

Office Data Generator

曹周鉉 · 安善熙 · 申在植 · 金庸漢

〈要 約〉

Data generator는 TDX 시스템을 신규로 설치할 때 가입자, 중계선 및 시스템 데이터와 국별로 정의해 주어야 하는 데이터를 일반 운용자가 알기 쉬운 형태로 입력하도록, 하고 이 원시 데이터를 TDX의 내부 자료구조에 맞는 형태의 초기 데이터 베이스로 만들어 주는 소프트웨어 생산 tool의 일종이다. 국 데이터는 그 양이 방대하여 시스템이 신규로 설치될 때마다 생산되어야 하므로 이국 데이터 생성 작업의 자동화는 TDX 상용화에 필수 조건이 된다. TDX의 data generator는 국 데이터 생성 작업의 전체 과정중에 기본 기능과 TDX 내부 구조와 밀접한 관련이 있는 부분에 대하여 우선적으로 portability가 높고 효율적인 구조로 개발함으로써 user-friendly interface 등 기능의 계속적인 추가 및 변경이 쉬우며 UNIX machine이면 거의 변경 없이 수행될 수 있도록 하였다.

I. 서 론

일반적으로 SPC 방식의 전자 교환기가 전 화국에 설치될 때 각 국마다 고유한 동작 환경을 정의해 주어야 한다. 이러한 동작환경에는 전화번호 체계, 국간 route 정보, 중계선별 신호방식, 가입자별 서비스 종류, 그리고 설치국에 맞는 system configuration에 관한 정보등이 있다. 이렇게 신규로 설치 혹은 증설하는 시스템마다 정의해 주어야 하는 고유한 정보를 국 데이터라 부르며 그 양은 설치용량에 비례하여 막대하다.

이 데이터의 생성 및 관리는 소프트웨어 생산이라는 측면에서 하드웨어 제작과 더불어 시스템 생산의 주요 부분을 차지하게 된다. 종전에는 TDX 국 데이터 생성을 각 시스템 프로그램(Generic program이라고도 하며 generic은 support tool과 대응하여 시스템 본체의 뜻으로 사용함)의 개발자들이 프로그래밍 언어의 데이터 선언문을 이용하여

TDX 내부 자료구조에 맞게 수작업으로 만들어 왔었다. 85년부터 TDX가 전국적으로 확산 설치됨에 따라 생산 및 운용 업무의 전문화 및 인력절감, 그리고 데이터 입력자의 오류 방지등의 면에서 국 데이터 생성을 효율화하지 않고는 TDX의 실용화는 불가능해졌다. 종전의 국 데이터 생성 방법으로는 다음과 같은 문제점이 있다.

- 국 데이터 생성을 위해 TDX 내부 자료구조를 알아야 하므로 시스템 개발자와 생산 및 운용자의 분리가 안된다. 이는 막대한 개발 인력의 손실을 가져온다.
- 분산 구조의 TDX에서 한 프로세서 내 혹은 다수의 프로세서에 걸쳐 상호 dependency가 있는 자료의 경우에는 data inconsistency를 유발하기 쉽다.
- 여러 기능에 대응하여 조합한 binary code를 직접 입력하므로 어렵고 오류가 발생하기 쉽다.

TDX의 data generator는 국 데이터 생산을 위해 시스템 수요자(한국전기통신공사: KTA)의 국 설계 기준에 의거하여 국 데이터 생산자(dg 사용자 혹은 operator라고도 함)가 알기 쉬운 입력 형식에 따라 symbolic하게 국 환경을 정의해 주도록 한다. 그리고 입력된 이 원시 데이터로부터 TDX의 분산된 각 프로세서마다 내부동작 형태의 데이터 베이스를 만들어 준다.

외국에서 도입하여 운용중인 시스템을 살펴보면 M10CN에서는 Software Production System이라 하여 국 설계 기준에 따라 software configuration 및 데이터 베이스를 정의하는 customer application engineering부분시간 backup system인 drum의 이미지체로 제작하는 software production의 분으로 구성되어 있다. 이 Software Production System은 대형 컴퓨터를 이용한 SD(Software Development & Support System)의 일부분으로 제공되어 있다.^{[1][2]} No.ESS에서는 국내 시스템 설치업체가

전화국의 요구사항을 분석하여 일정한 형식에 따라 정리하는 data collection, 그리고 국 데이터 생산을 위해 별도의 컴퓨터 시스템이 설치되어있는 외국 데이터 센터에서 수행하는 data generation의 두 단계로 이루어졌었다.^[3]

일반적으로 국 데이터를 포함한 소프트웨어 시스템을 생산하기 위해서 교환기 개발자가 제공하는 소프트웨어 패키지를 포함한 별도의 컴퓨터 시스템이 필요하다. 그리고 국 데이터 생성용 software tool은 교환기 시스템 소프트웨어의 내부 자료구조와 밀접한 연관이 있다. 따라서 외국에서 도입하는 전자교환기의 경우에는 그 software tool의 기술이전 문제와 낮은 portability 때문에 기존의 컴퓨터 시스템을 이용하기 어렵다.

TDX의 software 생산은 software generation, office administration, data generation, 그리고 system backup building의 4 단계로 나누어진다. TDX data generator는 data generation을 자동화 하기 위한 tool로서 이는 다시 user interface 부분과 기본기능 부분으로 분리하여 이 기본 기능에 대하여 우선적으로 portability가 높고 관리하기 쉬운 구조로 만들어 줌으로써 user-friendly interface 기능의 점진적 개선이 용이하도록 하고, TDX 내부 자료구조의 변경에도 쉽게 적용할 수 있도록 하였다. 또한 UNIX machine 이거나 표준 C language가 설치되어 있는 다른 OS에도 쉽게 수행될 수 있도록 하였다.

II. 시스템 생산 과정과 Data Generation

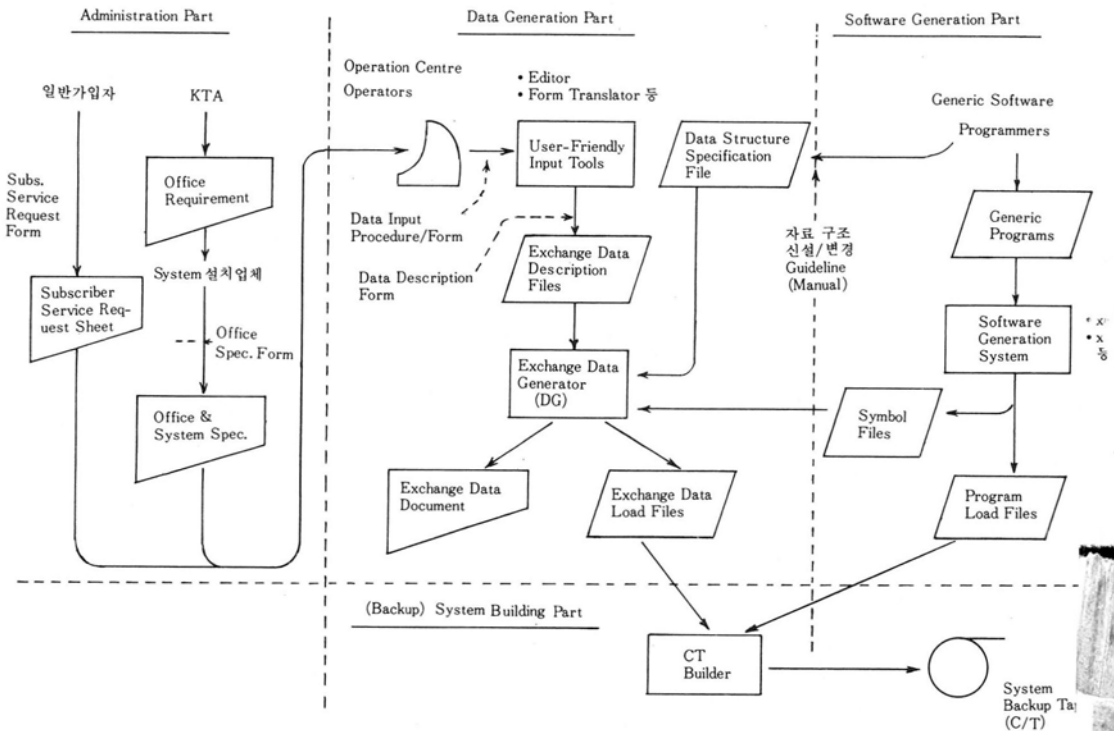
새로운 TDX 시스템을 생산하는 과정을 국 데이터 생성의 면에서 보면 <그림 1>과 같이 국 설치기준을 정하는 system administration, 그 기준을 컴퓨터에 입력하여 교환기 내부 데이터 베이스를 만드는 data generation, 교환기 시스템 소프트웨어를 주어진 co-

nfiguration에 맞게 컴파일링 해주는 software generation, 그리고 만들어진 국 데이터와 시스템 소프트웨어를 교환기에서 load 되도록 backup system (Tape) 을 만들어 주는 backup system building의 4 단계로 이루어진다. 이중 software generation 단계는 system software에 변경이 있을 경우에만 수행되고 나머지 3 단계는 시스템 생산시 마다 수행된다. Data generator는 system administration 단계에서 정리된 시스템 사용자의 요구사항을 컴퓨터에 일정한 형식대로 입력하도록 돕고 software generation 부분에서 제공되는 시스템 내부 자료구조 규격, symbol address 정보 등을 토대로 교환기 내부 데이터 베이스를 만든다.

III. Data Generator의 구조

1. Data Generator의 구조적 특징

Data generator는 교환기 소프트웨어 생산 tool의 일종이므로 신규 시스템을 생산할 때마다 사용되어야 하고 정의해 주어야 할 국 데이터양이 방대하므로 고도의 user-friendly interface 기능이 필요하다. 더우기 교환기 내부 데이터 베이스를 만들어 주어야 하므로 교환기 내부 자료구조에 매우 밀접한 관계가 있는데 교환기 소프트웨어의 내부 변경에 따라 그 생산 tool까지 변경되어야 하면 시스템 소프트웨어와 tool의 분리가 되지 않아 그 관리가 매우 복잡해진다.



〈그림 1〉 System 생산과정과 Data Generation

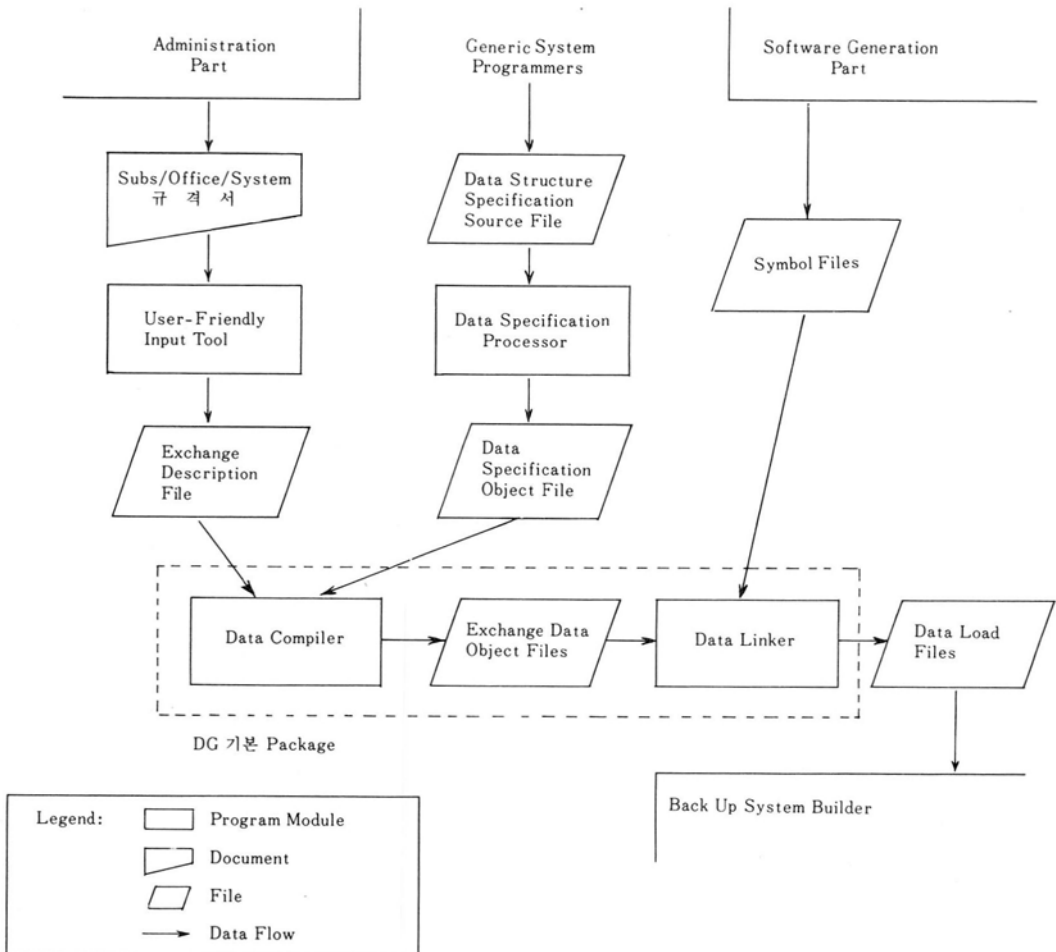
TDX의 data generator는 전체 data generation 기능을 서로 독립한 몇개의 모듈로 분리하고 이들 중 기본 기능에 해당되는 부분만을 우선적으로 개발하였고 그외의 것들은 독립적으로 그리고 점차 추가 개발을 하기 쉽도록 설계하였다. 즉, 두가지 front-end processor로서 user-interface tool과 data structure specification processor를 독립시켜 분리해냄으로써 입력 형식의 변경뿐 아니라 TDX 내부 data structure의 변경에도 data generator의 주요 모듈들에는 영향이 거의 없도록 하였다.

2. Data Generator 구조-Application Independent Architecture

Data generator는 크게 4개의 기능적 모듈, 즉 user-interface, data structure specification compiler, data compiler 및 data linker로 이루어진다.

가. User-interface (Input tool)

국 데이터를 오퍼레이터가 입력할 수 있도록 돕는 tool로서 menu-driven 혹은 window 등 오퍼레이터에게 입력의 편리를 돕는 각종 입력방식을 제공하며 출력은 국 data des-



〈그림 2〉 Data Generator의 구성도

cription language로 작성된 원시 국 데이터이다. 현재는 기존의 UNIX editor 등을 사용하도록 한다.

나. Data structure specification compiler
Generic software 개발자에 의해 작성된 국 데이터와 관련된 TDX 내부 data structure에 관한 specification을 data compiler가 사용할 수 있는 테이블 형태로 만들어주는 pre-processor이다. 이것은 generic software가 변경될 때만 수행된다. 현재는 이 컴파일러는 개발되지 않았고 각 generic programmer들이 data structure specification language로 정의한 것을 직접 data generator내에 테이블로 만들도록 하였다.

다. Data compiler
국 data description language로 기술된 국 데이터를 입력하여 TDX의 data structure specification table를 참조하면서 각 프로세서별 data object file을 만들어 주는 것으로 기본 패키지의 front-end 부분이다.

라. Data linker
Data compiler에 의해 만들어진 프로세서별 data object file을 입력하여 generic software의 생산 과정을 통해 각 데이터의 address가 결정되어 기록된 symbol file을 참조하여 각 데이터의 address가 binding된 각종 data load file을 만든다. 이는 data generator 기본 package의 back-end 부분이다.

IV. 국 Data 기술 및 Data Structure 정의 방법

Data generator의 기본 패키지는 데이터 컴파일러 및 데이터 링커로 구성되고 국 data description language로 기술된 국 data source file과 data structure specification language로서 정의된 국 데이터와 관련된 TDX 내부 data structure를 입력으로 하여 국 데이터를 생성시킨다. Data structure

는 프로그래머들이 시스템 소프트웨어를 변경할 때마다 정의해 주고 국 데이터는 일반 오퍼레이터들이 시스템을 생산할 때마다 기술 된다.

1. 국 데이터의 종류 및 국 Data Description Language

국 데이터는 크게 가입자 데이터, 중계선 데이터 및 시스템 데이터의 3가지 데이터 등급으로 구분된다. 가입자 데이터는 가입자 번호 배정, 가입자 line의 전화기 종류 등 일반 특성과 특수 서비스 종류, PABX line 배정 등에 관한 정보이다. 중계선 데이터는 시내 국번 및 전국 지역 번에 관한 정보, 중계선 route 구성 및 종류등을 포함한다.^[4] 시스템 데이터는 자국번호, 신호기, 과금주기 및 switch network 구조 등에 관한 정보이다.^[5]

각 data class는 다시 위에서 열거한 세부 기능별로 다수의 data set으로 나누어져 data generator가 읽어 처리하는 기본 단위가 된다. Data set은 다수의 data item으로 구성되는 데 각 data item은 명칭이 부여되어 그 명칭에 원하는 값을 배정하는 형식으로 국 데이터가 기술된다. Data set내에서 key item이 있는 경우에는 key item이 먼저 기술되나 그외의 item은 순서에 관계없이 기술된다. 여기서 key item이란 한 data set내에서 다수의 data entry가 있을때 각 data entry를 대표하는 item이다. 예로 subscriber line class (SCL)라는 data set내에 다수의 가입자에 대한 각종 서비스에 해당하는 data item을 나열하게 되는 데 여기서 가입자 번호가 key item이 된다. 배정하는 값은 숫자를 바로 넣지 않고 data structure specification에서 정의되어 있는 symbol로도 입력할 수 있다.

2. Data Structure Specification Language

Data structure specification의 주 기능은 국 데이터의 입력시 기술되는 data item 명칭들과 이들의 값이 TDX에서 실제로 저장되는 위치 및 방법을 연관시켜 주는 것이다. 국 데이터와 관련되는 TDX 내부 data structure는 이 data structure specification language에 의해 정의된다. 여기에는 각 개별 data structure 뿐 아니라 용도에 따라 개별 데이터간에 존재하는 복잡한 data dependency 그리고 특이한 데이터 구조에 대한 데이터 액세스 알고리즘 등을 정의해 줄 수 있다.

TDX의 data structure는 대부분이 간단한 일차원적 array, structure (혹은 record), integer등의 형태이다. Data generator에서는 이러한 간단한 구조의 data structure를 표준 구조라 하여 그 세부 bit field를 국 데이터 입력시 기술되는 data item들과 연관시켜 주도록 하였다. 표준구조에 속하지 않는 multi-dimensional table, tree structure 등은 고유구조라 하여 각 generic software의 개발자가 그 data structure에 대한 데이터 액세스 및 생성 알고리즘을 정의해 주도록 하였다. 이렇게 함으로써 generic software가 변경될 때마다 그 support tool인 data generator를 매번 변경해야 하는 복잡함이 없어진다.

V. 보완 개발 사항

현재의 data generator는 데이터 컴파일러 및 데이터 링커 등 기본 기능에 중점을 두어 개발 되었고 그의 user-interface 기능은 기존의 범용 tool을 이용하거나 일부 기능은 테이블화 하여, 어떤 기능의 독립적 관리를 최대한 도모하여 추가 개발이 쉽도록 하였는데 이러한 기능들은 대부분 customer application engineering에 해당되므로 TDX 생산자가 필요에 따라 개발함이 더 바람직하기 때문이다.

추가로 개발 되어야 할 주요사항은 다음 3가지를 들수 있다.

1. User-friendly Interface의 고도화

현재 입력 형식은 모든 항목에 이름을 주어 그 이름에 값을 배정하는 방식이나 재래적으로 많이 쓰는 batch 처리용 카드형 입력 방식과 호환성을 갖도록 할 수 있다. 또한 데이터를 입력할 때 menu-driven 혹은 window-driven query 방식을 사용하여 기존에 입력된 값 혹은 default값을 interactive하게 참조(Display) 하면서 변경할 필요가 있는 항목만 변경할 수 있도록 하면 많은 양의 데이터 입력을 매우 편리하게 할 것이다.

2. Data Structure Specification Compiler

Data generator는 시스템 소프트웨어의 data structure(Generic data structure라 함)에 맞게 값을 만들어 주므로 이 data structure에의 의존도가 매우 높다. 또한 응용에 따라 여러 종류의 데이터는 복잡한 상호 dependency를 가질 수 있다. 국 데이터와 관련된 generic structure뿐만 아니라 이들간의 상호 관계도 generic programmer가 정의 해주면 이 data structure specification compiler가 DG 내부 처리용 관련 테이블들을 자동으로 만들어 데이터 컴파일러에게 넘겨준다. 이렇게 함으로서 generic software가 변경되어도 data generator의 변경이 필요없어 support tool인 data generator의 관리를 매우 쉽게 한다.

3. 운용 국 데이터의 Administration

설치하여 운용되면서 변경되어온 국 데이터를 생산 시스템으로 가져와서 확인 및 수정하거나, 대량의 국 증설 혹은 변경등의 경우에 기존 운용 데이터와 대량의 신규 데

이터와 병합하여 새로운 서비스에 들어갈 수 있도록 할 필요가 있다. 이는 만들어진 국 데이터의 확인 및 검증을 위한 보조 수단으로 사용될 수 있다. 이 기능을 위하여 generic system에서 국 데이터를 disk(혹은 CT)에서 MT로 옮길 수 있어야 한다.

VI. 결 론

전자 교환기를 신규로 설치할 때 만들어 주어야 하는 국 데이터는 그 양이 막대하여 TDX가 시험 단계를 지나 확산 설치됨에 따라 국 데이터의 효율적인 생산 방법이 무엇보다 필요하였다. TDX에서는 data generator를 개발함으로써 이러한 국 데이터를 TDX 구조를 모르는 일반 운용자가 쉽게 입력하여 만들수 있었으며, '85년도 초부터 본격적인 개발에 들어가 '85년 10월부터 4개 지역의 약 20개 시스템에 걸쳐 성공적으로 적용되어 막대한 인력 절감을 가져왔다.

그간 시스템의 지속적인 보완에 따라 시스템내 data structure의 변경 및 추가가 빈번히 발생해 왔으나 대부분의 경우에 data generator 내의 data structure specification table을 변경함으로써 쉽게 해결되었다. 특히 국내에 도입하여 운용중인 No. 1A ESS나 M10CN의 경우는 이러한 software tool의 기술적 이전문제, 다른 기종 컴퓨터에의 portability 문제, 혹은 그 방대한 작업의 국

내와 외국 본사에서서의 중복화 등 어려움이 따랐으나 TDX의 data generator는 국내에서 널리 사용되는 UNIX machine이거나 표준 C language가 있는 시스템이면 별도의 장치없이 쉽게 설치될 수 있게 하였다.

앞으로 user-friendly interface, TDX data structure 변경 처리의 자동화, 운용 국 데이터의 관리 등의 부분을 중심으로 보완해 나가면 TDX가 세계적으로도 뒤떨어지지 않는 수준이 되는데 중요한 역할을 하게 될 것이다.

〈參 考 文 獻〉

1. ITT/BTM, M10CN Software Production Procedure, 삼성반도체통신(주), 1984.
2. ITT/BTM, M10CN Semi-permanent Data Part 1 (SPD 1) Production, 삼성반도체통신(주), 1984.
3. Western Electric Co. (WECO), International Translation Guide-No. 1AESS, Vol. 1~Vol. 5, 금성반도체통신(주), July, 1983.
4. 강석열, 조주현 외, "전전자 교환기 개발 사업 중 CP-S/W 개발 과제 보고서", 한국전자통신연구소, 1985. 6.
5. 김영시 외, 전전자 교환기 개발 사업중 M&A-S/W 개발 과제 보고서, 한국전자통신연구소, 1985. 6.



曹 周 鉉 (Cho, Ju Hyun)
 1953년 7월12 일생
 1978 : 서울대학교 전자공학과 공학사
 1984 : 한국과학기술원 전산학과 석사
 1978~1986 : 한국전자통신연구소
 1986. 8. 현재 : 운영체계연구실 선임연구원



安 善 熙 (Ahn, Sun Hee)
 1960년 4월24일생
 1983 : 경북대학교 전자공학과 공학사
 1983. 3~1986 : 한국전자통신연구소
 1986. 8 현재 : 운영체계연구실 연구원



申在植 (Shin, Jae Sik)
1961년 5월 17일생
1984 : 경북대학교 전자공학과
공학사
1984. 3~1986 : 한국전자통신연구소
1986. 8 현재 : 운영체제연구실 연구원



金庸漢 (Kim, Yong Han)
1959년 5월 28일생
1982 : 서울대학교 제어계측공
학과 공학사
1984 : 서울대학교 대학원 제어
계측공학과 석사
1984. 3~1986 : 한국전자통신연구소
1986. 8. 현재 : 운영체제연구실 연구원