TMN을 위한 ASN.1/GDMO 통합 환경 설계 구현

김 영 철†

요 약

ASN.1/GDMO는 TMN(Telecommunication Management of Network)에서 망 관리에 이용되는 관리 객체이다. 그러나 ASN.1/GDMO는 망 관리를 위해 직접 이용되는 것이 아니라 객체지향 요소를 갖는 또 다른 언어로 변환되어 이용된다. 따라서 ASN.1/GDMO를 조작할 수 있는 개발 환경이 필요하다. 본 논문에서는 ASN.1/GDMO의 요소를 사용자 인터페이스(GUI)를 통해 관리할 수 있는 편집기와 브라우저를 개발하고, DB와 연동될 수 있는 통합환경을 개발하였다. 본 논문의 구현은 UNIX상에서 이루어졌으며, 컴파일러 보조도구인 FLEX와 BYACC을 이용하였다. 또한 DB로는 Objectivity DB를 이용하였으며 인터페이스 언어로 Tcl/Tk를 사용하였다. 본 논문의 실험에서는 구현된 ASN.1 및 GDMO의 통합환경을 보여주며, 이 시스템을 활용하면 효율적으로 망 관리를 할 수 있다는 것을 보여준다.

Design and Implementation of ASN.1/GDMO Development Environment for TMN

Young-Chul Kim†

ABSTRACT

ASN.1/GDMO is the management object used in network management of TMN(Telecommunication Management of Network). However, ASN.1/GDMO is not directly used for managing the network, but translated into a language with object-oriented paradigm. Therefore we need a development environment for handling ASN.1/GDMO. In this paper we present an integrated development environment(IDE) which consists of an editor and a browser. A user manages ASN.1/GDMO elements with GUI. The IDE is implemented with FLEX and BYACC in UNIX. And Objectivity DB is used as the DB and Tcl/Tk is used for developing GUI. This paper shows how the integrated environment of ASN.1 and GDMO works, and that it enables to manage efficiently the network.

키워드 : 망관리(TMN), ASN.1, GDMO, 사용자인터페이스(GUI : Graphic User Interface)

1. 서 론

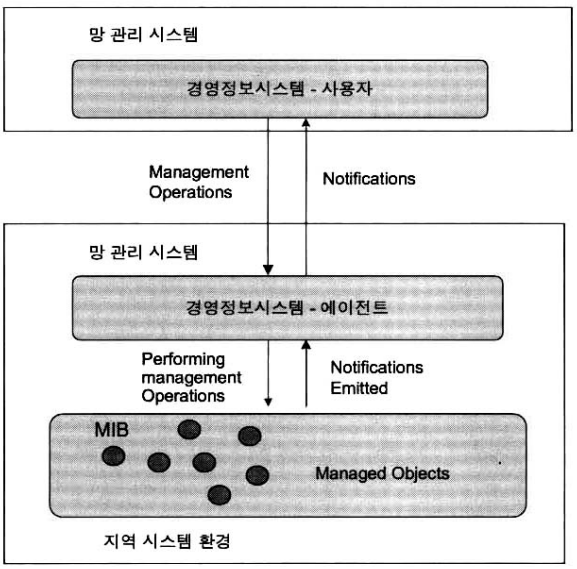
ITU-T에 의해 1988년에 처음 정의된 통신 관리망(TMN : Telecommunictions Management Network)은 다양한 종류의 운영체제와 통신 장비 사이에 표준화된 인터페이스를 이용하여 정보의 교환이 이루어지도록 한다. TMN의 방법론은 OSI 시스템 관리의 객체지향 패라다임을 적용하고 있으며, OSI에서 시스템 관리 모델로 다루어지는 정보는 관리되는 자원으로 "관리객체(manged object)"로 나타내어진다. 이 OSI 관리모델(MIN)은 시스템 관리 프로토콜에 의해 전송되는 관리 정보의 구조를 제공하기 위한 것으로 관리 객체를 다루는 정보 모델이다[1]. (그림 1)은 OSI 관리 정보 모델을 보여준다. 관리 정보 모델의 장점은 첫째, 관리시스템이 실제 자원을 구현과 독립되어 관리적인 측면에서 볼 수 있는 관점을 제시하고, 둘째, 원격 시스템과 관리 정보를 교환하는데 있어서 통일된 형식으로 올바르게 관리 정보를 해석할 수 있는 방법을 제시하며, 셋째, 캡슐화, 모듈화된 관리시스템을 구성함으로써 시스템의 확장성과 재사용성을 높이는 것이다. 이러한 기능을 가진 TMN은 다양한 종류의 운영체제와 통신 장비 사이에 표준화된 인터페이스를 이용하여 정보의 교환이 이루어지도록 하는 것이다[2-4].

이외에도 TMN에서는 통신망 관리를 위한 OSI 관리 모델과 정보모델, 표준 인터페이스 등의 여러 종류의 장비에대한 관리를 수행하고 있다. 이러한 체계 하에서는 인터페이스와 행위를 표준화된 형태로 정의하고 관리할 수 있는 도구가 요구된다. 이를 위해서 ASN.1(Abstract Syntax Notation One)[5]과 GDMO(Guide line Definition Managed Object)[6]가 표준으로 제정되었다[3].

ASN.1은 다양한 인터페이스나 통신 매체를 통하여 전송되어야할 정보를 기술하기 위한 언어이며, ASN.1 컴파일러는 ASN.1 명세를 특정 언어의 암호(encoding)와 해독(decoding) 루틴으로 바꾸어주는 작업이다[4-6]. 이러한 루

† 준 회 원 : (주)뉴스텍시스템즈 이사, 명지전문대학 교수
논문접수 : 2004년 3월 22일, 심사완료 : 2004년 6월 12일

틴들은 OSI 응용프로그램을 개발하는데 이용될 수 있으며, 비 OSI 통신 프로토콜을 구현하기 위해서도 이용될 수 있다. GDMO는 관리 정보를 구조적으로 모델링하는 기술 도구로 GDMO 그 자체로 망 관리에 이용되기는 어려우며 객체지향 언어로 번역되어 사용된다. GDMO를 객체지향 프로그램 언어로 번역하는 이유로 첫째, 네트워크 및 시스템 소프트웨어 개발자에게 객체지향 언어는 관리 객체의 기술이 GDMO에 의한 명세보다도 더 명확하며 이해하기 쉽다. 둘째, 객체지향 언어로 번역된 결과는 즉각적으로 MIB 구현에 필요한 OODB의 스키마로 표현될 수 있다.



(그림 1) OSI 관리모델

ASN.1과 GDMO의 개발 환경은 많이 발전되어 왔으나, 국내에서는 연구가 많이 미약하다. 또한 GDMO나 ASN.1 명세 언어를 처리할 수 있는 개발 툴이 없어 망 관리 개발자가 MIB 구현에 직접 응용할 수 있는 환경이 필요하다. 따라서 본 논문에서 제시한 시스템은 다음과 같은 많은 장점이 있다. 첫째, 객체지향 언어를 사용하는 네트워크 및 시스템 소프트웨어 개발자에게 관리 객체의 인터페이스가 GDMO에 의한 명세보다 더 명확하며 이해하기 쉽다. 둘째, 번역의 결과는 즉각적으로 MIB 구현에 사용할 수 있는 OODB의 스키마로 표현되어질 수 있다. 셋째, ASN.1으로 정의된 type을 프로그래밍 언어(C, C++등)로 변환하는 도구를 개발할 수 있는 원천 기술을 확보할 수 있다. 넷째, 자동화된 도구에 의해 정형화된 망 관리 소프트웨어의 제공이 가능하다. 다섯째, GDMO 컴파일러 및 스키마 테이블을 구현함으로써 망 관리 기술을 향상시킬 수 있다. 이 외에도 ASN.1/GDMO 통합환경은 TMN 망 관리 기능 구현이나 B-ISDN망 관리기능에 지대한 영향을 줄 것으로 기대된다.

본 논문의 구성은 다음과 같다. 제 2장에서는 관련 연구 사례에 대해서 언급한다. 제 3장에서는 ASN.1/GDMO 개발 환경모델에 대해서 언급하였으며, 4장에서는 구현 환경 및 평가에 대해서 설명하고, 마지막으로 5장에서는 결론을 맺었다.

2. 관련 연구

망 관리를 위한 ASN.1 개발환경에 대한 연구는 ASN.1 표현방법에 관한 표준이 정의되면서부터 이루어져 왔다. ASN.1의 표기법은 복잡하고, 개념 정립이 완전히 이루어지지 않아 ASN.1 컴파일러를 개발하는데 많은 어려움이 따른다. ASN.1 개발 환경에는 ASN.1 컴파일러, 브라우져, DB 인터페이스, 원시 코드 생성기, 응용 프로그램 등 많은 요소를 가지고 있다. 그러나 지금까지 개발된 연구사례를 보면 이러한 요소들을 모두 포함하는 전제적인 통합환경이 없다. 더군다나 인터넷이 급속히 발전하게 되면서 많은 정보가 처리가 이루어지는 시점에서 DB와의 연동은 필수적인 요소이다. 왜냐하면 ASN.1 명세는 언어 자체에 "import"나 "export" 기능을 포함하고 있으므로, 표준 기관이나 비공식적인 기관에서 정의된 많은 모듈들을 DB에서 포함하고 있어야한다. "import" 기능은 외부 모듈에서 정의된 형을 이용할 수 있는 기능이고, "export" 기능은 본 모듈에서 정의된 형을 외부 모듈에서 사용할 수 있게 해주는 기능이다. 그러나 지금까지 개발된 ASN.1 컴파일러들은 대부분 DB와의 연동이 되지 않고 있으며, 단지 구문분석과 의미분석에 의한 다른 언어로의 변환기 역할만을 수행하고 있다. 기존의 ASN.1 컴파일러에는 MAVROS[7], BBN[8], Snacc[9], ISODE의 PEPY/POSY[10], UBC의 CASN1[11] 등이 있다.

MAVROS는 크게 두 부분으로 구성되어 있다. 하나는 전처리기로 모든 ASN.1 명세의 형을 지원하며, 모든 ASN.1 값들을 초기화하는 단계이다. 두 번째 부분은 생성기 부분으로 여러 다양한 인코딩 루틴들을 생성하며, 프로그래밍 디버깅 도구도 지원한다. BBN 컴파일러는 ASN.1 명세를 C++로 변환할 수 있으며, Solaris, MSDOS, Win3.1에 이식이 가능하다. 이 컴파일러는 자동 구문 검사기능이 있다. Snacc은 ASN.1을 C나 C++ 코드로 바꿀 수 있는 변환기이며, GUI를 지원한다. CASN1 ASN.1 컴파일러는 3단계 컴파일러이다. 1, 2단계는 3단계를 위한 전처리기를 위한 함수로 구성되어 있으며, 3단계의 입력이 된다. 이때 생성되는 파일은 "out1"과 "out2" 파일이다. 이 컴파일러의 제 1단계에서는 복합 형 정의 및 이름이 할당된다. 제 2단계에서는 ASN.1 명세를 상향식(bottom-up) 순서로 형 정의를 하였으며, 결과 형들은 C 언어의 구조와 같은 구조로 바꾸는 단계이다. 제 3단계에서는 오류 회복과 C 코드를 생성한다.

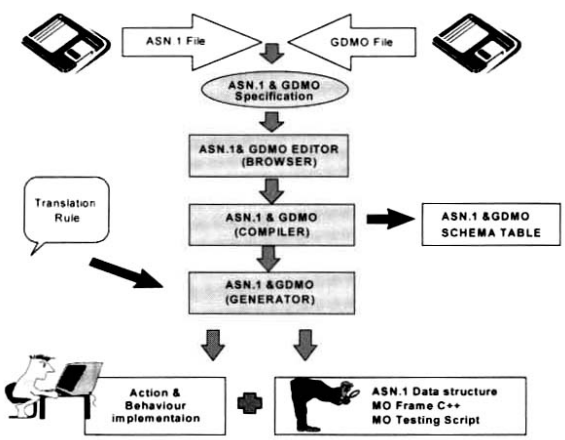
GDMO의 관리도구로는 HP OpenView GDMO 모델링 툴셋과 NOMAD GDMO 컴파일러 등이 있다. HP OpenView는 네트워크, 시스템, 응용프로그램 및 데이터베이스를

포함하는 통합된 네트워크와 시스템 관리를 하는 도구이다. 이 소프트웨어 도구는 OSI NM 객체 명세를 이용하여 그래픽 설계와 수정을 도와준다. GDMO 브라우저는 그래픽 사용자 인터페이스를 제공하며 객체 사전을 사용하여 관리 객체를 저장한다. 관리 객체의 입출력을 위해 GDMO import 툴과 GDMO export 툴을 제공하고 있다. OSCOM에서 만든 GDMO 컴파일러는 GDMO와 ASN.1 명세를 저장하는 GDMO 사전과 관리 객체와 ASN.1 정의를 포함한 텍스트 파일을 입력으로 받아 처리하는 컴파일러인 GDMO Dictionary Loader와 GDMO 브라우저 및 사전의 정의를 쉽게 접근할 수 있게 해주는 GDMO 질의어를 제공하고 있다.

그러나 이러한 기존의 컴파일러들은 특정한 컴퓨팅 환경에서만 동작하므로 이식성이 약하다. 또한 이 컴파일러에 의해 생성된 결과물은 특정 목적에 의존하고 있기 때문에 이를 이용하는 사용자나 설계자, 관리자들은 자신의 목적에 맞게 처리할 수 없다. 따라서 자신의 요구 사항에 맞추어 사용하기 위해서는 컴파일러의 결과물을 변환하여야하는데, ASN.1의 복잡한 자료 구조 때문에, 이 작업은 매우 어렵다. 또한 ASN.1 컴파일러나 GDMO 컴파일러 등은 많이 존재하나 편집과 관리, 사용자 인터페이스 지원, DB 연동 등과 같은 통합환경은 거의 지원하지 않는다.

3. ASN.1/GDMO 통합 환경 모델

망 관리를 위한 원시 코드 브라우저는 ASN.1/GDMO 명세 언어를 입력으로, 즉 관리 객체와 ASN.1 정의를 포함한 텍스트 파일을 입력으로 ASN.1/GDMO 컴파일러나 중간 자료 구조인 ASN.1/GDMO 스키마 테이블로 번역하여 삽입한다. 이렇게 작성된 ASN.1/GDMO 스키마 테이블은 템플릿 스키마 데이터베이스에 저장된다. 저장된 중간 자료 구조는 ASN.1/GDMO 원시 코드 생성기에 의해 객체지향 파라다임을 가지는 언어로 번역되게 된다. 본 논문에서 구현한 ASN.1/GDMO 개발환경 모델은 다음 (그림 2)와 같다.



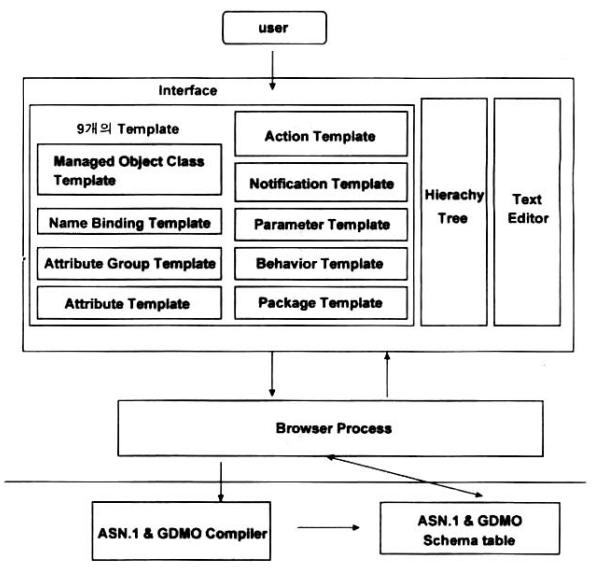
(그림 2) ASN.1/GDMO 개발 환경 모델

(그림 2)에서 처럼 ASN.1/GDMO 개발 환경은 ASN.1/GDMO 규격의 프로그램을 입력으로 하여 분석을 수행하는 ASN.1/GDMO 분석기와 편집과 수정을 위한 편집기, 편집 및 수정된 결과를 컴파일하여 스키마에 필요한 자료를 생성해 내는 컴파일러, 분석된 결과를 저장하는 스키마테이블(Schema Table), 저장된 결과를 원래의 원시 입력 형태로 보여주는 프리티프린터(Pretty Printer)로 구성되어 있다.

3.1 ASN.1/GDMO 편집기

ASN.1/GDMO 편집기는 ASN.1/GDMO 각각의 명세를 생성, 수정, 삭제 등의 편집을 지원하는 메뉴를 제공한다. ASN.1/GDMO 편집기는 여러 선택 사항을 제공하고 있다. DB에 저장된 ASN.1/GDMO 템플릿를 그래픽 형태로 나타내어 사용자 인터페이스를 강화하고, ASN.1/GDMO 템플릿에 익숙지 않은 사용자도 쉽게 이용할 수 있도록 해야한다. 편집기는 사용자에 의해 입력된 ASN.1/GDMO 명세 내용을 컴파일하여 ASN.1/GDMO 스키마 테이블의 형태로 바꾸어 데이터베이스에 저장하는 부분도 담당한다. 본 논문에서 구현한 ASN.1/GDMO 편집기는 다음 (그림 3)과 같다.

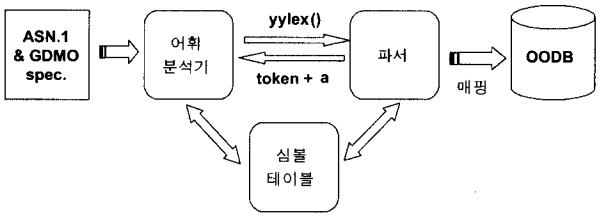
(그림 3)에서 보듯이 ASN.1/GDMO 편집기의 인터페이스는 9개의 템플릿 브라우저 및 상속, 포함 트리 브라우저와 텍스트 에디터를 가지고 있다. 사용자는 이러한 브라우저를 통하여 ASN.1/GDMO 명세을 입력하고 브라우저 프로세스는 이러한 입력을 처리하여 중간 스키마 테이블의 형태로 변역하는 과정을 거친다. 편집기는 찾기 기능을 가지고 있다. 템플릿의 이름이나 템플릿의 종류를 선택하면 ASN.1/GDMO 스키마 테이블의 내용을 템플릿 브라우저로 표현하여 준다.



(그림 3) ASN.1/GDMO 편집기의 구조

3.2 ASN.1/GDMO 컴파일러

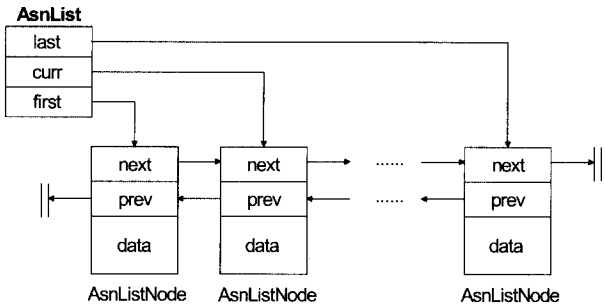
ASN.1/GDMO 컴파일러를 구현하기 위해서는 많은 컴파일러 보조 도구들이 필요하다. 대부분 이용되는 컴파일러 보조 도구들은 어휘분석 도구로서 lex[12]를 이용하며, 구문분석 도구로 yacc[13]을 이용하였다. 다음 (그림 4)는 ASN.1/GDMO 컴파일러 구성에 대해서 보여준다.



(그림 4) ASN.1/GDMO 컴파일러 개발환경

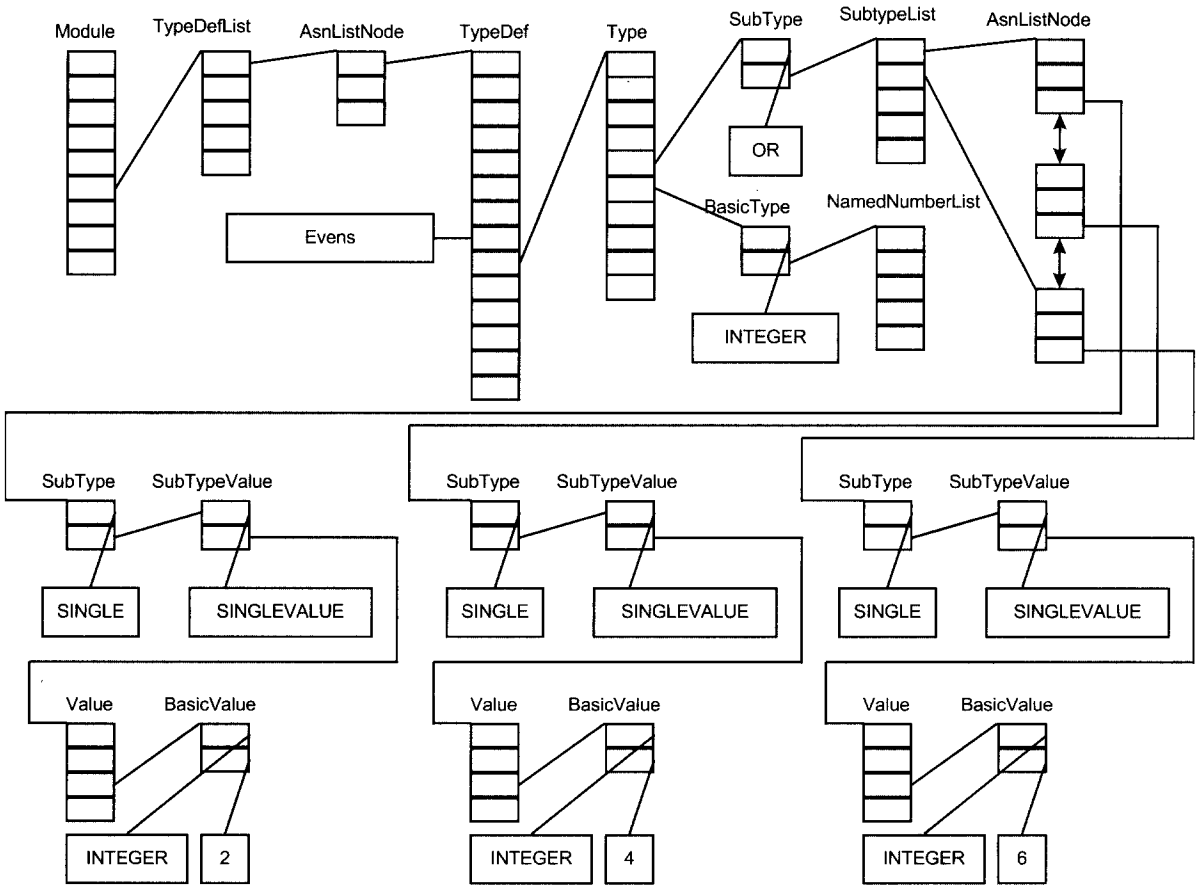
3.3 내부 자료 구조

ASN.1/GDMO 컴파일러에 의해 구성되는 구조에서 많은 부분이 리스트의 형태로 구성된다. 이러한 리스트를 별도의 구조에 따른 처리함수를 구성하기에는 많은 양의 코드를 작성하여야 하며, 또한 사용하기 어렵다. 따라서 리스트 구조를 관리하기 위한 asn1List라는 자료구조를 정의하고 이 구조를 사용하는데 필요한 함수들을 정의하여 복잡한 구조를 최대한 단순화 하였다. 다음 (그림 5)는 ASN.1의 리스트 자료 구조를 보여준다.



(그림 5) ASN.1 리스트 구조

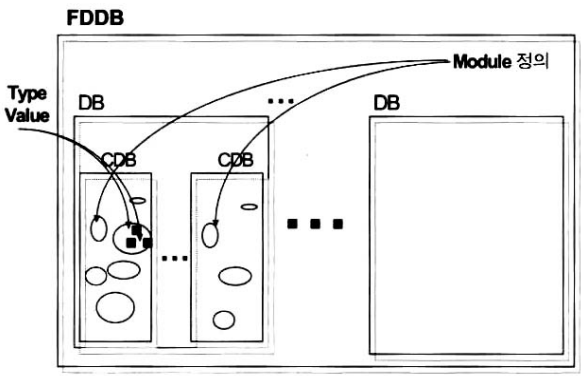
AsnListNode 구조는 각 자료구조에 대해 양 방향 리스트를 구성할 수 있도록 next와 prev 포인터를 가지고 있다. 이러한 구조의 정의는 각각의 정의된 자료구조별로 동일한 형태로 접근될 수 있도록 하는 자료구조와 보조함수들이 정의되어 있다. 예를 들면 다음과 같은 ASN.1 문장이 "Evens::= INTEGER(2 | 4 | 6)"와 같다면, 내부 표현 자료 구조 형태는 다음 (그림 6)과 같다.



(그림 6) "Evens : : = INTEGER (2 | 4 | 6)" 내부 자료 구조

3.4 ASN.1 스키마 테이블

ASN.1/GDMO 템플릿 스키마 테이블은 ASN.1/GDMO 컴파일러에 의해 컴파일된 문법 정보를 데이터베이스에 저장하기 위한 구조이다. ASN.1/GDMO 컴파일러가 생성하는 중간 자료 구조는 객체지향 형태로 되어있다. 그래서 이러한 구조를 저장하는 스키마 테이블의 형태는 OODB가 적당하다[14]. OODB를 사용하면 자료 접근시 인터페이스가 쉽고, 객체지향 패라다임의 특성을 가진다는 장점이 있다. 데이터베이스 객체 모델을 설계할 때 중간 자료 구조를 정의한다. 중간 자료 구조의 형태는 C++ 언어의 클래스와 유사하다. 이를 위해 Objectivity의 DDL(Data Definition Language)을 사용하여 스키마 테이블을 구성하였다[1]. ASN.1/GDMO에서는 Objectivity DB를 사용하는데 (그림 7)은 ASN.1 /GDMO 모듈과 Objectivity DB를 나타낸다.



(그림 7) ASN.1/GDMO 모듈과 Objectivity DB 구조

(그림 6)에서 처럼 Objectivity DB 구조는 다수의 FDDB (Federated DataBase), DB(DataBase), CDB(Contained Data-Base) 체계로 이루어져 있다. 가장 상위 계층인 FDDB는 Objectivity DB에서 여러개 만들 수 있으며, 하나의 FDDB는 다수의 DB를 포함할 수 있다. 따라서 하위 계층의 DB(혹은 CDB)를 생성하거나 열기를 할 경우, 상위 계층의 FD(혹은 DB)는 이미 열기가 수행되었을 때만 가능하다. 또한 FD를 삭제하였을 경우에 그 하위 계층에 있는 모든 DB 및 CDB가 함께 삭제된다.

4. ASN.1/GDMO 구현환경 및 실험

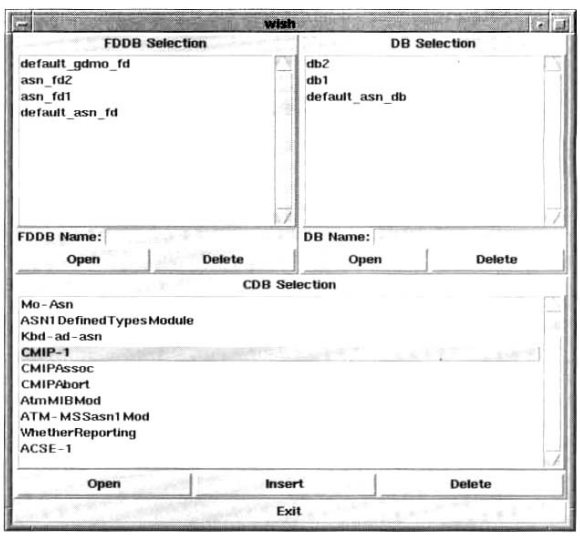
4.1 구현환경

본 논문에서 사용한 ASN.1 명세 언어의 문법은 표준안에 제정된 문법을 문맥 자유 문법 형태로 바꾸어 사용하였다. 문맥 자유 문법으로 변환하는 과정에서 많은 충돌을 일으켰지만 문법의 의미를 바꾸지 않고 문법을 재 기술함으로써 충돌을 해결하였다. 또한 어휘 분석을 위해서 GNU의 flex를 이용하였으며, 구문 분석을 위해서 byacc을 이용하였다. 또한 Objectivity DB[14]를 이용하였으며, DB와의 인터페이스를 위한 도구로 Tcl/Tk[15]를 이용하여 구현하였다.

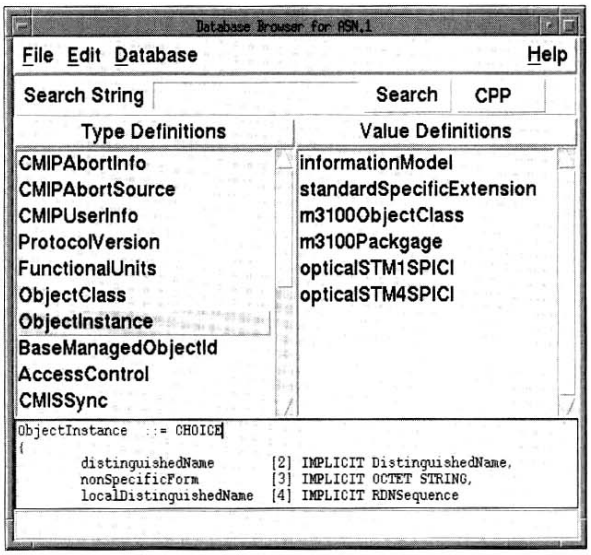
4.2 ASN.1 개발환경

사용자에게 시각적으로 보여주는 브라우저는 DB에 저장된 ASN.1의 모듈, 형, 값에 관한 내용을 보여준다. 본 시스템의 초기화면은 (그림 8)과 같다. (그림 8)은 사용자가 먼저 원하는 DB를 선택하면 이 DB에 포함되어 있는 모듈이 나타난다. 또한 함으로써 해당되는 모듈을 보여준다. 해당되는 모듈을 선택하면 (그림 9), (그림 10)과 같은 타입 및 모듈 브라우저가 나타난다. (그림 8)은 사용자가 선택한 모듈을 DB에서 읽어들여 사용자에게 보여주는 모듈의 리스트와 그 모듈의 상세 정보를 보여준다. 사용자가 모듈 정보를 호출하게 되면 DB로부터 모듈 리스트를 얻어오며, 모듈 리스트에 있는 특정 모듈을 두번 클릭하면 그 모듈의 상세 정보를 다시 DB로부터 읽어와서 브라우저를 통해 화면에 보여주게 된다.

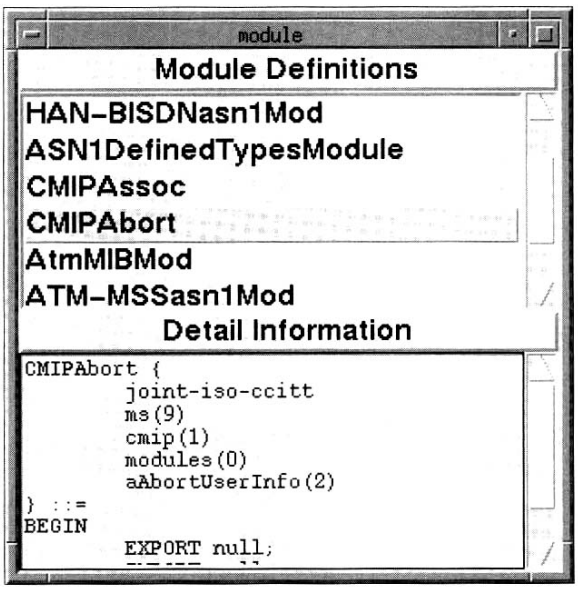
(그림 9)는 사용자가 (그림 8)에서 선택한 특정한 모듈의 형과 값에 대한 정보와 상세 정보를 브라우저를 통해 보여준다. 이 과정 역시 모듈에서 설명한 방법과 마찬가지로 이루어진다. (그림 9)의 "search" 버튼과 "CPP" 버튼은 각각 DB를 검색하는 버튼과 해당하는 모듈의 변환된 C++ 코드를 보여주는 버튼이다.



(그림 8) ASN.1 브라우저의 초기화면



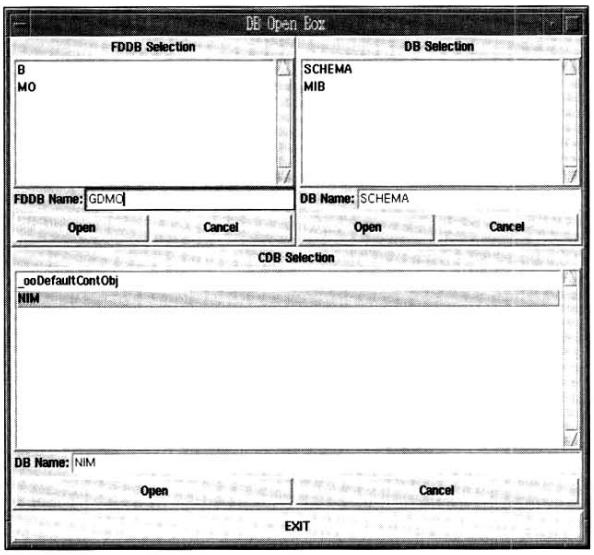
(그림 9) Type 및 Value 브라우저



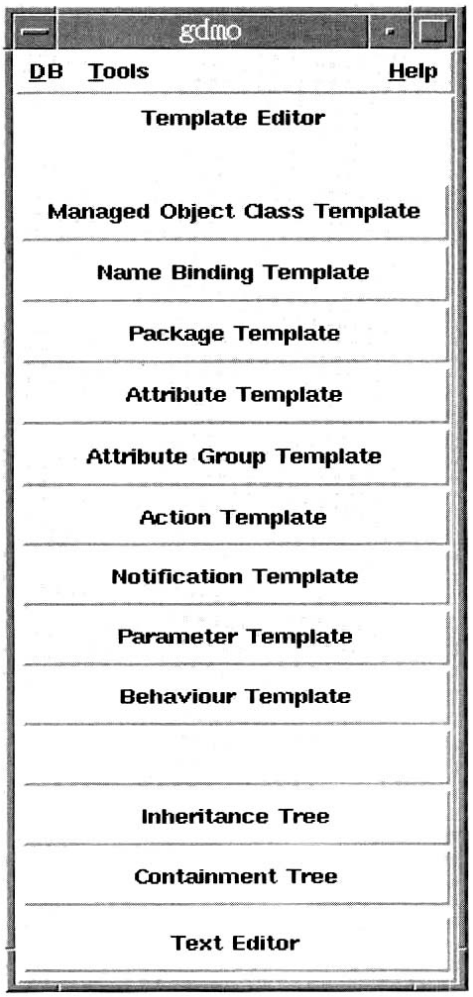
(그림 10) Module 브라우저

4.2 GDMO 개발환경

다음 (그림 11)과 (그림 12)는 GDMO의 초기화면과 주 메뉴 화면을 보여준다. 그림에서 나타나듯이 GDMO의 초기 화면은 DB로부터 읽어오는 정보를 표시하는 부분이 있으며, 앞서 언급한 특정한 DB를 선택하여 불러올 수도 있다. DB 선택이 완료되면 사용자가 여러 작업들을 주로 수행하게 되는 주 화면이 제시된다. 주화면의 구성은 (그림 12)에서 볼 수 있는 것처럼 상단의 메뉴, 9개의 템플릿을 작성할 수 있는 선택 버튼, 그리고 텍스트 편집기를 호출할 수 있는 버튼으로 구성되었다.



(그림 11) GDMO 초기화면



(그림 12) GDMO 주 메뉴 화면

다음 (그림 13)은 GDMO 9개의 템플릿 중 파라메타 (Parameter) 템플릿의 개발 환경을 보여준다. 파라메타 템

플릿은 파라메타 구문과 행위를 기술하고 기록, 저장할 수 있도록 해 준다.

(그림 13) Parameter Template

본 시스템의 특징은 ASN.1/GDMO 개발환경의 통합적인 지원이다. 또한 대부분의 다른 시스템들은 브라우저를 지원하지 않지만, 본 논문에서는 효율적으로 ASN.1/GDMO를 관리할 수 있는 브라우저를 지원한다. 또한 DB와 연동이 가능하게 함으로써 ASN.1/GDMO의 관리 객체를 보다 쉽게 관리할 수 있도록 설계한 것이 특징이다. 또한 본 시스템의 내부 자료구조는 향후에 ASN.1을 C++와 같은 객체 지향 패라다임으로 쉽게 번역할 수 있도록 설계되었다. 이는 ASN.1 원시코드 자동 생성기를 만들고자 할 때 유용하게 활용될 것으로 보인다.

5. 결 론

본 논문에서는 ASN.1/GDMO의 명세를 입력으로 받아들여 컴파일한 후, 컴파일 결과를 DB에 입력하는 ASN.1/GDMO의 개발 환경을 구현하였다. ASN.1/GDMO 개발 환경에는 편집기, 컴파일러, DB 및 스키마 테이블, 브라우저 및 원시 코드 생성기를 포함하고 있다. ASN.1/GDMO 편집기는 쉽게 DB와 연동이 가능하도록 만들었으며, 다양하게 관리객체들을 다룰 수 있도록 설계된 것이 본 논문의 특징이다. 이외에도 본 논문에서는 ASN.1/GDMO의 컴파일 기법과 스키마 테이블과 같은 자료구조를 개발하여 효율적으로 내부 표현구조를 다룰 수 있었다. ASN.1/GDMO 프리티 프린터는 DB에 있는 자료를 프리티 프린팅하여 사용자에게 시각적으로 보여지도록 구현하였다. 따라서 사용자는 브라우저를 통하여 기초적인 DB 조작을 할 수 있으며, DB에 존재하는 ASN.1/GDMO 명세를 조작하거나 검색할 수 있는 기능도 갖추고 있다.

향후에는 본 연구에서 생성된 결과물을 토대로 ASN.1 유틸리티를 위한 라이브러리를 구현하고, 망 관리 정보모델의 기능을 위한 원시코드 자동 발생기를 구현하여야 한다. 또한 DB에 저장된 목적 코드를 이용하여 코드의 단일화 및 정보 교환에 필요한 응용프로그램을 개발하여야 할 것이다.

참 고 문 헌

[1] Uyless Black, "Network Management standards," McGraw-Hill, 1994.

[2] CCITT Recommendation X.721(1992) | ISO/IEC 10165-2, Information techmology - Open Systems Interconnection - Structure of management information - Definition of Management information, 1992.

[3] CCITT Recommendation X.722(1992) | ISO/IEC 10165-4, Information techmology - Open Systems Interconnection - Structure of management information - Guidelines for the definition of managed object, 1992.

[4] CCITT Recommendation X.720 | ISO/IEC 10165-1 : 1992, Information techmology - Open Systems Interconnection - Structure of management information - Management information model, 1992.

[5] D. Steedman. *ASN.1 The Tutorial and Reference*, 1990.

[6] "TMN C++ : GDMO++, Managed Object Application Programming Interface," NM Forum, 1996.

[7] C. Huitema, *General Presentation of the MAVROS Compiler*, INRIA, 1990.

[8] IOS developers, *BBN Systems and Technology ASN.1 Compiler* version 1.4, available via at http://ests.bbn.com/ASNSRC.html, 1995.

[9] Mike sample, *Snacc ASN.1 Compiler version 1.2,* available via at http://www.fokus.gmd.de/ovma/mug/archives/mug-software/snacc.html, 1997.

[10] ISODE, available via at http://ftp.doc.ic.ac.uk/packages/isode, 1997.

[11] Nikos Drakos, 1995, *CASN1-ASN.1 to C Compiler,* available via at http://www.atri.curtin.e.edu.au/~duke/honours/compiler/casn1/casn1.html, 1995.

[12] V. Paxson, *On-line manual pages distributed with the Flex*

software package, available via anonymous ftp from ftp.ee.lbl.gov or prep.ai.mit.edu, 1990.

[13] R. Corbett, *BYACC parser generator version 1.9,* available via anonymous ftp.cs..berkeley.edu:/ucb/4bsd/byacc.tar.Z.

[14] "Objectivity/DB C++ Application Development : Version 3," Objectivity Inc, 1995.

[15] John K. Ousterhout "Tcl and the Tk Toolkit," Addison-Wesley, 1994.

김 영 철

e-mail : yckim@ss.ssu.ac.kr

1990년 한남대학교 전자계산학과(학사)

1996년 숭실대학교 전자계산학과 석사

2003년 숭실대학교 컴퓨터학과 박사

현재 (주)뉴스택시스템즈 이사, 명지전문대학 겸임조교수

관심분야 : 프로그래밍 언어, 망관리, 컴파일러, XML, 컴퓨터 통신