

소프트웨어 개발 방법론 지원 시스템에 관한 연구

—정보시스템 개발을 위한 메타시스템의 설계를 주로—

正會員 李 淑 姬*

A Study on System for Supporting Software Development Environment

—On the Design of Meta System Supporting the Development of Information System—

Sook Hee LEE* *Regular Member*

要 約 이 논문의 목적은 데이터베이스로의 정보 저장에 관한 작업을 담당하는 FLA(Front-end Language Analyzer)와 여기에서 추출된 데이터베이스의 일관성과 완전성을 쉽게 검사할 수 있고 사용자의 요구에 따라 다양한 형태의 문서를 생성해 주는 BIS(Back-end Information System)를 구축함에 있다. 앞으로 본 시스템은 소프트웨어 시스템 개발에 실제로 사용됨으로써 소프트웨어의 생산성과 신뢰성 향상에 기여할 수 있으리라 예상된다.

ABSTRACT The objective of this paper is to design and implement FLA(Front-end Language Analyzer) storing information to database, and BIS(Back-end Information System) analyzing completeness and consistency of database and generating all kinds of documents needed by user. By using this system to develop systems, we expect enhancement of software productivity and reliability.

I. 서 론

현재의 소프트웨어 개발 환경에서의 문제점은 사용되고 있는 개발 방법론이 소규모의 소프트웨어 개발과 대규모의 소프트웨어 개발에 동시에 적용될 수 없고, 방법론마다 응용 분야가 제한된다는 점이다. 또한 대부분의 개발 작업이 개발 일정을 맞추고 있지 못하여 보수 유지가 어려울

뿐만 아니라, 사용자가 원하는 기능을 제대로 수행하지 못하여 사용자의 불만을 초래하는 상황을 유발하는 등, 소프트웨어의 개발은 위기 상황에 처해 있다. 더욱이 컴퓨터의 하드웨어 기술은 급진적으로 발전하고 사용자의 소프트웨어에 대한 요구 사항도 점차 복잡해지고 있다. 따라서 이러한 요구 사항을 충족시키지 못하는 기술의 격차와 관리 기술의 부적합성, 전문 인력의 부족 현상은 이러한 위기를 더욱 심화시키고 있다.

사용자가 요구하는 소프트웨어의 기능을 완벽

* 동신공과대학 전자계산학과
Dept. of Computer Science, Dongshin Engineering College
論文番號: 90-82(接受1990. 5. 11)

하게 이해하여 명세화시키지 않고서는 개발 작업의 성공을 기대하기는 어렵다. 따라서 소프트웨어 요구 사항 명세서를 완벽하게 작성하는 것이 소프트웨어 개발 작업을 성공적으로 수행하는데 필수적인 요소이다. 요구 분석 과정은 소프트웨어 개발 계획서를 기초로 하여 계획서에 설정된 소프트웨어 영역을 보다 상세히 정의하고, 소프트웨어에 대한 여러가지 해결 방안을 비교 분석하여 소프트웨어의 기술 요소로 명명하는 과정이다.

이러한 일련의 처리 과정을 도울 수 있도록 여러 가지 요구 사항 분석 도구들이 개발되어 왔는데, 이들은 크게 수동식 도구, 자동식 도구, 메타 시스템으로 나누어진다.

수동식 도구는 분석 과정을 체계적으로 수행할 수 있도록 형식화된 기법으로서 일반적으로 도형적 분석 방법을 사용한다.

수동식 도구의 한 방법으로서 구조적 분석 및 설계 기법이 있는데, 이것은 시스템을 묘사하는 그래픽 언어와 이 언어의 사용 방법 및 관리 지침들로 구성되어 있다. 이 방법은 수작업에 의존하는 도구로서 단순하다는 공통점이 있지만, 기술 방법은 매우 엄격하고 응용 분야가 일반적이다.

수동식 도구의 또 다른 방법으로서 구조적 분석 기법은 하향식, 기능적 분해 방법을 이용하여 시스템의 요구를 정의하는 기법이다. 구조적 기법은 시스템의 도형적 모델에 기반을 두고 있기 때문에 사용자가 시스템의 모델을 쉽게 이해할 수 있어 분석가와 사용자 사이의 의사소통을 원활하게 한다. 따라서 오류 및 부정확한 기술들을 발견할 기회가 많아지고 기능적 분해에 의해서 시스템이 다루기 쉬운 부분으로 나누어지기 때문에 변경이 발생할 경우 수정하기가 쉽다. 구조적 분석 기법은 요구 분석 작업을 체계화하는데 크게 도움이 되고 응용에 따른 부하가 적어 널리 사용되고 있으나 DFD 상의 오류나 완전성, 일관성을 정확하게 검사할 수 없다. 이 기법이 좀 더 유용하게 사용되기 위해서는 오류나 완전성을 검사하기 위한 자동화 도구가 필요

하다.

자동화 도구는 키워드와 자연 언어로 표현된 명세화 언어를 사용하여 요구 사항을 기술하고 이 요구 사항을 처리기에 입력으로 주면 명세서와 함께 요구 사항의 일관성 및 구조에 대한 일련의 검토 결과를 출력한다.

PSL/PSA¹⁰⁾가 대표적인 자동화 도구인데 요구 명세를 기술하는 언어(PSL)와 이 언어의 처리기(PSA)로 구성된다. PSA는 PSL로 작성된 요구 명세를 처리하는 자동화된 분석기로서 PSL/PSA에 의해 제공되는 기능들을 활용할 수 있는 명령어 처리기, 입력된 PSL 문장들의 완전성, 일관성을 검사하는 분석기, 데이터베이스 관리 시스템, 보고서 생성기 등의 다수의 유틸리티 프로그램으로 구성된다. PSL/PSA는 사용자의 요구를 구분 규칙을 갖는 기술 언어로 표현하여 컴퓨터에 의해 처리가 가능하도록 하여 시스템 요구의 명확성, 일관성, 완전성 등을 향상시킨다. 또한 수집된 요구 자료는 일단 데이터베이스에 저장되므로 내용의 변경, 추가 및 삭제가 쉽고 필요한 내용은 언제나 보고서 형태로 작성할 수 있다. 그러나 PSL/PSA는 배우기가 어렵고 사용하기가 불편하다는 단점이 있다.

SREM¹¹⁾ 역시 자동화 도구 중 하나로서 미국의 미사일 방어 시스템의 실시간적인 차극에 아주 민감한 요구 사항들을 분석, 설명하기 위해 특별히 고안된 것이다. 그 구성 면에서는 PSL/PSA 와 비슷한 점이 많지만, 실시간 embedded 시스템에 적용이 가능하도록 병행 처리, 직분레이션 기능 등을 추가한 대규모 시스템이다.

메타 시스템은 자동화된 지원 도구와 방법들의 통합된 환경을 구축하는 추세에 맞추어 이러한 개발 환경을 지원해 주는 시스템이다.

메타 시스템의 하나인 RLP(Requirements Language Processor)¹²⁾는 응용 시스템 모델인 extended finite state machine 모델을 기본으로 하는 모든 종류의 언어로 작성된 입력을 받아들여 처리한다. 이 도구는 실시간 처리 시스템의 개발을 위한 언어의 분석을 위한 것으로 정적인

구조와 동적인 불일치에 대한 오류를 검사하여 발생된 오류를 보고서를 통하여 출력시킨다.

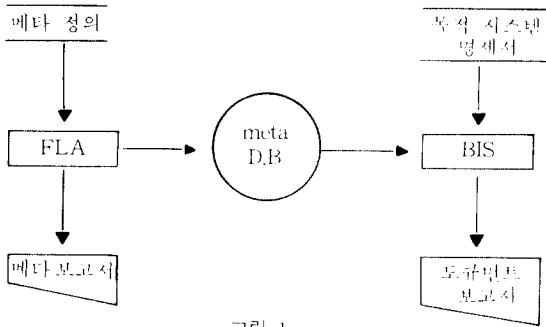


그림 1.

ISLDM / SEM⁽⁶⁾은 생명 주기 지원 시스템 프로세서의 구현을 위한 종합적인 소프트웨어 도구인 메타 시스템으로 목적 시스템을 기술하는 언어와 방법론, 그리고 자동적인 도구들로 구성된다.

앞에서 언급한 바와 같이 하나의 시스템을 위해 개발된 방법론은 그 시스템과 상이한 특성을 가진 시스템의 개발에는 적용하기가 힘들다. 이러한 문제점을 해결하기 위해 메타 시스템의 하나인 SEM이 도입되었다. 그러나 SEM은 FORTRAN 으로 구현되어 있고, 설치 기종과 운영 체제도 제한되어 있어서 이식성과 호환성이 낮다는 문제점을 가지고 있다.

따라서 본 연구는 이러한 SEM을 기반으로 하여 SEM의 특성을 살리고 단점을 보완하는 새로운 개발 방법론 지원 시스템을 UNIX 운영 체제 안에서 개발하고자 한다.

이 시스템은 데이터베이스로 정보 저장에 관한 작업을 담당하는 부시스템인 FLA(Front-end Language Analyzer)와 다양한 형태의 문서를 생성하는 BIS(Back-end Information System)로 구성되어 있는데, 그 대략적인 구조는 그림 1과 같다.

FLA는 목적 시스템 개발 환경을 만들어주기 위해 시스템 기술 언어를 입력으로 받아들여 데이터베이스 스키마와 시스템 화일들을 생성시키는 기능들을 한다. 그리고 시스템 기술 언어는 ER 모델을 이용하여 목적 시스템을 모델링한 결과를 기술하는 언어이다.

FLA는 구조는 그림 2에 나타내었는데, 다음과 같은 기능을 수행한다.

- 1) 시스템 서술은 이미 정의한 언어로 accept 한다.
- 2) 시스템 서술을 포함하는 데이터베이스를 운영한다.
- 3) 데이터베이스에 저장된 시스템 서술에 대한 분석을 수행한다.
- 4) 시스템 서술에 기반을 둔 문서나 그 밖의 출력을 생성한다.
- 5) 생명 주기 동안의 작업을 조절하고 제어 기능을 수행한다.⁽⁷⁾

FLA는 선택된 방법론에 기반을 둔 기본 모델과, 모델에 기반을 둔 작업을 지원하는 툴 시스템으로 구성된다. 이 중에서 툴 시스템은 기술된 시스템 서술을 저장하는 데이터베이스, 모델에 따라 목적 시스템에 관한 정보를 표현하는데 사용되는 시스템 서술 언어, 시스템 서술 언어로 표현된 시스템 서술을 수용해서 데이터베이스를 갱신하고 데이터베이스 내의 정보에 따라 분석하고 보고서를 생성하는 프로세서로 구성된다.

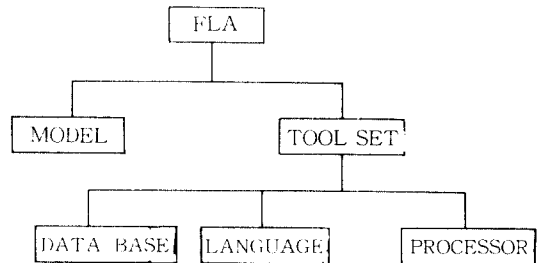


그림 2.

II. FLA(Front-end Language Analyzer)

2.1 모델

현실 세계의 추상화된 형태인 모델을 통해

적용할 방법론의 개념적 스키마 및 데이터베이스의 스키마가 결정되는데, 각 방법론의 개발을 위한 메타 시스템의 모델은 범용적이어야 하므로 FLA에서는 Chen의 ER 모델을 채택하였다.⁽⁹⁾⁽¹⁰⁾

ER 모델에서는 정보가 entity, relationship, attribute의 세 가지 기본 요소로 표현되는데, entity는 실제계의 관심 대상인 객체를 나타내고, 이것의 특성은 attribute로 표시하며, entity 간의 연결 관계는 relationship으로 표현한다.

2.2 메타 정의

FLA의 입력이 될 메타 정의의 구분은 ER 모델의 specification으로, 그 syntax는 부록에 나타내었다.

메타 정의는 OBJECT, PROPERTY, TEXT, RELATIONSHIP 등으로 구성되는데, 그 각각의 기능은 다음과 같다.

1) OBJECT : 메타 정의의 기본 단위로 PROPERTY 와 TEXT에서 그 속성이 정의된다. 그리고 다른 OBJECT와의 연관성을 RELATIONSHIP에서 정의된다.

2) PROPERTY : OBJECT의 속성을 정의하는데 사용되고 한 단어로 구성된다.

3) TEXT : PROPERTY와 마찬가지로 OBJECT의 속성을 정의하는데 사용되지만 그의 값에 대한 길이 제약이 없다는 점이 특징이다.

4) RELATIONSHIP : OBJECT간의 관계를 정의하는데 사용된다. 한 part에서 정의되는 OBJECT는 여러개가 올 수가 있고 각 part의 이름은 한 RELATIONSHIP에서 유일해야 한다. 그리고 해당 RELATIONSHIP내의 PARTS, CONNECTIVITY, KEYWORD에서 연관성을 유지해야 한다.

2.3 데이터베이스

데이터베이스는 개발하려는 목적 시스템의 해당 정보 및 데이터가 저장되는 장소이며, 이 데이터베이스를 통해 원하는 문서를 뽑거나 질의를 수행할 수 있고, 특정 보고서를 만들어 줄

수 있다. FLA에서 ER 모델로 기술된 개념 스키마인 메타 정의를 관계 데이터 모델의 스키마로 변환시킨다. 그리고 관계 데이터베이스에 대한 DDL(Data Definition Language)과 DML(Data Manipulation Language)의 중간 코드를 생성하게 되어 특정 유형의 데이터베이스에 관계없이 응용성과 이식성이 높아진다.

2.4 프로세서

각 프로세서를 제어하는 프로세서는 다음의 세 가지로 구분할 수 있다.

1) 언어 프로세서 : 시스템에 입력될 언어의 syntax, semantics를 이용하여 데이터베이스의 구축, 분석을 위한 DDL을 생성한다.

2) 입력 프로세서 : 실제계의 특정 데이터가 입력으로 들어온 경우에 제한 조건, 일치성 등을 검사하고, DML을 생성한다.

3) 데이터베이스 프로세서 : 언어 프로세서와 입력 프로세서에 의해 얻어진 정보를 이용하여 데이터베이스를 작성, 수정, 분석한다.

Ⅲ. BIS(Back-end Information System)

BIS는 그림 3과 같이 FLA에 의해 구축된 데이터베이스에 대하여 실제 데이터들을 입출력하는 시스템으로 데이터 입력을 위한 입력 처리기(Input Processor), 데이터 삭제를 위한 삭제 처리기>Delete Processor), 데이터의 갱신을 위한 변경 처리기(Replace Processor), 그리고 데이터베이스로부터의 정보 추출을 위한 질의어 처리기(Query Processor)들로 구성된다. 또한 데이터베이스 문서 생성기(Document Generator)도 BIS의 프로세서인데, 문서 생성기의 하위 프로세서

로는 데이터베이스들의 객체 이름을 추출하는 NS(Name Seletion), 데이터베이스로부터 정형화된 입력 문장들을 추출하는 FS(Formatted Statement), 객체 이름들의 계층적 구조를 주어진 관계에 의하여 추출하는 SC(Structure Chart), 그리고 SC에 의한 출력을 구조도 형태의 도표로 나타내주는 EP(Extended Picture)등이 포함된다.

BIS의 각 부프로세서들은 FLA로부터 메타 정의에 관한 여러 정보들을 화일을 통하여 받아들이는데 이를 위하여 BIS의 모든 원시 프로그램들은 새로운 메타 정의를 처리할 때마다 매번 다시 번역되어야 한다. 본 연구에서는 새로운 메타 정의가 처리될 때마다 생성되는 각 프로세서의 수행 가능 화일들을 한 디렉터리에 모아둠으로써 임의의 한 사용자가 목적 메타 시스템의 path를 지정함으로써 하나의 메타 시스템에만 접근 가능하도록 하였다.

이 처리기의 입력은 메타 정의로부터 입력을 추출한 entity table, attribute table, text table, statement table, constant table 등이다.

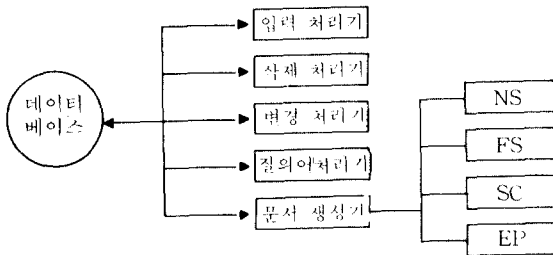


그림 3.

3.1 데이터 입력 처리기(Input Processor)

FLA에 의하여 구축된 데이터베이스에 대한 입력은 입력 처리기에 의하여 이루어진다. 입력 처리기에는 목적 데이터베이스의 이름이 주어져야 하며, 화일 이름이나 실제 IP 입력 문장이 주어져야 한다.

입력 처리기의 수행 중에는 여러 메시지와 함께 현재 입력이 목적 데이터베이스의 tuple로 변환된 결과, 그리고 객체명을 알 수 없는

데이터인 unknown instance들에 대한 처리 결과가 출력된다. IP의 처리가 끝난 후에는 unknown instance들의 리스트를 최종적으로 출력함으로써 사용자가 이름을 정의할 수 있도록 유도한다.

3.2 데이터 삭제 처리기(Delete Processor)

삭제 처리기는 데이터베이스에 저장된 정보 가운데 원하는 정보를 삭제하는 프로세서이다. 삭제 처리기에 의한 정보의 삭제는 해당 object의 삭제 및 이에 관련된 relationship을 삭제하는 경우와 관련된 relationship 만을 삭제하는 경우로 구분할 수 있으며 두 경우 모두 predicate를 사용하여 삭제의 범위를 제한할 수 있다. Predicate는 AND, OR, NOT, 좌괄호, 우괄호 등으로 구성된 논리식으로 이들 연산자간에는 미리 정의된 우선 순위가 적용된다.

삭제된 결과는 사용자에게 확인을 시켜주기 위해 화면에 출력하며 부적절한 입력이 들어온 경우에는 이에 상응하는 오류 메시지를 화면으로 보여 준다. 삭제 처리기의 사용은 다른 프로세서와 달리 매우 심각한 결과를 초래할 수 있으므로 wild character의 사용을 제한하는 부분이 존재한다.

3.3 변경 처리기(Replace Processor)

데이터베이스에 대한 변경 작업은 메타 시스템의 특성상 객체 이름에 대한 변경, 속성 값에 대한 변경, 텍스트 내용에 대한 변경, statement의 변형이라고 하는 네가지 기능을 필요로 하지만 속성 값에 대한 변경 작업은 IP에서도 이루어질 수 있기 때문에 RP에서는 이들에 대한 변경 기능을 제외하였다. 또한 statement에 의한 변경은 무결성의 문제가 많은 문제를 야기시키기 때문에 속성과 텍스트의 경우와 마찬가지로 개별적인 statement의 변경 기능은 수용하지 않았다.

따라서 본 연구에서는 객체의 이름만을 변경시키는 기능을 수행하며 객체의 이름을 변경하는 과정에서 referential integrity를 만족시키기 위하여 관계 relation의 내용에 대해서도 수정

작업을 수행한다.

변경 처리기의 사용은 주로 틀린 글자의 수정이나 의미있는 이름으로의 변환을 위해서 사용하게 되는데 이 프로세서도 사용자가 변경시키고자 하는 객체의 이름과 변경할 이름을 입력으로 주도록 되어 있다.

3.4 질의어 처리기(Query Processor)

질의어 처리기는 데이터베이스에 저장된 정보를 검색하는 프로세서이다. QP는 FLA로부터 생성된 정보를 이용하여 데이터베이스를 검색하며 목적 언어로 작성된 정보의 일치성 및 완전성을 분석한다. 검색을 할 때 predicate를 사용하여 검색하고자하는 범위를 축소할 수 있다.

object들간의 관계를 이용하여 검색을 하고자 할 때는 메타 언어에서 정의된 statement를 이용하며, object간 직접적인 관계 뿐만 아니라, 간접적인 관계를 이용하여 검색 가능하다. QP의 처리 결과로 데이터베이스에 저장된 임시 relation 들은 나중 검색을 위하여 사용이 가능하며, 임시 relation 간에 연산("+" : 합집합, "-" : 차집합, "*" : 교집합, "unary -" : 보집합)을 수행할 수 있다.

3.5 문서 생성기(Document Generator)

이 프로세서는 데이터베이스에 저장된 정보를 총괄적으로 추출하고자하는 경우에 사용될 수 있으며 이는 NS, FS, SC, EP 등으로 나뉘어진다.

3.5.1 NS(Name Selection)

이 프로세서는 데이터베이스에 저장되어 있는 객체 이름들을 객체 형과 함께 출력시켜 준다. NS의 입력에서 database name은 목적 데이터베이스의 이름으로, 새로운 이름이 주어지지 않는 경우에는 시스템에 미리 정의된 데이터베이스 이름이 주어진다. NS의 입력인 object type은 추출하고자 하는 객체의 형을 의미하며, object type을 명시하지 않은 경우에는 모든 객체 형에 대하여 출력을 생성한다.

NS의 출력은 사용자에게 의하여 특정한 화일에 저장될 수 있는데 그렇지 않을 경우에는 임시 출력 화일에 저장된다.

3.5.2 FS(Formatted Statement)

FS는 NS와는 반대로 데이터베이스에 있는 내용으로부터 원래의 입력 문장들을 추출하는 프로세서로 데이터베이스의 내용이 사용자가 의도한 것과 같을가를 감지하기 위해 사용될 수 있다. 또한 FS의 출력은 원래 입력과 달리 상당히 정형화된 형태로 추출되므로 그 자체가 하나의 보고서 형식을 갖는다.

FS는 데이터베이스로부터 입력 문장을 추출하지만 원래 입력 문장과 완전히 일치하지는 않는다. 그 이유는 원래의 입력 문장들이 데이터베이스에 입력되는 과정에서 semantics 분석을 통해서 원래 입력 문장에 의하여 함축된 의미는 구체적인 데이터베이스 입력으로 변환되므로 데이터베이스로부터의 FS 결과는 원래의 입력 문장보다 좀 더 자세한 정보를 포함하기 때문이다.

FS의 경우에는 NS와는 달리 하나의 객체 이름을 입력으로 줄으로써 그 객체의 이름에 대한 정보만을 추출할 수 있다. FS의 출력 결과는 NS와 마찬가지로 임시 출력 화일에 저장된다.

3.5.3 SC(Structure Chart)

데이터베이스의 관계 relation에 저장된 정보는 statement 입장에서 볼 때 목구조의 의미를 갖게 된다. 이러한 목구조는 데이터베이스에 저장된 정보의 분석을 위해서 유용한 역할을 할 수 있는데 SC 프로세서는 이러한 목구조의 보고서를 생성해내는 프로세서이다. SC는 사용자로부터 관계있는 statement 들을 입력으로 받아 그 statement 들을 트래버스하면서 인덴트된 형식의 정보를 출력한다.

SC의 주요 입력으로는 데이터베이스 이름과 statement이며 선택적 입력으로 목구조의 루트에 해당하는 객체 이름이 주어질 수 있다. SC는 주어진 statement로부터 관계 relation과 part

에 관한 정보를 추출하여 데이터베이스에 대하여 pre order 검색을 수행하며 출력 리스트를 위한 자료 구조를 구성한다. 이러한 자료 구조를 구성한 후 임시 출력 화일에 저장된다.

3.4.5 EP(Extended Picture)

EP는 SC에 의하여 출력되는 정보를 도표 형식으로 나타내는 프로세서로 SC와 같은 형태의 입력을 받아들이며 EP의 작업 방법과 자료 구조는 SC와 동일하다.

IV. 결 론

현재 사용되는 개발 방법론의 문제점을 해결하기 위한 효율적인 소프트웨어 개발환경 구축은 소프트웨어 개발 생명 주기의 각 단계에서 필요한 방법론을 지원하는 자동적인 방법론 지원 시스템을 통해 효과적으로 수행될 수 있다.

본 논문에서는 소프트웨어 개발 방법론 지원 시스템을 2 level 접근 방식을 이용하여 설계하였다.

FLA를 원하는 시스템의 개념 모델과 언어를 자유롭게 선택하여 설계할 수 있으므로, 확장성, 융통성, 이식성 등이 뛰어나고 이를 BIS와 통합시키면 소프트웨어 생명주기 초기 단계에 응용되어 개발 시간의 단축, 소프트웨어의 신뢰도 향상에 기여할 것이다.

향후 연구 과제로는 ER 다이어그램의 생성과 분석을 가능하게 하는 그래픽 인터페이스의 추가와 요구 사항 만큼 가지고 빠른 시제품화가 가능한 연구, 그리고 한글 환경 하에서 사용할 수 있도록 하는 연구가 남아 있다.

부 록

SYSTEM system_name ;
 LANGUAGE_NAME language_name ;
 ANALYZER analyzer_name ;
 ENTITY entity_type_name :

SUBSETS entity_type_name_list ;
 DOCUMENTATION ;
 document_text ;
 ATTRIBUTE attribute_name ;
 APPLIES {entity_type_name_list ;
 ALL
 ALLBUT entity_type_name_list ;
 VALUES value_type ;
 DOCUMENTATION ;
 document_text ;
 TEXT text_name ;
 APPLIES {entity_type_name_list ;
 ALL
 ALLBUT entity_type_name_list ;
 DOCUMENTATION ;
 document_text ;
 RELATIONSHIP relationship_name ;
 DOCUMENTATION ;
 document_text ;
 PARTS part_name_list ;
 COMBINATION part_name [thing]
 WITH part_name [thing]
 ... ;
 CONNECTIVITY {[ONE MANY]
 part_name,; ... ;
 STATEMENT statement_name ;
 DOCUMENTATION ;
 document_text ;
 USED relationship_name part_name ;
 FORM {statement_string part_name ;
 ... ;
 where "thing" is
 {entity_name_list ALL ;
 ALLBUT entity_name_list ;
 VALUE_FOR value_name ;

참 고 문 헌

1. A.J. Wasserman, "Information System Design Methodology," Journal of the American Society for Information Science, Jun., 1980.
2. C.P. Gane and T. Sarson, "Structured System Analysis : tools and techniques," Prentice Hall, 1979.
3. D. Teichroew, P.Macasovic, E.A. Hershey III, and Y.Yamamoto, "Application of the entity-relationship approach to information processing systems modeling," Proc. Int'l Conf. Entity Relationship Approach to System Analysis and Design, 1979.
4. D. Teichroew and E.A. Hershey III, "PSL / PSA : A

- Computer aided technique for structure documentation and analysis of information processing system," IEEE Trans. on Software Engineering, Vol. SE-3, No.1, Jan, 1977.
5. ISDOS, "An introduction to the use of information system language definition manager(ISLDM) and system encyclopedia manager(SEM)," ISDOS Ref. #Mo 320 0, ISDOS Inc., Sep, 1981.
 6. J. Demetrovics, E.Knuth and P.Rado, "Specification meta systems," IEEE, COMPUTER, May, pp29-35, 1982.
 7. M. Hammer and G Ruth, "Automating the software development process," Research Directions in Software Technology, Peter Wegner, ed., MIT press, Cambridge, Mass., and, London, England, 1979.
 8. P.P. Chen, "The entity-relationship model: Toward a unified view of data," ACM. Trans, on Database System, Vol. 1, No. 1, Mar., 1976.
 9. P.P. Chen, "The entity relationship approach to logical database design," Q.E.D. Information Sciences Inc., 1977.
 10. P.P. Chen, "ER: A historical perspective and future direction," ER approach to SE., North Holland, 1983.
 11. R. Fairley, "Software engineering concepts," McGraw-hill, 1985.
 12. S.C. Johnson, "YACC-Yet Another Compiler-Compiler," Bell laboratories, 1975.
 13. T.E. Bell, D.C. Bixler and M.E. Dyer, "An approach to computer aided software requirements engineering," IEEE Trans, on Software Engineering, Vol. SE-3, No. 1, Jan., 1977.
 14. Chun You Seek, "A life-cycle support system for real-time system," Ph. D.Dissertation, Seoul National University, 1986.



李淑姬(Sook Hee Lee) 정희필
1979년 : 순명여자대학교 문과대학 졸업
1982년 8월 : 농성대학교 경영대학원(간
사원장) 입학
1988년 2월 : 성균관대학교 대학원 박사
과정 수료(전학전공)
1987년 3월~현재 : 농성여자대학 전자
계산학과 조교수