한다. 이러한 특징은 바로 CORBA를 이용하여 얻을 수 있다.

CORBA를 기반으로 구현하면 시스템 구조를 상황에 따라 유연하게 설계할 수 있다. 예를 들어, NPAC을 구현할 경우 NPAC 자체를 기능별 또는 용량별로 여러 대의 서버로 분리하여 구축할 수 있다. NPAC에서는 상당히 많은 양의 정보를 관리해야 하므로 하나의 서버에서 모든 정보를 관리하는 것은 성능 저하를 가져올 수 밖에 없다. 실제로 미국에서는 지역별로 7개의 NPAC이 운영되고 있다. NPAC을 CORBA로 구현할 경우 LSOA와 LSMS는 특정 정보를 가지고 있는 NPAC을 구별할 필요없이 ORB(Object Request Broker)와

COSS(Common Object Service Specifications)를 통해 원하는 요청을 할 수 있다. 이것은 CORBA에서 제공하는 위치 투명성에 의한 것으로, CORBA 클라이언트로 구현된 LSOA는 서버인 NPAC의 위치를 지정하지 않고 서버의 위치와 무관하게 서비스를 받을 수 있다.

CORBA 객체는 컴퓨팅 환경에 무관하므로 각 시스템 별로 적절한 프로그램 언어, 프로토콜, 플랫폼 등을 선택해서 구현할 수 있다. 사용자 인터페이스가 필요한 부분에는 Java를 이용한 인터페이스를 사용자에게 제공해 줄 수 있으며, 다른 시스템은 C++ 또는 C로 구현할 수 있다. 어떤 시스템이든 ORB를 통해 연동할 수 있기 때문이다. 또한 CORBA IDL을 사용하여 시스템 기능을 명시하므로 통신을 위한 별도의 프로토콜을 필요로 하지 않는다.

다음은 각 관리 구조별로 CORBA를 적용하는 방법이다.

2.4.1 중앙 집중형

NPAC을 통한 중앙 집중형 번호 이동성 관리 구조에 CORBA를 적용하는 방법은 적용 범위별로 생각해 볼 수 있다 (그림 3 참조). 우선 NPAC은 통신 사업자가 아닌 제 3사업자에 의해 관리되고 새로 구축이 필요하므로 CORBA를 적용하기에 가장 용이할 것으로 판단된다. 따라서, NPAC만 CORBA를 이용해 구현하는 경우, 통신 사업자의 서비스 관리 시스템까지 CORBA를 적용한 경우, 그리고 그 이하의 SCP, SSP까지 적용하는 경우로 나누어 살펴보고자 한다. 이러한 내용을 표로 정리하면 표 2와 같다.

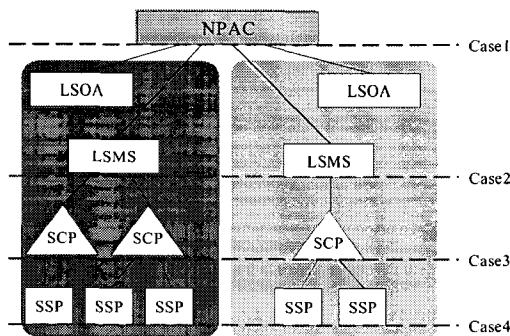


그림 3 중앙 집중형 관리 구조에 CORBA 적용 방안

CORBA 시스템과 타 시스템의 연동을 위해 프로토콜 변환을 수행하는 게이트웨이가 필요하며, 이러한 연구는 JIDM을 주축으로 수행되고 있다. 여기서 언급되는

프로토콜은 CMIP으로 CORBA/CMIP 게이트웨이를 살펴보면, 기존의 CMIP 에이전트를 CORBA 매니저가 관리하는 경우에 필요한 게이트웨이와 CORBA 에이전트를 CMIP 매니저가 관리하는 경우에 필요한 게이트웨이로 구분된다. CORBA를 적용하는 경우 필요한 게이트웨이는 대부분 전자에 해당하며 이에 대해서는 JIDM의 제안 [15, 18]에 따라 다양한 제품이 나와있다. 하지만 후자의 경우는 수요가 거의 없고 게이트웨이에 대한 스펙은 나와있으나 제품은 나와 있지 않다. 그러므로 CORBA 에이전트를 CMIP 매니저가 관리하는 구조에서는 게이트웨이의 개발도 필요하다. 표 2의 기술적 고려 사항에서 LSOA/NPAC 연동과 LSMS/NPAC 연동은 각각 CORBA 에이전트/CMIP 매니저 구조를 포함하고 있다. 그러므로 별도의 게이트웨이를 개발해야하는 부담이 있다.

표 2 중앙 집중형 번호 이동성 관리에 CORBA의 적용 방안

구분	CORBA 적용 범위	기술적 고려 사항
Case 1	NPAC	LSOA/NPAC, NPAC/LSMS 연동
Case 2	NPAC, LSOA, LSMS	LSMS/SCP 연동
Case 3	NPAC, LSOA, LSMS, SCP	성능, SCP/SSP 연동
Case 4	NPAC, LSOA, LSMS, SCP, SSP	성능, 기존 SSP와의 연동

SCP나 SSP까지 CORBA를 적용하는 경우에 있어서는 CORBA의 성능이 문제가 된다. No. 7 프로토콜의 경우 라운드 트립 타임이 적어도 500 miliseconds가 필요하다. 하지만 현재 CORBA로는 이러한 성능을 낼 수 없다. CORBA의 성능을 개선하기 위해 실시간 CORBA 등과 같은 연구가 진행되고 있으며 지속적으로 개선될 것이다. 또한 SCP나 SSP가 CORBA 인터페이스를 갖는 경우, 모든 SCP와 SSP가 CORBA를 가지면 문제가 없으나, 일부만 CORBA로 구현된 경우에는 기존 SCP나 SSP와의 연동에 문제가 있다. 이러한 부분도 연구가 진행 중에 있다.

결론적으로 현재 상황에서 CORBA를 적용하기에는 Case 2에서와 같이 NPAC과 사업자 단의 서비스 관리 시스템까지를 CORBA로 구현하는 것이 가장 적합하다고 볼 수 있다.

2.4.2 분산형

사업자의 서비스 관리 모듈 간 연동을 통한 분산형 번호 이동성 관리 구조에 CORBA를 적용하는 방법 또한 동일하게 적용 범위별로 생각해 볼 수 있다. 분산형의 경우에는 사업자 단의 서비스 관리 시스템을 CORBA를 이용해 구현하는 경우와 그 이하의 SCP, SSP까지 적용하는 경우로 나누어 볼 수 있다. 사업자인 SMS에 CORBA를 적용하는 경우, 사업자들의 SMS는 CORBA IDL로 정의된 인터페이스를 가지며, CORBA ORB를 통해 연동한다. 그러나, 일부 사업자인 CORBA를 적용하여 CORBA SMS와 기존 지능망 SMS가 공존하게 되는 경우에는 CORBA SMS와 지능망 SMS의 연동을 위해 마찬가지로 CORBA 에이전트/CMIP 매니저 구조의 게이트웨이가 필요하다.

SCP와 SSP까지 CORBA를 적용하는 경우에 대대선 중앙 집중형과 동일하다.

3. NPAC의 분석

NPAC[9, 10]은 번호 이동성 가입자 정보 관리 시스템으로 번호 이동성은 NPAC에 의해서 지원, 유지, 관리된다. 이하에서 언급하는 모든 NPAC에 관한 내용은 미국 NANC에서 제작한 NPAC 스펙[9, 10]을 기반으로 한다.

모든 가입자 루팅 정보의 변경은 NPAC에서 관리하며, 이를 위해 NPAC은 통신사업자의 LSOA, LSMS와 연동을 통해 LSOA로부터 가입자 정보를 입력 받고 변경된 정보를 LSMS에게 전달하여 새로운 정보를 기반으로 서비스를 수행하도록 한다.

번호 이동성을 제공하기 위해 NPAC에서는 서비스 제공 (Provisioning Service), 서비스 해제 (Disconnect Service), 서비스 복구 (Repair Service), 충돌 제거 (Conflict Resolution), 재해 복구 및 백업 (Disaster Recovery and Backup), 주문 취소 (Order Cancellation), 감사 요청 (Audit Request), 리포트 요청 (Report Request), 자료 관리 (Data Management) 등의 기능을 제공한다.

NPAC은 가입자 정보를 받고, 변경된 가입자 정보를 사업자에게 전달하기 위해 LSOA, LSMS와 통신한다. LSOA와 NPAC 간 인터페이스는 매니저/에이전트 관계로 이루어지며, LSMS와 NPAC간 인터페이스는 수행 기능에 따라 매니저/에이전트 또는 에이전트/매니저 관계로 형성된다. NPAC의 인터페이스와 각 인터페이스에서 제공하는 기능을 정리하면 표 3과 같다.

표 3 NPAC의 관리 자료

관리 정보	상세 항목
가입자 정보 (Subscription Version Data)	가입자에 대한 전화번호, LRN, 가입 사업자
사업자 정보 (NPAC Customer Data)	번호 이동성 서비스에 참여하는 사업자에 대한 이름, 주소, 망 정보
망 정보 (Network Data)	번호 이동성을 제공하는 망에 대한 토폴로지 및 루팅 정보

표 4 NPAC 인터페이스

인터페이스	제공 기능
LSOA-NPAC 인터페이스	가입자 관리(Subscription Administration) 감사 요청(Audit Request) 통지(Notification) 사업자 정보 관리(Service Provider Data Administration)
LSMS-NPAC 인터페이스	가입자 정보와 망 정보 다운로드 (Subscription Version and Network Data Download) 사업자 정보 관리 (Service Provider Data Administration) 통지(Notification)

이러한 기능을 제공하기 위해 NPAC은 가입자, 사업자 및 망에 대한 정보를 관리한다.

4. NPAC의 설계

본 논문의 연구 범위 내에서 CORBA를 이용해 구현한 NPAC을 CONPAC (CORba based NPAC)이라 명명하였다. 이 장에서는 CONPAC 설계 시 고려했던 점들을 기술한다. 전체적인 시스템의 구성과 TMN에 적합하게 명시된 NPAC을 CORBA로 구현하기 위해 고려했던 사항들을 기술한다.

CORBA를 이용해 시스템을 개발한다면 기존에 TMN 기반하에 명시된 시스템과 동일한 기능을 TMN 기반과 관계없이 CORBA 기반으로 정의하여 개발할 수도 있다. 하지만 이것은 이미 설계된 사항에 대해 중복 투자가 필요하게 된다. 또한 대부분의 개발자들이 TMN 환경에 익숙하리라는 관점에서 TMN을 무시하고

CORBA 기반의 개발 환경만 고려하는 것은 적절하지 않다. 따라서, 본 연구에서는 TMN 기반으로 명시된 시스템 명세를 최대한 이용하고 가능한 TMN과 유사한 환경을 제공하는 것을 전제로 하였다.

4.1 CORBA IDL 생성

TMN 기반으로 GDMO/ASN.1을 이용해 명시된 인터페이스 규격을 따라 CORBA를 이용해 시스템을 개발하려면 가장 먼저 GDMO/ASN.1에 대한 CORBA IDL 매핑이 필요하다. JIDM에서 작업한 CMIP/CORBA Gateway에서는 이러한 작업을 Specification Translation이라고 칭하고 있다. JIDM에서는 Specification Translation[15]에 필요한 수행 작업을 명시하고 있는데 이것이 거의 업계 표준이 되고 있다. 본 연구에도 JIDM의 명세 내용에 준하여 변환을 수행하였다.

그림 4는 본 연구에서 사용한 Specification Translation 구조를 보여주고 있으며, 그림 5와 그림 6

은 이를 기반으로 수행된 ASN.1/CORBA IDL, GDMO/CORBA IDL 간 변환 예를 보여준다.

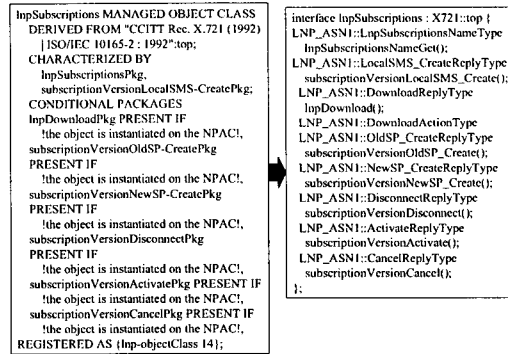


그림 6 GDMO to CORBA IDL 매핑 예

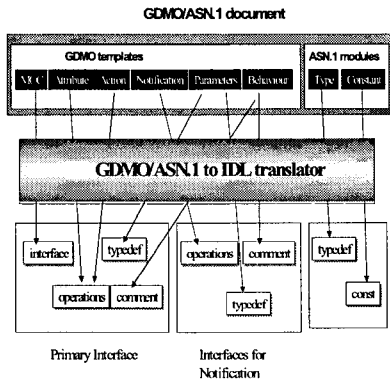


그림 4 Specification Translation 구조

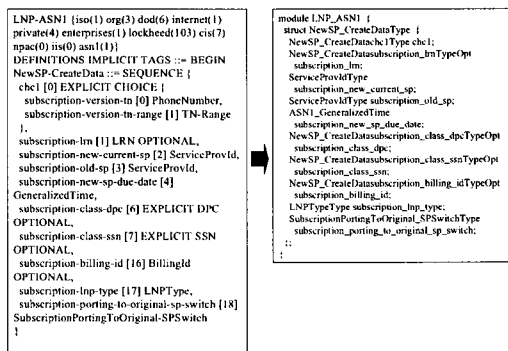


그림 5 ASN.1 to CORBA IDL 매핑 예

4.2 CMIP/CORBA IDL Operation 매핑

NANC 문서에서 정의된 NPAC 인터페이스는 M-CREATE, M-DELETE, M-SET, M-GET, M-EVENT-REPORT, M-ACTION 등의 프리미티브들을 이용해 구현된다. 본 연구에서는 CMISE 프리미티브로 작성된 인터페이스를 분산 객체 지향 기술인 CORBA를 이용해 구현하였다. CORBA를 이용해 구현된 경우, 모든 프로세스간의 통신은 CORBA의 ORB를 통해 이루어지므로 하위 통신 프로토콜은 문제가 되지 않는다.

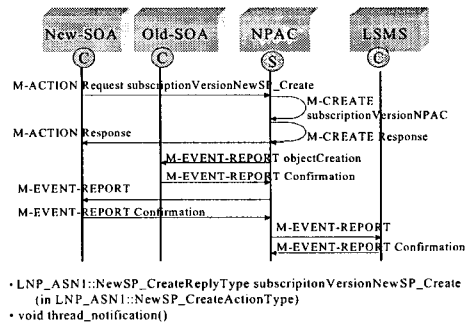


그림 7 CMIP to CORBA IDL Operation 매핑

CMIP/CMISE 오퍼레이션으로 정의된 기능은 CORBA IDL로 대체된다. 경우에 따라 하나의 CMIP operation이 하나의 IDL method로 대응될 수도 있고, 여러 개의 operation들이 하나의 IDL method로 대응될

수 있다. 예를 들어, 그림 7에 표현된 CMIP operation들은 subscriptionVersionNewSP_Create()라는 함수와 이벤트 처리를 위한 thread_notification()으로 모두 구현된다.

4.3 NPAC 인터페이스에 대한 Naming Graph

TMN에서는 관리 대상인 MANAGED OBJECT (MO)를 정의하고 이들의 관계를 NAME BINDING으로 설정하여 인터페이스의 MIT(Managed Information Tree)를 구성한다. MIT의 각 노드는 하나의 MANAGED OBJECT로 각각 정의된 객체를 표현하고 이름을 가진다.

이러한 MIT를 CORBA에서 구현하기 위해 우선 GDMO에 대한 CORBA IDL이 필요하다. 그리고 정의된 객체에 MIT와 같은 Naming Hierarchy를 제공해주어야 한다. CORBA에서는 이렇게 객체에 이름을 지정하는 방법을 COSS인 Naming Service를 이용하여 제공할 수 있다. Naming Service는 COSS 중 하나로 대부분의 CORBA 제품에서 지원이 되고 있다. 그러나 TMN의 MIT와 CORBA Naming Service에서 제공하는 Naming Graph의 가장 큰 차이점은 Naming Graph에서는 단말 노드만 객체를 지정할 수 있고 중간 노드들은 단말 노드를 명명하기 위한 문맥만을 제공한다는 점이다. 그러므로 TMN의 MIT를 그대로 CORBA Naming Graph로 표현할 수 없으며 별도의 매핑 방법이 필요하다.

GDMO에서는 명명을 NAME BINDING에서 명시한 Attribute의 OBJECT IDENTIFIER와 해당 값으로 구성된 쌍의 리스트로 표현된다. CORBA Naming Graph는 노드가 아닌 에지에 이름을 주고 이들의 리스트로 단말 노드를 표현한다. 각 에지의 이름은 id와 kind라는 스트링의 쌍으로 표현된다. 본 연구에서는 GDMO의 NAME BINDING을 CORBA Naming Graph에 매핑

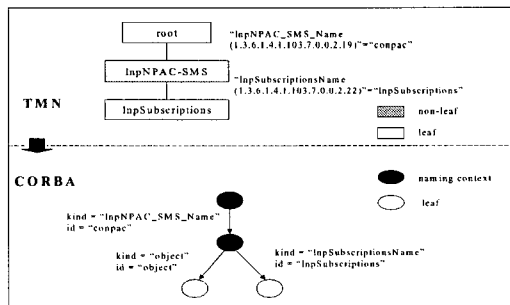


그림 8 TMN Naming과 CORBA Naming 매핑

하기 위해 CORBA Naming Graph의 에지의 kind를 Attribute의 타입으로 id를 Attribute의 값으로 대응시켜 중간 노드를 형성한다. GDMO의 중간 노드에 해당하는 MO를 표현하기 위해 새로운 단말 노드를 추가한다. 이를 그림으로 표현하면 그림 8과 같다. 명명 문맥은 동일하게 매핑이 되고 중간 노드에 대해선 "object"라는 에지를 가지는 단말 노드를 새로 생성한다. 단말 노드는 단일한 단말 노드로 매핑된다.

5. NPAC 구현

이 장에서는 분석, 설계의 내용에 따라 CONPAC을 구현한 내용을 기술한다. TMN과 CORBA의 연동을 위해 JIDM에서는 Specification Translation 외에 Interaction Translation을 또한 제안하고 있다. Interaction Translation은 이미 정의된 인터페이스 외에 Specification Translation에서 다 표현이 안된 부분과 실제 연동상에서 기능을 수행하기 위해 필요한 부분에 대한 처리를 나타낸다. 하지만 JIDM에서 제안한 Interaction Translation은 OMA의 모든 COSS를 가정하는 것으로 실제 구현을 하기 위해서는 여러 가지 문제가 있다. 이에 반해 CONPAC은 COSS의 Event Service와 Naming Service 만을 사용하여 구현되었다.

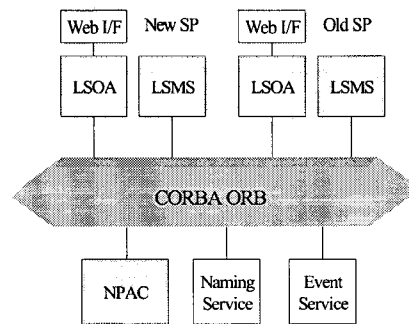


그림 9 CONPAC 시스템 구성도

그림 9는 CONPAC 시스템 구성도를 보여준다. 기존 사업장에서 새로운 사업자로 이동을 보이기 위해 사업자 부분인 LSOA와 LSMS는 두 쌍이 존재하고 가입자의 정보 입력을 위한 LSOA의 인터페이스는 웹을 이용한 GUI를 제공한다. NPAC의 MIB을 볼 수 있는 MIB Browser가 역시 웹을 통해 제공된다. 그림 10은 웹을 통해 제공되는 MIB Browser 데모 화면이다.

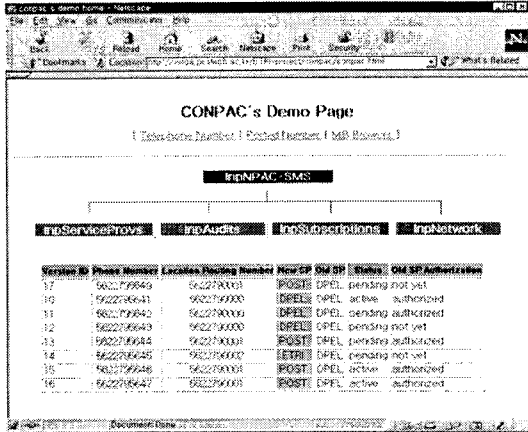


그림 10 CONPAC MIB Browser 데모 화면

그림 11의 CONPAC 연동 예는 일반적인 CONPAC의 연동 과정을 보여준다. NPAC은 시스템 초기화 과정에서 자신이 제공하는 MIB의 내용을 Naming Service에 등록한다.

- (1) 통신 사업자의 LSOA는 MIB 정보를 얻기 위해 Naming Service에 접속한다.
- (2) object 이름을 통해 object reference를 얻는다.
- (3) object reference를 얻으면 이것을 사용하여 NPAC의 object 구현을 사용할 수 있다.
- (4) NPAC에서는 기능 수행 중에 필요에 따라 Event Service를 통해 이벤트를 발생시킨다.
- (5) LSOA는 이벤트를 받아 적당한 조치를 취한다.

CONPAC의 구현은 IONA사의 Orbix 2.3 CORBA 제품을 사용했으며, C++과 사용자 인터페이스를 위해 CGI를 사용했다.

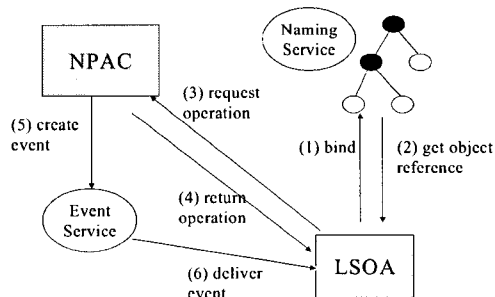


그림 11 CONPAC 연동 예

6. 결론 및 향후 과제

본 논문은 지능망 서비스 관리에 대한 연구의 일환으로 분산 객체 기술(CORBA)을 이용하여 지능망 서비스 관리 방안에 대한 연구를 통해 번호 이동성 서비스 관리에 대한 적용 방안을 제시한다.

번호 이동성에 대해서 그 정의와 필요성, 국제적으로 추진되고 있는 상황, 지능망을 이용하여 번호 이동성 구현 및 관리 방안을 살펴보았다. 번호 이동성 관리 방안은 정보 관리 방식에 따라 마스터 데이터베이스를 두는 중앙 집중형과 각 사업자간 연동을 통해 정보를 유지하는 분산형으로 구분된다. 각각의 경우 CORBA를 적용하는 방안에 대해 제시하고 장점과 문제점을 지적하였다. 기술적인 측면으로 현재 가능한 방안은 Case 2가 가장 적합하다. Case 2는 NPAC과 사업자 단의 NPAC과 연동을 가지는 LSOA, LSMS까지 CORBA를 이용하는 방안으로 NPAC을 개발하면서 사업자들도 CORBA를 이용하여 LSOA, LSMS를 개발해야 한다.

본 논문에서는 지능망에 CORBA를 적용하기 시작하는 단계로 중앙 집중형 번호 이동성 관리 시스템인 NPAC의 프로토타입을 CORBA를 이용하여 구현하였다. 또한 NPAC의 기능을 보이기 위해 NPAC과 연동하는 사업자 단의 LSOA와 LSMS도 구현하였다. JIDM에서 제시하는 Specification Translation에 준하여 NPAC의 GDMO/ASN.1을 변환하여 CORBA IDL을 생성하였고, CORBA Naming Service와 Event Service를 사용하여 TMN을 사용해 명시한 NPAC을 CORBA를 이용해 구현하였다. 본 연구의 내용은 지능망과 CORBA의 연동에 좋은 예이다. 또한 TMN으로 구현된 기존 시스템을 CORBA 환경으로 통합할 경우 도움이 될 것이다.

본 연구에 추가하여 연구가 필요한 사항은 크게 두 가지로 나누어 볼 수 있다. 첫 번째 사항은 지능망에 CORBA 적용 범위를 확장하는 것이다. TINA나 OMG에서 지능망과 CORBA의 연동 방안에 대한 연구가 진행되고 있으나, 현재 상황으로는 CORBA의 성능이나 COSS의 구현 등의 문제로 적용이 어려운 상황이다. 앞으로 이러한 부분에 대한 연구가 더 필요하다.

두 번째 사항은 TMN 시스템을 CORBA를 이용해 개발할 경우의 문제점으로, 본 연구에서 해결책의 일부분을 제시하고 있으며 그 외 부분에 대한 추가 연구가 필요하다. 구체적인 사항으로는 GDMO 인터페이스를 Naming Service로 변환시킬 때, GDMO는 MO의 이름을 속성 값을 나타내는 임의의 구조로 지정할 수 있으

나, Naming Service에서는 Name Component의 string 만으로 표현할 수 있다는 한계가 있다. 또한 TMN과 CORBA의 연동에서 가장 큰 문제가 되는 Scoping & Filtering에 대한 사항이다. 이 부분은 JIDM의 Interaction Translation에 제안된 방법을 통해 해결할 수 있을 것이다.

참 고 문 헌

- [1] Nortel, Local Number Portability Backgrounder, <http://www.nortel.com/cool/lnp/background.html>.
- [2] 한국전자통신연구원, 번호 이동성 연구 동향, 주간기술동향 No. 98-29, pp. 1-17, 1998.
- [3] VERTEL, TMN tutorial, <http://www.webproforum.com/vertel/>.
- [4] DSET, Enabling Leadership in Local Number Portability (LNP), <http://www.dset.com/applications/apps1.html>.
- [5] Evolving Systems, LNP for CLECS: An Industry White Paper, <http://www.evolving.com/whitepapers/lnpclecpaper.html>.
- [6] Office of the Telecommunications Authority, Requirements for Number Portability by Database Solution, <http://www.ofta.gov.hk/number/hkta2102.html>, September 1996.
- [7] ITU-T, TD PL/11-65R1, Scope and Number Portability Capability Set 1 Architecture, May 1998.
- [8] ITU-T, TD PL/11-66R1, Capability Set 1 Portable Number Call Control for Service Provider Portability, May 1998.
- [9] North America Numbering Council, Functional Requirements Specification, NPAC SMS, ver. 1.8. Feb. 1998.
- [10] North America Numbering Council, Interoperable Interface Specification, NPAC SMS, ver. 1.8. Feb. 1998.
- [11] M. Kockelmans and E. de Jong, Overview of IN and TMN Harmonization, *IEEE Communications Magazine*, Vol. 33, No. 3, March 1995. pp. 62-66.
- [12] OMG, The Common Object Request Broker: Architecture and Specification, Revision 2.0, Jul. 1995. <http://www.omg.org/news/begin.htm>.
- [13] OMG, <http://www.omg.com/>.
- [14] EURESCOM Project P508, "CORBA as an Enabling Factor for Migration from IN to TINA," <http://asp2.etri.re.kr/bios/mscho/in.html>, 1996.
- [15] X/Open, Document Number P509, Inter-domain Management: Specification Translation, 1997.
- [16] M. S. Cho, "Introduction of Intelligent Network Technology," <http://asp2.etri.re.kr/bios/mscho/in.html>, 1996.
- [17] ETSI, "Baseline Document on the Integration of IN and TMN," ETSI NA43308, 1992.
- [18] JIDM, "JIDM Interaction Translation," Initial Submission to OMG's CORBA/TMN Interworking, RFP2, 1998.



안 숙 경

1993년 한국과학기술원 전산학 학사 졸업.
1993년 ~ 1996년 테이콤 종합 연구소 근무.
1998년 포항공과대학교 정보통신대학원 석사 졸업. 1999년 ~ 현재 테이콤 종합연구소 근무. 관심분야는 네트워크 및 서비스 관리, 분산 시스템

홍 원 기

정보과학회논문지: 컴퓨팅의 실제
제 6 권 제 1 호 참조



김 상 기

1981년 서울대학교 산업공학 학사 졸업.
1983년 서울대학교 산업공학 석사 졸업.
1984년 ~ 현재 한국전자통신연구원 책임연구원(팀장). 관심분야는 지능망, 에이전트 시스템, 분산/이동 객체 응용