

교환 시스템의 국 데이터 자동 생성을 위한 데이터 분석 및 설계 방법

정회원 정 창 신*, 정 순 기**

Data Analysis and Design Method for automatically generating Office Data of Switching System

Chang-Shin Chung*, Soon-Key Jung** *Regular Members*

요 약

교환 시스템에서 사용되는 국 데이터는 시스템 형상 데이터, 교환에 필요한 데이터 및 시스템의 운용보전에 관련된 데이터 등, 그 규모가 방대할 뿐만 아니라 데이터의 기능 및 설계방법에 따라 그 종류도 다양하다. 이러한 교환 시스템의 국 데이터는 해당 시스템의 설치 장소에 종속되며, 국 데이터의 오류 발생으로 인해 파급되는 영향은 매우 크게 나타난다. 본 논문에서는 국 데이터의 오류로 인해 교환 시스템의 개발 및 현장 적용 단계에서 발생하는 많은 시간과 노력을 축소시키고, 시스템의 신뢰도를 제고시키기 위하여 교환국의 설치 물량과 시스템 형상에 따라 국 데이터를 자동 생성할 수 있는 데이터 분석 및 설계 방법을 제안하였다.

ABSTRACT

The office data for telecommunication switching systems included system configuration data, processing data, maintenance and administration data on a very large scale. Those data are dependent on functions of the system and the place of system installation. The effect of errors of office data is very serious. In order to reduce time and effort on the system development phase and to enhance system reliability, in this paper we proposed a data analysis and design method for automatically generating office data that are dependent on installation capability and system configuration of the switching office.

I. 서 론

교환 시스템은 전화 가입자 상호간을 연결시켜 주는 스위칭 장치로서, 가입자가 통화하고자 하는 상대 가입자의 전화번호를 직접 다이얼링하면 자동적으로 상대 가입자의 전화회선을 연결시켜주는 기능을 수행한다.

교환 시스템은 분산된 데이터를 실시간으로 접근, 처리해야 하므로 상당한 시간적인 제약성을 갖는다. 그러므로 정확하고도 신속한 데이터 접근을 위해서는 시스템의 성능 제고는 물론, 신뢰성과 가용성을 높일 수 있는 데이터베이스 관리 시스템(DBMS:

DataBase Management System)의 사용이 필수적이다.

복잡도가 높은 정보를 신속하게 처리할 수 있는 DBMS를 사용함으로써 교환 시스템 내부에서 정보의 표현 방법에 따라 발생하는 데이터 운용 및 관리상의 문제점들을 제거할 수 있다. 예를 들면, 데이터 중복성(redundancy)의 감소를 통해 데이터 일관성(consistency)을 증대시킬 수 있으며, 데이터 투명성(transparency)을 보장함으로써 프로그래머는 데이터 위치와 구조 등을 숙지하고 있을 필요가 없으며, 데이터의 표준화, 보안(security) 및 데이터 무결성(integrity) 유지가 가능하다^[1].

* TTA IT시험연구소 SW시험인증센터 (cschung@tta.or.kr), ** 충북대학교 컴퓨터공학과 (soonkey@cbucc.chungbuk.ac.kr)
논문번호 : 010270-1008, 접수일자 : 2001년 10월 8일

시스템 개발자는 DBMS를 사용하기 전에 일반적으로 시스템의 규모와 형상(configuration)을 파악하고, 이에 따른 데이터 내용이나 이용상태를 예측할 수 있기 때문에 데이터베이스 스키마 정의나 데이터베이스 구축 등에 관련된 일부 온-라인 작업들을 오프-라인(off-line) 작업으로 전환, 수행할 수 있다²⁾.

교환 시스템용 DBMS를 이용하여 오프-라인 상에서 데이터베이스의 설계와 구축 과정을 수행하며, 모든 응용 소프트웨어로부터의 데이터베이스 접근을 관리할 수 있다. 그러므로 오프-라인 상에서 데이터베이스의 효율적 설계와 오류 없는 정보의 구축 과정은 DBMS뿐만 아니라 교환 시스템의 안정성과 신뢰성에 직결되는 문제이므로 데이터베이스의 설계에서 구축까지의 각 단계별 작업과정에서 자동화 도구(CASE tools)의 사용은 필수적이다.

교환 시스템용 DBMS는 데이터베이스 스키마 정의, 데이터 등록 및 생성 등의 각 단계에서 유용한 자동화 도구를 제공하고 있다. 그러나 위와 같은 개발과정에 앞서 수행되어야 할 데이터의 분석과 설계 과정을 소홀히 취급함으로써 체계적이고도, 종합적인 데이터 구축이 이루어지지 않아 데이터 일관성이 결여되는 경우가 많이 발생하고 있다. 이러한 문제들의 한 예로서, TDX-ISDN 교환 시스템의 기능 시험에서 국 데이터에 관련된 오류가 7%에 이르는 것으로 분석되었다³⁾.

교환 시스템에서 사용되는 국 데이터는 시스템 형상 데이터, 교환에 필요한 데이터, 운용보전에 관련된 데이터 등, 그 규모가 방대할 뿐만 아니라 시스템의 기능 및 설계방법에 따라 그 종류도 다양하다. 교환 시스템의 국 데이터는 해당 시스템의 설치장소에 종속되며, 국 데이터의 오류로 인해 파급되는 영향도 매우 크게 나타난다.

시스템의 개발단계에서 국 데이터의 오류 발생으로 인하여 패키지의 개발과 시험을 반복적으로 수행해야 하며, 개발이 완료된 시스템의 현장 적용 시 발생하는 오류는 막대한 시간과 비용을 초래하며, 시스템 성능에 대한 대외 신뢰도 문제를 발생시킨다.

따라서 국 데이터의 오류 발생을 사전에 방지하기 위하여 교환 시스템 개발 회사들은 국 데이터의 설계 기술 개발과 생산 환경 구축에 많은 투자를 하고 있으며, 현장 운용 요원들에게도 국 데이터의 운용 기술 교육을 강화하고 있다.

본 논문에서는 교환국의 설치장비 물량 및 시스템

형상에 따라 변경이 가능한 교환국 데이터를 생산할 수 있는 데이터 분석과 데이터 설계 방법을 제안한다. 제 2장에서는 교환 시스템에서 처리하는 데이터의 종류를 설명하고, 제3장에서는 초기 국 데이터를 생성하는 절차에 대해 설명한다. 제4장에서는 국 데이터의 분석 단계로서, 특성에 따라 데이터를 분류하며, 제5장에서는 현재 운영중인 교환 시스템에서 발생하는 국 데이터의 오류 유형을 분석한다. 제6장에서는 국 데이터의 설계 단계로서, 데이터베이스 스키마 설계 단계에서 E-R(Entity-Relationship) 모델을 기반으로 하는 릴레이션 관계도와 ODBS (Office DataBase Specification) 에 대해 설명한다. 마지막 제 7장에서는 본 연구의 결과를 기술한다.

II. 교환 시스템의 데이터

교환 시스템의 국 데이터 종류는 시스템 형상과 관련하여 시스템 동작 중에도 변경되지 않는 영구 데이터, 가입자 정보 등과 같이 시스템 운용 중에 변경이 가능한 반영구 데이터, 그리고 교환 시스템의 상태 등에 관련되어 수시로 변경되는 임시 데이터들로 분류된다. 이러한 데이터들은 각 응용 소프트웨어에 따라 고유하거나, 가입자 데이터와 번호 번역 데이터들과 같이 교환국마다 상이한 데이터들로 구분되며, 교환 시스템의 효율성을 증대시키기 위하여 여러 프로세서들에 중복 또는 분산적으로 할당될 수 있다. 또한 이러한 데이터들은 오직 한 프로세서에만 존재할 수도 있으며, 이들의 사용 상태와 내용이 사용자에게 미리 알려진다는 특징을 가지고 있다⁴⁾.

다른 측면에서 교환 시스템의 데이터는 운용 데이터와 초기 국 데이터로 분류할 수 있다. 운용 데이터는 모든 교환국의 초기 값으로써 동일한 데이터 값을 가지며, 운용 중에 데이터 값이 변경되므로 보존되어야 하는 특성을 갖는다. 초기 국 데이터는 교환국별로 서로 다른 특성을 갖는 데이터로서, 통신 사업자가 생성한 국 데이터의 초기 동작에 이용되는 데이터이다.

교환 시스템의 신설 및 증설 시에 교환국 설계자는 운용 중에 발생하는 데이터의 초기 값과 가입자별 서비스 데이터, 중계선 데이터 및 시스템 형상 데이터 등 시스템의 초기 동작에 필요한 국 데이터를 입력해야 한다.

III. 국 데이터의 생성 절차

일반적으로 교환 시스템의 신규 설치 또는 응용 소프트웨어의 변경으로 인한 시스템의 재 가동 시에 교환국 데이터를 초기화해야 하며, 이는 교환 시스템의 운용 중에 발생하는 동적 데이터와 교환국마다 다르게 지정되는 가입자 서비스 데이터, 중계선별 신호방식에 따른 데이터 및 시스템 형상 데이터 등의 국 데이터를 시스템의 내부 데이터 구조에 따라 동작 형태로 생성하는 것을 의미한다⁵⁾.

국 데이터의 생성 절차는 그림 1과 같으며, 입력 파일의 작성 과정과 데이터베이스의 제작 도구를 사용하는 과정으로 구분된다. 입력 파일의 작성 과정은 ODBS 정의 및 생성 작업과 GRB (Generation Rule Base) 정의 및 코딩 작업으로 구분된다. ODBS는 교환 시스템의 구축 시 교환국 데이터 규격서에 명시된 가입자별 서비스 데이터, 중계선 데이터 및 시스템 파라미터 등 시스템의 초기 동작에 소요되는 입력 데이터 명세로서 교환국 설계자가 데이터베이스 규격서를 이용하여 용이하게 기술할 수 있다. GRB는 교환국별로 상이한 값을 가지며, 상호간에 복잡한 관계성에 따라 값이 결정되는 CAE(Customer Application Engineering) 데이터들을 처리하기 위하여 데이터에 대한 정보와 데이터간의 관계성을 명세한 알고리즘이다(알고리즘은 파일에 저장됨).

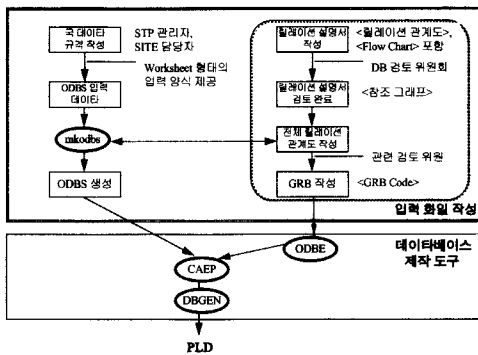


그림 1. 국 데이터의 생성 절차

데이터베이스 제작 도구는 ODBE(Office Data Base Engine), CAEP(Customer Application Engineering data Processor) 및 DBGEN(DataBase GENERator)으로 구성된다.

IV. 국 데이터의 분석

국 데이터의 분석 단계에서는 교환 시스템에서 사용하는 데이터를 그 타입(특징)에 따라 CDE (Customer Design Engineering), CAE 및 SYS (SYStem) 데이터로 구분하고, 각각의 데이터를 분석한다. CDE 데이터는 그림 2와 같이 교환국별로 동일한 값을 갖으며, 운용 중에도 초기 데이터는 변경되지 않는 특정한 데이터 타입을 갖는다. 또한 운용 중에는 변경될 수 있으나, 시스템의 버전 갱신이나 시스템 증설 시에는 반영되지 않는 데이터 타입을 포함하고 있다.

```

SYNMODE R_CMD_INF=RELATION(
  D_CMD_CRN          int(0:9999).bin,
  D_CMD_CLS          int(-128:127),
  D_CMD_GRP          bit(8),
  D_RMND_READ        bool,
  D_LOG_READ         bool,
  D_TERM_DEV_TYPE    int(-128:127),
  D_ForMincBatch     bool,
  D_SysType          int(-128:127)
);

DCL F_100 ARRAY(1:1000) R_CMD_INF;

F_100 := (
  300, 1, B'10000000, TRUE, FALSE, 3, FALSE, 6]
  401, 3, B'10000000, FALSE, FALSE, 3, FALSE, 6]
  402, 3, B'10000000, FALSE, FALSE, 3, FALSE, 6]
  403, 1, B'10000000, TRUE, TRUE, 3, TRUE, 6]
  404, 3, B'10000000, FALSE, FALSE, 3, FALSE, 6]
);
    
```

그림 2 CDE 데이터의 사례

CAE 데이터는 그림 3과 같이 교환국별로 시스템의 상태에 따라 수시로 변경되는 데이터로서, 변경된 내용을 시스템의 버전 갱신이나 증설시 반영시킬 수 있는 데이터 타입을 소유하며, ODBS를 입력으로 받는다.

```

SYNMODE M_SIGPROTO_TYPE =
  SET (q2931,q2971,uni31,uni40,pnni,bisup,etc sig);
SYNMODE M_CodeSts = SET (empty,next,lasf,changed);

SYNMODE R_TDN_TBL = RELATION(
  D_TDN_PTR          int(0:1110):index(10),
  D_TDN_DGT          int(0:9):index,
  D_TDN_STS          M_CodeSts,
  D_BEFORE_TDN_PTR  int(-1:1110),
  D_NEXT_TDN_PTR    int(-1:1110),
  D_NEXT_TDN_STS    bit(10),
  D_ALS_NO           int(0:61),
  D_IM_NO            int(0:30),
  D_LINK_NO          int(0:71),
  D_SIGPROTO_TYPE    M_SIGPROTO_TYPE,
  D_REG_MACT         bit(32),
  D_ILMI_DEF         bool,
  D_ILMI_REG         bool,
  D_REG_MAC2         bit(16),
  D_TDN_INDX         int(-1:511)
);

DCL F_79 ARRAY(1:1110) R_TDN_TBL;

F_79 := ((0, 0, empty, -1, -1, B'0000000000, 0, 0, 0, etc sig,
  B'00000000000000000000000000000000, false,
  false, B'0000000000000000, -1)
);
    
```

그림 3 CAE 데이터의 사례

SYS 데이터는 그림 4와 같이 ODBS를 입력으로 받지 않으나 운용 중에는 변경이 가능하며, 시스템의 버전 갱신 또는 증설 시에 반영되는 데이터 타입을 갖는다.

```

SYNMODE M PORT TYPE =
SET (DEBUG,RESERV,DUAL,ALARM,NSYNC);

SYNMODE R_PORT_INF = RELATION (
D_PORT_NO      int(0:4):index,
D_PORT_TYPE    M_PORT_TYPE,
D_SERVICE      bool
);

DCL F_105 ARRAY(1:5) R_PORT_INF;

F_105("C00") := (
[ 0, DEBUG, FALSE ]
[ 1, DUAL, FALSE ]
[ 2, RESERV, FALSE ]
[ 3, ALARM, FALSE ]
);
    
```

그림 4 SYS 데이터의 사례

CAE 데이터는 CDE나 SYS 데이터와는 달리 수시로 변경되는 임시 데이터로서, 데이터 상호간에 관계성이 복잡할 뿐만 아니라 교환 시스템의 데이터 중에서 90% 이상을 차지한다^[4]. 시스템의 시험 버전에서 CDE:CAE:SYS의 데이터 비율은 1:20:2이다.

V. 국 데이터의 분석

데이터의 설계, 생성 등 국 데이터에 영향을 주는 모든 요소를 고려해서 운영중인 교환 시스템에서 발생할 수 있는 국 데이터의 오류를 분류하면 다음과 같다.

1) 데이터 값의 오류

relation.dg 값이 부정확하거나 누락된 경우로, 데이터 구조의 변경이나 STP(System Test Plant) 형상의 변경 등 개발 단계에서 많이 발생하는 오류이다. 그러므로 데이터 구조의 변경이나 STP 형상의 변경 시 관련된 데이터도 함께 갱신되어야 한다.

2) 데이터 특성 분류상의 오류

SYS 데이터는 시스템 증설시 CAE 데이터의 특성을 갖게 되므로, 초기 국 데이터의 생성과 증설시 SYS 데이터는 생성 환경으로 분류되어야 한다. 예를 들면, SYS 데이터 특성을 갖는 ABD_POOL 등은 시스템 증설시 시스템 운용자가 운용상태 정보를 저장하였다가 증설 분 PLD(Processor Load Data)를 설치하고, 운용상태 정보를 다시 입력하여야 한다. 시험용 데이터의 경우도 이와 비슷하다.

3) 데이터 갱신상의 복잡성으로 인한 오류

국 데이터의 증설에는 많은 기간이 요구된다.

PLD의 백업, PLD의 증설 분 생성, 통신사업자의 검증 및 현장 적용에 이르기까지는 약 15일이 소요된다. 단순한 보드 증설이나, 가입자 프로세서의 추가 시는 MMC(Man Machine Command)의 사용이나 PLD의 부분 적용이 가능해야 하며, 유지 보수용 프로세서와 번호 번역용 프로세서의 데이터도 자동으로 갱신되어야 한다.

4) 과도한 양의 ODBS로 인한 오류

정보에 관련된 ODBS의 양은 매우 방대하다. 이는 랙(Rack)의 형상이 다양하고, 증계선의 경우는 T1, E1급등이 혼용되고 있기 때문에 발생한다. 따라서 시스템의 설계 차원에서 랙(Rack)의 표준화 등이 선행되어야 한다. 현재 국 데이터의 생성 환경에서 입력으로 사용되는 ODBS의 경우는 그 양이 방대하며, ODBS의 생성시도 전화국 규격, 하드웨어 규격 및 PLD 규격 등을 따라야 하므로 수작업으로 인한 오류의 발생 확률이 높다.

5) ODBS 내용의 중복으로 인한 오류

ODBS의 설계가 완벽하지 못하면 내용들이 상호 중복되며, 불일치 되는 경우가 발생한다. 이러한 문제는 테이블의 애트리뷰트 관점에서 해결되어야 한다. 예를 들면, ASP_INFO 만을 이용하여 Sub_hway 값들이 자동 생성 되도록 하여야 한다.

운용 데이터의 유형에 따라 발생된 오류의 종류를 분석한 결과, 3가지 데이터 타입(CDE, CAE, SYS) 중에서 대부분이 CAE 데이터에서 오류가 발생하였다. 예를 들면, 테이블간의 상호 관계성을 정확히 파악하지 못하여 ODBS 데이터 집합을 테이블에 정확히 전달하지 못하는 경우와 CAE 데이터를 잘못 분류하여 시스템 운용 중에 추출하는 백업 데이터베이스의 데이터 집합을 정확히 생성하지 못하는 경우이다. 이러한 오류들의 발생 비율은 TDX-ISDN 교환 시스템에서 발생된 전체 기능오류의 7%에 해당되는 것으로 분석되었다^[5]. 그러므로 국 데이터의 설계 시 테이블간의 릴레이션 관계도와 ODBS설명서를 작성하는 업무는 오류를 사전에 방지하기 위해 반드시 수행되어야 한다.

VI. 국 데이터의 설계

국 데이터의 설계는 ERD(Entity Relationship Diagram)을 기반으로 하는 릴레이션 관계도와 ODBS 설명서를 작성하는 단계로서 데이터베이스의 스키마 설계에 해당된다.

릴레이션 관계도는 테이블과 테이블 상호간의 관계성을 표현하며, 교환 시스템 개발과정에서 도출된 데이터베이스 설명서, 기술 문서(technical document) 등을 참고로 하여 작성된다.

릴레이션 관계도는 다음과 같이 3 종류로 구분할 수 있다.

- 1) 전체 릴레이션 관계도
 - 모든 테이블 상호간의 관계성을 정의한다.
- 2) 부분 릴레이션 관계도
 - 다른 테이블의 부분적인 애트리뷰트들로 생성되는 릴레이션 관계도를 의미한다.
- 3) 참조 릴레이션 관계도
 - 테이블 상호간에 내용을 참조하기 위해 특정한 애트리뷰트를 통하여 관계성을 구성하는 릴레이션 관계도를 의미한다.

전체 릴레이션 관계도는 그림 5와 같이 교환 시스템에서 사용되는 모든 테이블들로 구성되며, 테이블 이름 대신에 테이블 식별자(ID)를 사용한다.

테이블 ID는 테이블의 고유번호로서 테이블의 기능에 따라 다음과 같이 할당된다.

- ◆ 000 ~ 199: 호처리 기능
- ◆ 200 ~ 299: 운영.보전 기능
- ◆ 300 ~ 319: 가입자/중계선 정합 기능
- ◆ 320 ~ 355: 운영체제 기능

테이블 상호간의 관계성은 화살표로 표시하며, 화살표상에 테이블간의 관계성을 나타내는 의미 있는 식별자를 부여한다. 예를 들면, E164_01은 번호번역 기능 E.164에 관계된다는 것을 의미한다.

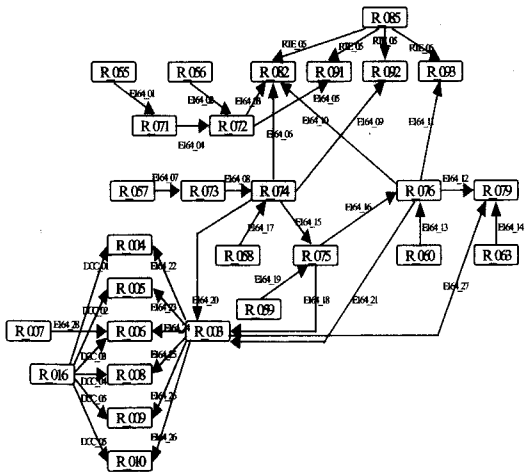


그림 5 전체 릴레이션 관계도

전체 릴레이션 관계도를 이용하여 테이블 상호간의 관계를 쉽게 파악할 수 있기 때문에 국 데이터 값이 정확히 할당되었는지 여부를 분석할 수 있다.

부분 릴레이션 관계도는 그림 6과 같이 테이블간의 상이한 애트리뷰트들로 구성된다. 예를 들면, R_TC_INF(R_074) 테이블상의 애트리뷰트 D_TC_INF_PTR는 R_TCINF_BFM 테이블상의 애트리뷰트 D_TCINFBFM_PTR, D_TCINFBFM_STS와의 부분 릴레이션을 표현한다.

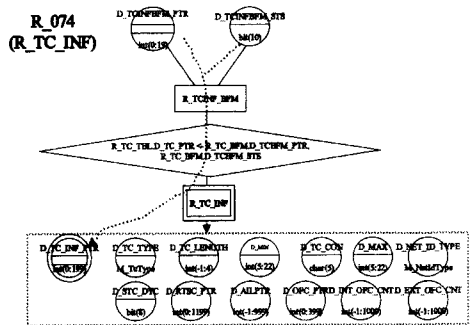


그림 6 부분 릴레이션 관계도

참조 릴레이션 관계도는 그림 7과 같이 이미 생성된 테이블의 특정 애트리뷰트와 동일한 애트리뷰트를 갖는 다른 테이블의 참조용으로 사용된다. 예를 들면, R_TC_INF(R_074) 테이블은 자신의 애트리뷰트 D_TC_INF_PTR와 R_TC_TBL 테이블에 있는 동일한 애트리뷰트 D_TC_INF_PTR의 매칭(matching)을 통해 R_TC_TBL 테이블을 참조한다.

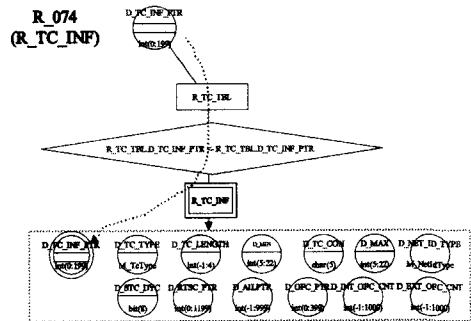


그림 7 참조 릴레이션 관계도

ODBS 설명서 작성 시 전체 릴레이션 관계도, 부분 릴레이션 관계도 및 참조 릴레이션 관계도를 반드시 참조해야 한다. ODBS 설명서는 교환 시스템의 시스템 형상을 참고로 하여 데이터 세트의 기술과 데이터 세트에 관련된 테이블에 데이터의 할당

시 이용된다.

ODBS 설명서는 다음과 같은 요소들로 구성된다.

- 1) 총괄 ODBS 데이터 세트
- 2) ODBS 데이터 세트-애트리뷰트 관계도

ODBS 데이터 세트의 입력 데이터에 관련된 테이블의 애트리뷰트에 데이터를 정확히 할당하기 위하여 관계도를 작성한다.
- 3) GRB 흐름 설명서

테이블의 애트리뷰트에 데이터를 할당하는 방법을 기술한다.
- 4) GRB 흐름도

ODBS 데이터 입력과 시간 흐름순서에 따른 초기 데이터의 할당을 표현한다.

그림 8과 같은 총괄 ODBS 데이터 세트를 통해 소요되는 데이터 세트를 한 눈에 파악할 수 있다. 총괄 ODBS 데이터 세트는 상호 종속적인 테이블들로 구성되며, GRB 프로그램의 기본 단위가 된다. 예를 들면, 데이터 세트 R_PFX_INF는 테이블 R_CONF_INT_ADDR(R_153), R_CONF_EXT_ADDR(R_154) 및 R_CONF_TN_INF(R_155)로 구성된다.

NODE_INFO R_150(R_NODE_INF)	R_PFX_INF R_153(R_CONF_INT_ADDR) R_154(R_CONF_EXT_ADDR) R_155(R_CONF_TN_INF)	PORT_INFO R_156(R_CONF_LINK_INF) R_157(R_CONF_RAIG_INF)
---------------------------------------	--	--

그림 8 총괄 ODBS 데이터 세트

그림 9와 같은 ODBS 데이터 세트-애트리뷰트 관계도는 ODBS 입력과 해당 테이블의 애트리뷰트를 관련지어 작성한다. 예를 들면, 데이터 세트 R_PFX_INF의 입력 데이터 R_PFX는 테이블 R_CONF_INT_ADDR(R_153)과 R_CONF_EXT_ADDR(R_154)의 애트리뷰트 D_REACH_PFX와 관련된다.

R_PFX_INF	mandatory	OPTION																											
<table border="1"> <tr> <th>OUT_RAIG</th> <th>IN_RAIG</th> <th>PORT_ID</th> <th>TYPE</th> <th>R_PFX</th> </tr> <tr> <td>INF</td> <td>INF</td> <td>INF</td> <td>INF</td> <td>R_PFX</td> </tr> <tr> <td>D_OUT_RAIG_INF</td> <td>D_IN_RAIG_INF</td> <td>D_PORT_ID</td> <td></td> <td>D_REACH_PFX</td> </tr> <tr> <td>INF</td> <td>INF</td> <td>INF</td> <td></td> <td>CHANGED</td> </tr> </table>	OUT_RAIG	IN_RAIG	PORT_ID	TYPE	R_PFX	INF	INF	INF	INF	R_PFX	D_OUT_RAIG_INF	D_IN_RAIG_INF	D_PORT_ID		D_REACH_PFX	INF	INF	INF		CHANGED	<p>ODBS INPUT</p> <p>DATABASE TYPE</p> <p>ATTRIBUTE</p> <p>ATTRIBUTE TYPE</p>								
OUT_RAIG	IN_RAIG	PORT_ID	TYPE	R_PFX																									
INF	INF	INF	INF	R_PFX																									
D_OUT_RAIG_INF	D_IN_RAIG_INF	D_PORT_ID		D_REACH_PFX																									
INF	INF	INF		CHANGED																									
<table border="1"> <tr> <th>TN_LENQ</th> <th>NDOTOKEN</th> <th>NET_ID</th> <th>SCOPE</th> <th>VP_FLAG</th> </tr> <tr> <td>INF</td> <td>INF</td> <td>INF</td> <td>INF</td> <td>INF</td> </tr> <tr> <td>D_TN_LENQ</td> <td>D_NDOTOKEN</td> <td>D_NET_ID</td> <td>D_SCOPE_ADV</td> <td>D_VP_FLAG</td> </tr> <tr> <td>INF @100000</td> <td>INF @120</td> <td>CHANGED</td> <td>INF</td> <td>INF</td> </tr> </table>	TN_LENQ	NDOTOKEN	NET_ID	SCOPE	VP_FLAG	INF	INF	INF	INF	INF	D_TN_LENQ	D_NDOTOKEN	D_NET_ID	D_SCOPE_ADV	D_VP_FLAG	INF @100000	INF @120	CHANGED	INF	INF									
TN_LENQ	NDOTOKEN	NET_ID	SCOPE	VP_FLAG																									
INF	INF	INF	INF	INF																									
D_TN_LENQ	D_NDOTOKEN	D_NET_ID	D_SCOPE_ADV	D_VP_FLAG																									
INF @100000	INF @120	CHANGED	INF	INF																									
<table border="1"> <tr> <th>R_CONF_INT_ADDR</th> <th>R_CONF_EXT_ADDR</th> <th>R_CONF_TN_INF</th> </tr> <tr> <td>D_REACH_PFX</td> <td>D_REACH_PFX</td> <td>D_NET_ID</td> </tr> <tr> <td>D_REACH_PFX_LENQ</td> <td>D_REACH_PFX_LENQ</td> <td>D_TN_INF</td> </tr> <tr> <td>D_C_INF_ADDR_INF</td> <td>D_C_INF_ADDR_INF</td> <td>D_INF_ADDR_INF</td> </tr> <tr> <td>D_INF_ADDR_INF</td> <td>D_INF_ADDR_INF</td> <td>D_INF_ADDR_INF</td> </tr> <tr> <td>D_OUT_RAIG_INF</td> <td>D_OUT_RAIG_INF</td> <td>D_INF_ADDR_INF</td> </tr> <tr> <td>D_IN_RAIG_INF</td> <td>D_IN_RAIG_INF</td> <td>D_INF_ADDR_INF</td> </tr> <tr> <td>D_VP_FLAG</td> <td>D_VP_FLAG</td> <td>D_INF_ADDR_INF</td> </tr> <tr> <td>D_SCOPE_ADV</td> <td>D_SCOPE_ADV</td> <td>D_INF_ADDR_INF</td> </tr> </table>	R_CONF_INT_ADDR	R_CONF_EXT_ADDR	R_CONF_TN_INF	D_REACH_PFX	D_REACH_PFX	D_NET_ID	D_REACH_PFX_LENQ	D_REACH_PFX_LENQ	D_TN_INF	D_C_INF_ADDR_INF	D_C_INF_ADDR_INF	D_INF_ADDR_INF	D_INF_ADDR_INF	D_INF_ADDR_INF	D_INF_ADDR_INF	D_OUT_RAIG_INF	D_OUT_RAIG_INF	D_INF_ADDR_INF	D_IN_RAIG_INF	D_IN_RAIG_INF	D_INF_ADDR_INF	D_VP_FLAG	D_VP_FLAG	D_INF_ADDR_INF	D_SCOPE_ADV	D_SCOPE_ADV	D_INF_ADDR_INF		
R_CONF_INT_ADDR	R_CONF_EXT_ADDR	R_CONF_TN_INF																											
D_REACH_PFX	D_REACH_PFX	D_NET_ID																											
D_REACH_PFX_LENQ	D_REACH_PFX_LENQ	D_TN_INF																											
D_C_INF_ADDR_INF	D_C_INF_ADDR_INF	D_INF_ADDR_INF																											
D_INF_ADDR_INF	D_INF_ADDR_INF	D_INF_ADDR_INF																											
D_OUT_RAIG_INF	D_OUT_RAIG_INF	D_INF_ADDR_INF																											
D_IN_RAIG_INF	D_IN_RAIG_INF	D_INF_ADDR_INF																											
D_VP_FLAG	D_VP_FLAG	D_INF_ADDR_INF																											
D_SCOPE_ADV	D_SCOPE_ADV	D_INF_ADDR_INF																											

그림 9 ODBS 데이터 세트-애트리뷰트 관계도

그림 10과 같은 GRB 흐름 설명서는 ODBS 데이터 세트의 입력 데이터에 관련된 테이블의 애트리뷰트에 데이터를 정확히 할당하는 방법을 기술한 설명서로서, ODBS 입력과 해당 테이블의 애트리뷰트를 관련지어 작성한다. 특히, 오류 처리에 필요한 설명서를 첨부하여야 하며, 이는 GRB 프로그램의 예외처리 루틴을 사용한다.

```

R_PFX_INF DATASET 설명서

1. R_PFX : internal, external Reachable Address를 표시 한다. 최대 40char까지 구성 가능 하나
   처음 0 - 15 char만 사용된다. R_CONF_INT_ADDR.D_REACH_PFX, R_CONF_EXT_ADDR.D_REACH_PFX값과
   D_REACH_PFX_LENQ를 표시. 하나의 Reachable Address가 생성 될때 D_C_INF_ADDR_SEQ
   값을 연이따 생성 한다(최대 127개).
2. TYPE : 0과 1만 사용 가능 0이면 Internal Reachable Address, 1이면 External Reachable Address
3. PORT_ID : PORT_ID는 PORT_INF에 기술 되어 있는 PORT_ID만 사용 가능 한다.
4. IN_RAIG : PORT_INF에서 생성된 R_CONF_RAIG_INF.D_RAIG_REF값
5. OUT_RAIG : PORT_INF에서 생성된 R_CONF_RAIG_INF.D_RAIG_REF값
6. VP_FLAG : 미정
7. SCOPE : Exterior Reachable Address의 scope Advertisement level을 의미 한다.
8. NET_ID : R_CONF_TN_INF.D_NET_ID값이며 D_TN_INF.D_TN_INF.D_NET_ID.D_TN_INF.D_NET_ID.D_TN_INF.D_NET_ID.D_TN_INF.D_NET_ID
9. NDOTOKEN : R_CONF_TN_INF.D_TN_INF.D_TN_INF.D_TN_INF.D_TN_INF.D_TN_INF.D_TN_INF.D_TN_INF.D_TN_INF.D_TN_INF.D_TN_INF.D_TN_INF
10. TN_LENQ : R_CONF_TN_INF.D_TN_INF.D_TN_INF.D_TN_INF.D_TN_INF.D_TN_INF.D_TN_INF.D_TN_INF.D_TN_INF.D_TN_INF.D_TN_INF.D_TN_INF
    
```

```

ERROR CODELIST

/* STEP1: ASSIGN INPUT VALUE */
/* START : assign input value to input variable */
/* 1.1 : missing R_PFX */
/* 1.2 : range over R_PFX */
/* 1.3 : missing TYPE */
/* 1.4 : different TYPE */

/* STEP2: ASSIGN R_PFX VALUE */
/* START : assign input value to relation */
/* 2.1 : Already equip internal */
/* 2.2 : Already equip exterior */
    
```

그림 10 GRB 흐름 설명서

그림 11과 같은 GRB 흐름도는 ODBS 데이터 입력과 초기 데이터를 시간 흐름순서에 따라 할당하는데 사용되며, GRB의 프로그래밍을 위해 흐름도(Flow Chart)를 사용한다.

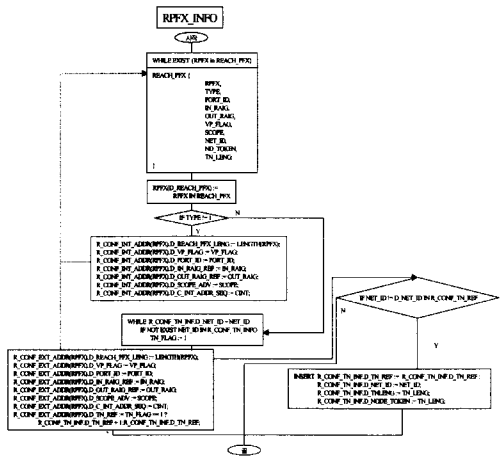


그림 11 GRB 흐름도

국 데이터 설계 단계에서 릴레이션 관계도와 ODBS 설명서를 충분히 활용한다면 GRB 프로그램에 잠재될 수 있는 오류를 사전에 제거할 수 있으며, 오류 발생률도 줄일 수 있다.

VII. 결 론

본 논문에서는 교환 시스템에서 교환국의 설치 물량과 시스템 형상에 따라 변경되는 교환국 데이터를 대상으로 국 데이터를 자동 생산할 수 있는 데이터 분석 및 설계 방법을 제안하였다.

특히, 본 논문에서는 국 데이터 분석과 오류 유형의 분석을 통해 국 데이터의 설계 단계에서 필요한 릴레이션 관계도의 작성 방법과 ODBS 설명서의 작성 방법을 제안하였다.

릴레이션 관계도는 전체 릴레이션 관계도, 부분 릴레이션 관계도 및 참조 릴레이션 관계도로 구분되며, 이러한 관계도를 ODBS 설명서 작성시 이용한다. ODBS 설명서는 총괄 ODBS 데이터 세트, ODBS 데이터 세트-에트리뷰트 관계도, GRB 흐름 설명서 및 GRB 흐름도로 구성되며, 이들을 GRB의 프로그래밍 시 이용한다.

TDX-ISDN 교환 시스템의 기능시험에서 7%에 해당되는 국 데이터에 관련된 오류가 발생하였으나^[3], 본 논문에서 제안한 릴레이션 관계도와 ODBS 설명서를 국 데이터 설계 단계에 적용함으로써 ODBS 데이터 세트가 테이블에 정확히 전달되지 않는 오류를 개선할 수 있었다. 또한 교환 시스템 운용 중에 추출하는 백업 데이터베이스의 데이터 세트도 정확히 생성할 수 있었다.

향후 연구과제로는, 릴레이션 관계도에는 정의되어 있으나 ODBS 데이터 세트에서는 사용되지 않는 데이터 항목들에 대한 검증 문제를 들 수 있다.

참 고 문 헌

- [1] 한국전자통신연구소, "TDX-10 개요", TDX-10 총서 제 1권, pp. 96-97, 1994.1.
- [2] 박유미, 윤용익, 조주현, "교환기 시스템에서의 데이터 자동 생성을 위한 모델링 방법에 관한 연구", JCCI'96 학술대회 논문집, pp.704-707, 1996.4.
- [3] 정창신, 이형호, 양재동, "교환시스템의 국 데이터 자동 생성을 위한 모델링 도구의 구현", JCCI'97 학술대회 논문집, pp.1102-1106, 1997.4.
- [4] 박유미, 이길행, 우왕돈, 조주현, "주기억 장치 상주 분산 실시간 DBMS를 위한 데이터베이스

제작도구의 구현", 대한전자공학회 추계학술발표대회 논문집, Vol.27, No2, pp.510-513, Nov. 1994.

- [5] 한국전자통신연구소, "TDX-10 개발지침서", TDX-10 총서 제 10권, pp270-271, 1994.1.

정 창 신(Chang-Shin Chung)

정회원



1983년 2월 : 홍익대학교 전자계산학과 졸업

1987년 8월 : 홍익대학교 전자계산학과 석사

1984년 3월 ~1992년 2월 : ETRI 연구원

1992년 3월 ~2000년 7월 :

ETRI 신입연구원

2000년 8월 ~2000년 11월: ETRI 책임연구원

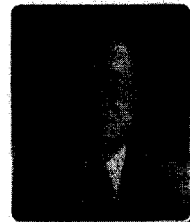
2000년 12월~2001년 11월: ETRI S/W품질인증팀장

2001년 12월~현재 : TTA IT시험연구소 S/W시험인증센터 평가2팀장/책임연구원

<주관심분야> S/W제품품질평가, S/W프로세스개선, 통신S/W, DBMS기술

정 순 기(Soon-Key Jung)

정회원



1982년 8월 : Uni. of Dortmund, Informatik, Dipl. Inform 취득

1994년 2월 : Uni. of Groningen, Computing Science, Dr. 취득

1985년 5월~현재 : 충북대학교 컴퓨터공학과 교수

1994년 8월 : 충북대학교 전자

계산소장

1998년 11월 : 한국과학재단 한독기초과학협력위원회 정보분과위원장

2000년 4월 : 충북대학교 도서관장

<주관심분야> 데이터베이스 시스템, 소프트웨어공학, 소프트웨어실시간 시스템