

모델관리시스템의 諸 接近法에 대한 比較研究

김유일(부산대학교 경영학과 교수)
김진수(부산대학교 경영학과 조교수)
정대울(동명전문대학 경영정보과 전임강사)

제 1 장 서 론

모델링의 영역은 매우 넓고, 다양한 패러다임이 존재하기 때문에 각 영역마다 다양한 형태의 모델이 개발되고 사용되어 왔으며, 이를 관리할 수 있는 매카니즘이나 컴퓨터지원 도구의 개발에 많은 관심을 보이고 있다. 이 중 의사결정지원을 위한 수리적 모델링 영역에서 모델링 자원을 효과적으로 관리하려는 모델관리시스템(Model Management Systems : MMS)에 대한 연구가 최근 MIS(DSS)의 중요 연구분야로 부상하고 있다. MMS는 모델베이스내의 모델을 효과적으로 관리하는 기능 즉, 모델의 개발 및 저장, 조작 및 접근, 통제하는 것을 용이하게 하는 소프트웨어이다. 그런데 지금까지 MMS에 대한 많은 연구들이 이루어 졌으나 이들 연구 결과들에 대한 종합적 분석을 위한 프레임워크가 제시되지 않고 있다.

이에 본 연구는 MMS에 대한 선행연구들을 체계적으로 분류하고, 이들 기존의 연구들을 체계적으로 분석하기 위한 하나의 프레임워크를 개발함으로써 향후 MMS에 대한 연구방향을 제시하는 데 그 근본 목적이 있다. 이를 위해서 본 연구에서는 기존의 MMS에 대한 연구들을 ①수리적 모델링언어지향 ②데이터베이스지향 ③지식중심 ④그래프중심 ⑤구조적 모델링 ⑥객체지향으로 분류하고 있다.

MMS의 비교를 위해서 두 가지 차원(모델링언어와 MMS의 기능)을 제시하며, 여기에 부가하여 모델링언어는 또 다시 두 가지 측면(모델링언어 특성과 문법)에서 비교되어진다. 먼저 특성비교를 위한 기준으로 ①일반성 ②비절차성 ③노동집약성 ④구체성 ⑤개념적 모듈성 ⑥모델독립성 ⑦언어의 수준 ⑧메타언어 등을 제시하고, 각 접근법에서 제시된 모델링언어들이 이들 기준을 어느 정도 충족시키는지 서로 비교한다. 다음으로 모델링언어의 문법분석을 위해서 각 언어의 상위수준문법(high-level syntax)을 도출하고 이를 통하여 각 모델링언어에서 모델정의 및 조작을 위해 사용한 ①객체타입 ②인덱스 ③모델메소드(model method) 등을 비교한다. 이를 통해서 모델링언어가 갖추어야 할 특성과 모델링언어 설계시 고려되어야 할 사항들을 체크할 수 있게 한다.

또한 MMS의 기능비교를 위해서 모델링 수명주기(①문제정의 ②모델수립 ③해산출 ④결과해석)를 도입하였다. 즉 MMS가 모델링 수명주기의 각 단계를 어떻게, 어느 정도 지원하는가를 비교함으로써 모델링활동의 각 단계를 지원하는 데 유용한 대안들과 연구과제를 제시한다.

본 연구에서 제시하는 이들 분류기준과 평가척도들은 MMS의 바람직한 특성들을 추출하는 데 기여할 뿐만 아니라, 기존 또는 향후 개발된 MMS와 모델링언어를 분류하고,

비교/평가하는 데 있어 하나의 중요한 프레임워크가 될 수 있다는 데 그 의의가 있다.

제 2 장 모델관리시스템에 대한 기존 연구

모델관리에 대한 연구를 체계적으로 정리하기 위해서는 기존 연구들의 공통점을 모아 몇 가지의 접근법(approach)으로 대별하고, 또한 각 접근법내에서는 몇 가지의 프레임워크(formework)으로 분류하는 것이 필요하다. Chung & O'Keefe(1992)는 모델관리에 대한 기본적인 접근법으로 ①모델/모델링의 인식 접근법 ②데이터-모델 유추 접근법 ③지식중심의 접근법 ④모델표현 접근법 ⑤기타 접근법으로 분류하고 있으며, Bharadwaj et.al.(1992)는 ①대수적 모델링 언어 ②데이터베이스 접근법 ③그래프중심의 접근법 ④지식중심 접근법으로 분류하고 있다.

<표 2-1> 모델관리를 위한 접근법

| 기본적 접근법 | 프레임워크 | 학문적 이론 및 영역 |
|--------------|---------------------------------------------------------|---------------------------------|
| 수리적 모델링언어 지향 | 매트릭스 생성기 대수적 표현 B-S(Block-Schematic)표현 | 경영과학/OR의 수리적 모델링 |
| 데이터베이스 지향 | 실체-관계(E-R)모델 네트워크 관계형 | 데이터베이스관리의 개념적/논리적모델링 |
| 지식 중심 | 논리중심 (일차술어연산 / 생성규칙) 의미네트/프레임 | 인공지능의 지식표현 |
| 그래프 중심 | 모델그래프 논리 그래프 유속성그래프 (attribute graph) | 이산수학의 그래피이론 네트워크이론 |
| 구조적 | 구조적 모델링 | 이산수학의 그래피이론과 소프트웨어공학의 구조적 기법 |
| 객체지향/시스템 | 모델추상화 객체지향 DB 객체지향 KB 모델가시화 DB/KB 통합 시스템 | 객체지향개념 시스템개념 |

본 연구에서는 모델표현방법에 주목하여 Bharadwaj et.al.(1992)의 분류기준을 약간 수정하여 대수적 모델링을 전통적 MS/OR의 수리적 모델링 접근법 내의 한 프레임워크로 분류했으며, 객체지향/시스템 접근법을 추가하고, 구조적 모델링 프레임워크를 그래프 중심접근법에서 분리하여 하나의 독립된 접근법으로 보고자 한다. 왜냐하면, 구조적 모델링 접근법이 비록 이산수학의 그래프 이론에 그 기반을 두고 있으나, SML(Structured Modeling Language)(Geoffion 1987, 1992b)이라는 패러다임 독립의 모델링 언어를 가지고 있으며, 이것은 소프트웨어 공학의 모듈개념에 그 기반을 두고

있다고 볼 수 있다. 또한 최근 소프트웨어공학의 여러분야에 연구의 주류를 이루는 객체지향 접근법을 모델관리에 도입하려는 연구가 1990년대 이후의 주요 패러다임으로 부상하고 있기 때문이다. 각 접근법별 세부적인 프레임워크와 그 기반이 되는 학문적 이론 및 영역을 제시하면 (표 2-1)과 같다.

여기서 모델링 접근법(modeling approach)은 MMS을 설계하기 위한 개념을 가져온 학문이론 또는 분야를 말한다. 예를 들어 모델관리를 위한 그래프중심의 접근법은 수리모델을 표현하기 위하여 그래프이론의 개념에 근거를 두고 있다. 그리고 모델링 프레임워크(modeling framework)은 모델링 접근법 내의 특정 형태이다. 예를 들어 일차술어연산(first order predicate calculus)은 지식중심의 접근법 내의 특정 프레임워크이다.

1. 수리적 모델링언어 지향 접근법

컴퓨터를 통한 모델중심의 의사결정에 대한 중요성이 증가됨에따라 많은 모델링언어(특히 LP 모델링언어)들이 개발되기 시작하였다. LP 모델링언어로서 가장 먼저 나타난 것은 아마 1963년에 개발된 MAGEN(이후에 OMNI로 발전)일 것이며, 그 직후 GAMMA(이 후에 GAMMA/2000으로 발전)와 DATAFORM 등이 계속적으로 개발되었다. 1960년대와 1970년대에 개발된 이 들 시스템은 보고서 작성기(reprot writers)나 매트릭스생성기(matrix generators)를 갖춘 정도이며, 특정 컴퓨팅환경에서만 작동되고 사용자에게 상당한 프로그래밍 기술을 요구하였으므로 진정한 모델링언어로는 볼 수 없다. 모델 타당성 체크, 유지보수의 용이, 사용의 편리성 등의 기능을 제대로 갖춘 진정한 의미의 대수적 모델링 시스템을 최초로 시도한 것은 아마 1970년대 후반에 나온 GAMS 일 것이다. 1970년대 후반 이후에 나온 주요 대수적 모델링 시스템 또는 언어와 이들에 대한 연구자를 요약하면 (표 2-2)와 같다.

<표 2-2> 대수적 모델링시스템 또는 언어의 선행연구

| 프레임워크 | 모델링시스템 또는 언어 | 연구자 또는 기관 |
|-------------|---------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| 매트릭스 생성기 | OMNI, MAGEN DATAFORM GAMMA/2000 | Haverly Systems(1976) Ketrion Management Science(1987) Bonner and Moore Management Science(1989) |
| 대수적 표 현 | GAMS MGG LINDO LPL GNGEN AMPL MODELER | Bisschop & Meeraus(1982), Simmons(1986) Schrage(1987) Hürlimann & Kohlas(1988) Forster(1988) Fourer et.al.(1990) Greenberg(1992) |
| B-S표현 | PAM MIMI MathPro | Welch, Jr. (1987) Chesapeake Decision Sciences(1988) MathPro, Inc.(1990) |

1) 매트릭스 생성기 언어

매트릭스 생성기 시스템(matrix generator systems)은 많은 강점을 가지고 있으며 오랫동안 성공적으로 사용되어 왔다. 이에 해당하는 시스템으로는 OMNI, MAGEN,

DATAFORM, GAMMA 등을 들 수 있다. 매트릭스 생성기 언어는 반복문(loops), 조건문 또는 GOTO문 등과 같은 절차적 성격을 지닌 문장을 요하지 않는 비절차적 언어의 특성을 상당히 지니고 있어 프로그래밍 효율을 향상시킬 수 있다. 그러나 모델표현에 있어 MPS와 같은 최적화기(optimizer)에서 사용하는 표현과 거의 가깝다. LP 매트릭스 생성기의 초창기에 속하는 것들은 MPS 문장을 직접 작성하도록 되어 있어 집합(sets)과 데이터 테이블과 같은 객체를 조작하여 MPS 포맷에 맞도록 데이터 리스트를 만든다. 이러한 MPS 지향적인 표현은 문제분해(problem decomposition)나 개념적 모듈성(conceptual modularity)의 여러 측면들을 직접적으로 지원하지 못한다. 또한 비선형 모델을 표현할 수 없으므로 한정된 일반성(limited generality)을 지니며, 모델수립자가 직접 모델내의 각 원소를 MPS 문장의 열과 행으로 전환해야 하므로 모델링언어 중 에서 구체성(concreteness)이 가장 낮다(Murphy, Stohr, and Asthana, 1992, p.974).

2) 대수적 표현

매트릭스 생성기 언어의 단점을 보완하면서 모델수립자의 대수적 관점(특히 MS에 정통한 사용자)에 초점을 두고 비절차적인 서술문 형태로의 모델표현을 시도한 최초의 시스템은 아마 GAMS일 것이다. GAMS는 수리적 모델을 일련의 명령어 집합으로 표현하므로써 대규모의 복잡한 모델을 표현하기 쉽게 만들어져 있다.

GAMS와 같은 초기 모델링 시스템의 가장 큰 장점은 기호적 모델표현(modeler's form)을 컴퓨터가 읽을 수 있는 형태(algorithms's form)로 전환하는 데 있어 특수한 매트릭스 생성기 프로그램을 작성할 필요가 없다는 점이다. 그러나 GAMS는 진정한 의미에서의 MMS으로 보기는 어렵다. 왜냐하면 사용자는 모델의 모든 측면을 명세해야 한다. 또한 데이터가 모델내에 명세되어야 하므로 모델-데이터 독립성이 없으며 수정과 유지보수를 어렵게 한다. GAMS와 유사한 목적과 설계로 된 최근의 모델링언어로 AMPL을 들 수 있다. AMPL의 표현법은 GAMS와 유사하나 데이터 값이 집합, 파라미터, 변수의 명세와 결합되지 않고 독립된 섹션에 명세할 수 있어 유지보수를 용이하게 한다.

대수적 표현을 기반으로 하는 시스템에는 GAMS와 AMPL 외에도 (표 2-2)에서 보는 바와 같이 MGG, MODELER, LPL, LINDO 등 대부분의 모델링 시스템이 이 범주에 속한다. 대수적 표현은 모델-데이터, 모델-알고리즘의 독립성을 제공할 수 있는 여지를 준다. 또한 모델표현의 일반성과 비절차성 및 간결성을 지닌다. 반면에 대수적 표기법은 계층적 분해나 상향적 통합을 통한 개념적 모듈성을 제공하기가 어렵다. 더우기 문제의 실물의 구조가 명기되지 않으므로 구체성이 낮고 추상성이 높다. 따라서 대수적 모델링 시스템은 MS 전문가용으로 적합하다.

3) B-S(Block-Schematic) 표현

LP 매트릭스를 블럭 지향적 관점(block-oriented view)에서 보면 문제구조를 이해하기 쉽다. 대부분의 LP매트릭스는 영(0)으로만 된 블럭이 군데군데 있는 영(0)이 아닌 데이터구조의 블럭으로 구성되어 있으므로 블럭 지향적인 모델수립이 가능하다. 따라서 B-S(Block-Schematic) 시스템은 모델수립자가 대규모의 2차원적인 LP 매트릭스에 데이터블럭을 쉽게 넣을 수 있게 하는 언어 정의문을 제공해준다. (표 2-2)의 PAM, MIMI, MathPro 등이 B-S표현을 지원할 수 있는 시스템이다.

MS 전문가들은 대수적 표현을 쉽게 이해할 수 있으나 실무자들은 주로 공정중심으로 모델을 보는 것이 훨씬 쉽다. B-S 표현법은 추상적인 대수적 관점(algebraic view)

과 공정중심의 활동성 관점(activity view) 간을 연결시켜 준다. 따라서 B-S표현법은 수정된 대수적 관점으로 생각할 수 있다. 즉 변수는 머리부분에 한번만 쓰고, 각 행은 어디에 그 변수가 나타나는가를 결정하면 된다.

또한 이러한 B-S접근법은 각 블럭을 LP매트릭스와 직결시킬 수 있어 다수의 속련된 LP모델수립자의 사고방식과 일치한다(Welch 1987). B-S표현법은 계층적 접근법과 고도의 추상화(abstraction)를 통하여 인지적 복잡성을 줄여준다. 그리고 B-S표현법은 절차적 언어라기 보다는 선언적 언어이기 때문에 초기의 매트릭스 생성기 시스템보다는 다소 덜 노동집약적일 것이다. 그러나 B-S표현법은 모델화될 시스템의 물리적 측면보다는 모델수립과정의 최종결과에 초점을 두므로 표현이 추상적이며, 단지 LP시스템에만 적용되므로 제한된 일반성을 지니고 있다.

2. 데이터베이스 지향 접근법

모델수립단계에서의 많은 량의 데이터 수집과 처리에 대한 필요성과 대규모 모델에서 나온 상당량의 결과해석에 대한 요구는 자연히 수리계획시스템(mathematical programming systems)과 DB 기술간의 통합에 대한 연구를 촉진시켰고 DB 기술에 대한 연구결과들을 모델관리에 도입하려는 일련의 연구들을 가져왔다(표 2-3).

모델관리에 DB 접근법을 시도하는 이들의 기본적인 생각은 특정의 데이터모델(관계형 데이터 모델, 네트워크형 데이터 모델, E-R 모델 등)을 사용함으로써 의사결정모델을 조직화할 수 있으며, 기존의 데이터베이스관리시스템(DBMS)을 이용하여 세부적인 물리적 모델베이스 구축과 조작을 용이하게 할 수 있다는 것이다.

<표 2-3> 데이터베이스 지향 MMS의 선행연구

| 프레임워크 | MMS 또는 언어 | 연구자 |
|-------|--------------------------------------|-----------------------------------------------------------------------|
| 네트워크 | nn GMMS GXMP | Stohr & Tanniru(1980) Konsynski(1980) Doik(1986a) |
| 실체-관계 | ni ni WXDSS ERLMP, ni | Elam(1980) Blanning(1986) Chen, Y.S.(1988) Choobineth(1991b) |
| 관계형 | (TQL, MQL), ni (SQLMP), ni MOS | Blanning(1982, 1985, 1987) Choobineth(1991a) Lari & Nag(1993) |

* nn : no name, ni : not implemented.

1) 네트워크 데이터모델 프레임워크

네트워크 데이터모델(network data model)은 데이터를 관련 레코드들의 네트워크로 표현하고, 각 레코드 타입은 네트워크내의 하나의 마디(node)가 된다. 마디 A에서 B로의 화살표는 A와 B간의 관계가 1:m관계(one-to-many association)임을 의미한다. Stohr & Tanniru(1980)는 네트워크 DBMS를 사용하여 모델의 저장과 실행을 위한 시스템을 구현하였다. Doik(1986a)는 Konsynski(1980)의 GMMS의 개념적 프레임워크에 근거하여 CODASYL DBMS를 사용한 GXMP(generalized experimental mathematical programming

system)라는 프로토타입 시스템을 구현하였다.

2) 실체-관계 프레임워크

P.P.S. Chen(1976)의 E-R접근법을 모델관리에 도입할 경우 모델화될 실체(entity), 정보 또는 데이터가 기억되는 실체 그리고 이들 간의 관계를 표현하는 데 아주 유용한 수단을 제공해줄 수 있으며, 개념적 수준에서의 데이터와 모델표현능력을 향상시킬 수 있다. 뿐만 아니라 모델간 인터페이스 관계를 명시적으로 표현할 수 있어 DB 및 모델베이스의 조직화에 도움을 준다.

E-R(Entity-Relationship)모델개념을 모델관리에 최초로 적용한 사람은 아마 Elam(1980)일 것이다. 그는 실체와 관계의 속성(attribute)을 통제가능(controllable) 속성과 통제불가능(uncontrollable) 속성으로 분류했으며, 통제가능 속성은 사용자, DB, 또는 선행모델의 출력으로 제공되어지며, 반면 통제불가능 속성은 현행모델의 출력으로 보았다. Blanning(1986)은 모델베이스 구축을 위한 개념적 틀을 제시하는 데 있어, 기존의 DB에서 사용되는 E-R접근법과 어떻게 결합될 수 있는가를 보이기 위해 순환적 모델뱅크(cyclic model bank)를 위한 E-R 다이어그램을 제시하였다. Choobineth(1991b)는 선형수리계획법을 위한 E-R모델(ERLMP) 및 다이어그램 기법(ERLMD)을 개발하였다.

3) 관계형 프레임워크

Coad(1970)의 관계형 데이터 모델을 모델관리에 도입할 경우 다양한 데이터 원천(data source)에의 접근을 가능하게 하므로써 신축성있게 DB와 연결이 가능하게 하며, 모델과 데이터의 조작을 용이하게 한다. Blanning(1982,1985)은 하나의 모델을 한 릴레이션(relation)으로 보고 모델의 입력과 출력요소를 릴레이션의 속성으로 하였다. 입력속성의 집합은 모델 릴레이션의 키(key)가 되며 이것이 출력속성(non-key)을 결정한다. 그리고 그는 모델조작을 위하여 실행(execution), 최적화, 민감도분석을 위한 세 개의 대수적 연산자를 정의하고 MQL(Model Query Language)와 TQL(Tabular Query Language)를 제시하였다. 그리고 모델뱅크의 조인(join)을 수행하기 위한 ETQL(Extended TQL)을 제시하였다. 이와 유사한 연구로서 Choobineth(1991a)의 SQLMP도 있다.

3. 지식중심 접근법

Elam & Konsynski(1987)는 MMS의 능력을 향상시키기 위하여 모델수립, 분석, 결과해석 분야에서 여러 가지의 인공지능기법의 적용이 가능함을 보이고 있다. 모델관리분야 중에서도 특히 이들 분야에서는 인간의 지적활동을 필요로 하므로, 이러한 분야에서 인간의 지적활동을 대신할 수 있는 MMS의 설계는 보다 더 지적 능력을 갖춘 MMS의 실현을 가능하게 한다. 이러한 인간의 지적활동을 대신하는 MMS를 개발한다는 것은 곧바로 지식기반의 전문가시스템(knowledge-based expert systems) 또는 지식시스템을 개발하는 것과 유사하다.

특히 지식표현방법은 모델링활동의 가치와 우선순위를 반영하고 있으며 시스템의 성과(performance)와 다이얼로그에 영향을 미친다. 모델관리와 관련하여 제기되는 지식표현의 문제는 모델을 지식베이스내에서 어떻게 표현할 것인가와 관련이 되어 있으며 효율적 지식표현방법의 개발이 시스템 개발활동에 지대한 영향을 미치므로 모델관

리에 대한 인공지능기법의 도입에 대한 기존의 대부분의 연구노력들이 여기에 초점을 두고 있다. 지금까지 연구되어진 결과들을 종합해 보면 (표 2-4)과 같이 논리중심의 일차술어연산과 생성규칙을 사용한 방법과 의미네트와 프레임을 사용한 방식으로 대별해 볼 수 있다.

이러한 지식중심의 접근법을 모델관리에 적용할 경우 다음과 같은 점을 기대할 수 있다. 첫째, 자동화된 모델수립이다. 사용자가 수작업으로 주어진 문제에 대한 수리적 모델을 수립하는 데는 상당한 지식과 노력이 요구되므로 지식중심의 MMS(모델링 전문가시스템)를 개발함으로써 사용자는 모델링의 기술적 측면에 대한 노력에서 해방될 수 있다. 그리고 모델수립시 모델수립전문가(expert intermediary)의 개입이 없어도 되므로 의사결정자와 모델수립전문가와의 잘못된 의사소통에 따른 위험을 줄일 수 있다.

<표 2-4> 모델관리 지식표현과 지식중심의 MMS를 위한 선행연구

| 프레임워크 | MMS 또는 언어 | 연구자 |
|-------------------------------------|----------------------------|-----------------------------|
| 논리(Logic) (일차술어연산 또는 생성규칙) | ni | Elam, et.al.(1980) |
| | ni | Bonczek, et.al.(1981) |
| | ni | Dutta & Basu(1984) |
| | ASC | Sivasankaran & Jarke(1985) |
| | nn | Murphy & Stohr(1986) |
| | nn | Lee & Miller(1986) |
| | nn | Sklar, et.al.(1987) |
| | LPFORM | Ma, et.al(1989) |
| | TIMMS | Liang(1988) |
| | ni | Shaw, et.al.(1988) |
| | nn | Mannino et.al.(1990) |
| | PDM(PM*) | Krishnan(1989,1990,1991) |
| | AIMM | Liu, et.al.(1990) |
| | ni | Choobineth(1991a) |
| TEFA | Bhargava & Kimbrough(1993) | |
| nn | Mitra, et.al.(1994) | |
| 의미네트/ 프레임 | ni | Elam, et.al.(1980) |
| | SPMMS | Applicate, et.al(1986) |
| | nn | Binbasioglu & Jarke(1986) |
| | GUTS | Fedorowicz & Williams(1986) |
| | LPFORM | Ma, et.al(1989) |
| | nn | Sklar, et.al.(1987) |
| | AIMM | Liu, et.al.(1990) |
| | PDM | Krishnan(1991) |
| | nn | Pracht(1990) |

* nn : no name, ni : not implemented.

둘째, 지식중심의 MMS가 사용자의 질의에 융통성있게 답할 수 있는 능력이다. MMS는 모델베이스내의 있는 모델을 검색하여ダイナ믹하게 의사결정을 지원할 수 있어야 한다. 이를 위해서는 MS/OR전문가의 전문지식을 시스템내에 담고있어야 한다. 이러한 전문지식과 전문가가 사용하는 추론방법을 추출하기 위해서는 인공지능기법을 적용할 수 있다.

세째, 사용자의 훈련도구로 사용할 수 있는 모델링 전문가시스템을 구축할 수 있다. 이 전문가시스템은 사용자가 모델을 수립, 해석, 이해하는 것을 돕는 설명기구

(explanation facilities)를 제공할 수 있어야 한다.

네째, 지식중심 MMS의 지식베이스는 모델링에 대한 정보를 명기할 수 있으며, 언제든지 접근하여 지식의 추가, 갱신, 삭제 등과 저장된 지식의 일관성을 체크할 수 있는 매카니즘을 가지고 있어 유지보수와 확장(지식보강활동)이 용이하다.

1) 논리적 프레임워크

논리적 지식표현방식은 의사결정환경에서 가장 많이 사용되는 지식표현방식으로 일차술어연산이나 생성규칙을 사용한다.

일차술어연산은 특정 영역에 관한 진술(statement)을 형성하기 위하여 공식적 언어와 일련의 추론규칙을 생성하는 일반적인 논리시스템이다. 술어연산은 특정 응용분야에서 그 분야에 대한 지식을 WFF(well-formed formula)형식으로 표현할 수 있게 해주므로 그 응용분야에 대해 표현된 모든 WFF들의 집합은 그 응용을 위한 공리집합이 될 수 있다. 이 경우 특정 응용분야에 대한 지식을 공리(또는 WFF)로 나타낼 때 이들간의 일관성(consistency)과 비중복성을 유지하는 것이 중요하다. 이러한 일관성 있는 공리집합이 주어진 경우 이 집합으로부터 논리적으로 타당한 추론을 하는데 리졸루션(resolution)원리가 적용된다.

생성시스템이나 규칙중심의 시스템에서 규칙은 IF<antecedents> THEN<consequents> 형식으로 표시되며, 추론은 목표(goal)나 결론에 도달할 때 까지 순환적으로 "antecedents"나 "consequents"를 검토하므로써 전방향(forward) 또는 역방향(backward)으로 이루어진다. 이렇게 하므로써 시스템은 인간의 추론과정을 흉내낸다.

MMS가 DB의 데이터를 검색하고, 모델베이스의 모델을 검색하며, 필요시 모델의 합성과 분해에 의해 사용자가 요구한 정보를 제공하는 기능을 생각하면, 논리적 지식표현방식은 모델관리자들로 부터 관심을 끌기에 충분하다. 또한 논리적 지식표현방식은 확고한 추론방법을 가지고 있고, 데이터와 모델링지식을 하나의 체계안에서 함께 표현할 수 있으며, Prolog와 같은 시스템의 구현을 용이하게 해주는 강력한 프로그래밍 언어를 가지고 있다는 장점이 있다. 또한 리졸루션 추론기법을 적용할 경우 정보검색, 무결성(integrity)검토, 그리고 문제해결을 용이하게 한다.

따라서 (표 2-4)에서 보는 바와 같이 많은 연구들이 모델관리를 위한 지식표현을 위한 프레임으로 논리적 프레임워크를 채택하고 있다. 이들 중에서 일차술어연산을 사용한 연구로는 Bonczek, et.al.(1981), Dutta & Basu(1984), Sivasankaran & Jarke(1985), Murphy & Stohr(1986), Liang(1988), Shaw, et.al.(1988), Krishnan(1989,1990,1991), Choobineth(1991a), Bhargava & Kimbrough(1993) 등이며, 생성규칙을 사용한 연구로는

Elam, et.al.(1980), Sivasankaran & Jark(1985), Murphy & Stohr(1986), Lee & Miller(1986), Sklar, et.al.(1987), Ma, et.al.(1989), Liang(1988), Shaw, et.al.(1988), Mannino et.al.(1990), Krishnan(1989,1990,1991), Liu, et.al.(1990) 등이다.

이러한 논리중심의 지식표현 시스템들의 생명은 ① 지식베이스 내에 어떠한 지식들이 담겨져 있느냐와, ② 그러한 지식들이 어떤 식으로 조직화되어 있느냐에 달려있다. 따라서 논리중심의 지식표현방법이 모델관리에 유용하기 위해서는 지식베이스를 구조화하고, 이러한 지식베이스와 데이터관리요소들과의 통합능력이 요구된다.

2) 의미네트/프레임 프레임워크

의미네트웍(semantic network)에 의한 지식표현방식은 객체(object)와 이들 객체들 간의 관계(relationship)를 중시한다. 여기서 객체란 데이터와 절차(procedure)들의 특성을 합한 실체를 말하며 계산을 하거나 특정 부분의 상태를 저장하는 역할을 한다. 의미네트웍에 의한 지식표현은 객체들을 마디(node)와 화살표(arc)로 구성되는 有方向 그래프(direct graph)로 표시한다. 지식베이스내의 추론은 그대프내의 적절한 연결(link)를 따라서 이루어 진다.

추론능력면에서 볼 때 논리적 지식표현방법은 "IF A THEN B"와 같은 확정적 추론에 적합한 반면, 네트워크 지식표현법은 "A is like B except ----"와 같은 휴리스틱한 추론 및 탐색에 적합하다. 즉, 논리적 지식표현법은 휴리스틱한 추론이나 추상화(abstraction)개념을 적절하게 지원하지 못하는 반면 네트워크 지식표현법은 확정적 추론능력이 약하다. 그런데 모델링과 모델관리를 위해서는 이러한 두 종류의 추론이 모두 유용하므로 두 지식표현방법의 장점을 혼합한 혼합형 지식표현방법이 필요하다.

M. Minsky(1985)에 의해서 제시된 프레임에 의한 지식표현법은 여러가지의 지식표현 개념을 결합한 지식표현방법으로 프로토타입화된 개념(prototyped concept)이나 상동적 개념(stereotyped concept)을 및 상황을 표현하는 데 적합하다. 프레임이란 어떤 상황에 대한 일반적인 틀/framework)으로써 상황이 바뀔에 따라 그 상황에 잘 적합되도록 되어있다. 또한 프레임은 슬롯(slot)이라는 구조를 통하여 유사한 상황들을 동시에 표현할 수 있는 능력을 갖기 때문에 프레임에 의한 지식표현에는 어떤 슬롯들이 정의되느냐가 매우 중요한 문제이다.

프레임 시스템은 모델관리문제와 관련시켜 볼 때 다음과 같은 중요한 특성을 지니고 있어 MMS의 설계에 있어 유용한 수단이 될 수 있다(Applegate, et. al. 1986). 첫째, 단일 모델의 표현이 프레임을 사용하여 적절히 이루어질 수 있다면 사용자의 문제에 관한 기술과 실제 모델 및 데이터 사이의 연결을 쉽게 한다. 둘째, 프레임 시스템은 일단 표현된 모델의 관리를 원활히 할 수 있는 정보시스템구조를 제공해 줄 수 있는 장점이 있어, 프레임은 모델링환경을 위한 유용한 지식표현수단이 될 수 있다.

4. 그래프중심 접근법

수리적 모델을 하나 이상의 그래프 또는 이중그래프(digraph)로 표현하므로써 개념적 명료성과 프로그래밍 및 조작의 용이성을 기할 수 있다. 그래프는 모델의 의미를 담은 마디와 화살표로 구성되어 있다. 그래프에서 일반적으로 마디는 객체로 해석되며 유방향 화살표는 객체들 간의 의미적 관계로 여겨진다. 그래프를 통한 복잡한 수리적 관계의 표현은 분석가와 의사결정자 사이의 의사소통을 원활히 하는데 도움을 준다. 모델관리에 그래프이론을 도입한 연구자들 보면 (표 2-5)과 같다.

인간의 모델링과정은 주어진 문제의 초기상태(사용가능한 정보)를 원하는 최종상태(산출정보)로 바꾸는 방법을 찾기 위해 많은 수의 가능한 관계들을 조사하는 것으로 볼 수 있다. 따라서 모델링 지식표현은 방향을 가진 네트워크 그래프를 형성하는 과정으로 생각할 수 있다. Liang(1986, 1988)은 이 개념을 사용하여 모델그래프를 구성하므로써 모델관리를 할 수 있다고 생각했다. 모델의 기본적인 표현은 ①입력(input) ②출력(output) ③무결성(integrity) ④타당성(validity) ⑤계산을 위한 서브루틴(subroutine)으로 된 5개의 릴레이션(relation)을 이용한다. 모델수립을 위한 매카니즘은 DB와 모델베이스의 탐색을 통한 모델그래프를 만들므로써 이루어진다.

<표 2-5> 그래프중심 접근법의 선행연구

| 프레임워크 | MMS 또는 언어 | 연구자 |
|------------------------------|------------------------------------------|----------------------------------------------------------------------------|
| 모델 그래프 | TIMMS | Liang(1986,1988) |
| 논리 그래프 | ni | Kimbrough(1986) |
| 유속성그래프 (attributed graph) | GBMS Networks/SM (NETFORM) gLPS | Jones(1990,1991) Jones(1992) Glover(1990) Collaud & Boltuck(1994) |

* ni : not implemented.

모델그래프는 데이터를 나타내는 마디와 정보의 변환기능을 나타내는 화살표로 구성된다. 모델그래프가 수립되면 모델선택을 위한 추론전략이 적용된다. Liang(1988)은 추론전략으로 최적화전략(optimizing strategy)과 만족화전략(satisficing strategy)을 제시하고 있다. 만족화전략을 적용할 경우 모델평가함수의 개발이 시스템의 정보제공능력에 지대한 영향을 미친다. Liang & Jones(1988)는 모델평가함수개발을 위한 다양한 지침들을 제공해주고 있다.

그래프 접근법의 또 다른 연구로는 Kimbrough(1986)의 논리그래프(logic graph)를 들 수 있다. 그는 계획 및 정책분석에서 문제해결을 위해서 "논리모델(logic model)"을 위한 그래프 중심의 접근법을 제시하였다. 그는 논리그래프를 논리그래프 상에 정의된 특정 그룹의 오퍼레이션을 지닌 수리적 모델로 표현함으로써 어떻게 MMS에 포함될 수 있는가를 보여주고 있다. 논리모델에 그래프적인 구조를 제공함으로써 Kimbrough는 이들 모델들이 컴퓨터를 사용하여 조작될 수 있음을 또한 보여주고 있다.

여러가지 상이한 모델들을 표현하기 위한 프레임워크로 최근 각광을 받고 있는 연구로는 유속성그래프(attributed graphs)를 이용한 모델표현법들을 들 수 있다. 유속성그래프는 마디(nodes)와 가지(edges)를 여러 타입(type)으로 나눈 그래프이며, 각 타입은 그 마디나 가지와 관련하여 속성(attribute)을 갖고 있다. 유속성그래프의 이론적 기반은 그래프문법(graph grammars)에 있다(Jones 1992, p. 281).

그래프문법(graph grammars)에 근거한 Jones(1990,1991)의 GBMS(Graph Based Modeling System)는 유속성그래프 프레임워크의 대표적인 연구로 들 수 있다. GBMS는 컴퓨터 상에서 직접 그래프구조를 만들고 상호대화식으로 구조를 분석할 수 있는 능력을 지니고 있어 사용자 편리성이 높은 도구이다. 이 시스템은 또한 그래프 상에서 DB와 같은 질의를 지원한다. 따라서 이 시스템의 장점은 문제표현에 매우 유용하며, 그래프구조생성의 용이성은 분석자와 사용자 간의 의사소통을 증진시킬 수 있다.

또한 Jones(1992)는 Geoffrion(1987)이 제시한 구조적 모델링개념이 모델을 표현하는 데 있어 유속성그래프의 한 종류로 나타낼 수 있다는 사실에 착안하여 그래프 문법이 어떻게 구조적 모델링을 지원할 수 있는가를 보이고 있다. 그는 그래프 문법은 구조적 모델링을 위한 그래프적인 문법지향 편집환경을 제공할 수 있음을 제시하였다. 그리고 GBMS에 기반을 둔 Networks/SM 프로토타입 시스템을 개발하였다. 이러한 그래프 문법이 모델관리에 적용될 경우 다음과 같은 장점을 지닌다(Jones 1992, p. 287).

- ① 모델표현을 위해 널리 사용되는 포맷을 지니며, 그래프 조작에 대한 특정한 접근방법이 존재한다.
- ② 공식적 언어(formal languages) 이론에 근거한 확고한 이론적 기반

을 지니고 있어 그래프로 표현된 모델에 대한 명확한 명세를 제공해 줄 수 있다. ③ 그래프 상에서 수행될 수 있는 조작(그래프 문법 용어로는 productions이라 부름)을 표현하기 위한 간편한 시각적 언어(visual language)를 지니고 있다. 특히 프로덕션은 그래프로 표현되어 질 수 있다. ④수리계획이나 컴퓨터 시뮬레이션과 같은 특정 해법과는 독립적이다.

유속성그래프를 이용한 모델표현의 또 다른 연구로는 Glover(1990)의 NETFORM을 들 수 있다. NETFORM은 일반화네트워크(generalized network)문제를 방향그래프(directed graph)로 표현하는 모델링 툴(tool)이다. 또한 Collaud & Boltuck(1994)은 선형계획문제의 정의와 조작을 위한 그래픽 툴인 gLPS(graphical Linear Programming System)의 개발하였으며 모델표현을 위해 유속성그래프를 이용하였다. gLPS는 전통적인 선형계획문제를 그래픽 형태의 객체(graphical object)를 이용하여 특정 모델수립규칙에 따라 네트워크로 표현한다.

이들 그래프 중심의 주된 장점은 수리적 심볼이나 공식을 사용하지 않고 문제를 단 순명료하게 표현할 수 있는 데 있다. 그러나 수백개의 마디와 화살표를 가진 대규모의 문제를 그래프로 표현한다는 것은 실용적이지 못할 것이다.

5. 구조적 모델링 접근법

구조적 모델링(Structured Modeling)(Geoffrion 1987)은 다양한 형태의 모델을 표현하기 위한 강력한 이론적 프레임워크를 제공해 준다. 구조적 모델링의 공식적 프레임워크는 이산수학에 근거를 두고 있으며, 모델이나 모델클래스를 표현하기 위해서 계층적으로 조직화되고 분할된 유속성 비순환 그래프(attributed acyclic graph)를 사용한다.

구조적 모델링은 세가지 수준의 구조, 즉 요소적 구조(elemental structure), 일반적 구조(generic structure), 모듈러 구조(modular structure)를 갖는다. 요소적 구조는 특정 모델인스턴스의 정의적 세부사항을 표현하는 것으로 다섯 종류의 요소(primitive entity, compound entity, attribute, function, test element)로 구성된다. 요소적 구조는 이들 다섯가지 요소(nodes)와 이들 간의 정의적 참조를 나타내는 가지(arc)로 구성된 방향그래프(요소그래프)로 볼 수 있다. 일반적 구조는 유사한 요소들을 묶은 "Genus graph"로 표현한 것으로 문제 클래스를 나타낸다. 모듈러 구조는 일반적 구조내의 관련 구조체를 상위 객체(모듈)로 그룹화하여 일반적 구조를 계층적으로 조직화한 모듈러 트리(modular tree)로 나타낸다.

구조적 모델링의 이러한 그래프적인 구조는 사용자가 수리적 복잡성에서 벗어나 문제요소와 이들 요소들 간의 관계에 주목하도록 해준다. 그래프는 단지 효과적인 의사소통 목적이지만 곧바로 컴퓨터 실행가능한 표현법은 아니다. 실질적인 실행을 위해서 구조적 모델링은 텍스트 중심의 모델스키마를 제공해 준다. 스키마 표현을 위한 모델링 언어가 바로 SML(Geoffrion 1992)이다.

Geoffrion(1991)은 구조적 모델링의 장점을 다음과 같이 제시하고 있다. ①하나의 단일화된 엄격한 형식(formalism) 내에 다양한 형태의 모델과 모델링 패러다임을 다룰 수 있다. ②어떠한 모델에 대해서도 손쉽게 의사소통할 수 있는 다이어그램을 제공한다. ③모델부분들 간의 상호관계를 중요시 하고 있다. 이것은 대부분의 현행 모델링 형식들이 간과하고 있다. ④해산출기를 모델변경에서 독립시키고 있다. ⑤복잡성을 관리하기 위한 접근법으로 계층적 조직화를 이용한다. ⑥모델을 사용자가 직접

이해할 수 있으면서 곧바로 컴퓨터 상에 실행가능하도록 한다. ⑦데이터관리를 위해서 관계형 DB 툴을 쉽게 이용할 수 있도록 한다.

그러나 조그마한 문제라 할지라도 “genus graph”를 수작업으로 그리기가 쉽지 않다. 따라서 구조적 모델링을 지원하는 소프트웨어는 SML 스키마로 부터 그래프를 자동적으로 생성할 수 있어야 할 것이다. 표현의 구체성 측면에서 볼 때, 또한 구조적 모델링에서의 문제진술과 그래프는 상당히 추상적이어서 근본적인 문제구조를 인식하기가 다소 어렵다. 따라서 구조적 모델링은 모델명세를 위한 이상적인 외부표현을 제공하지는 못한다.

구조적 모델링 개념을 전적으로 지원할 수 있는 시스템으로는 Geoffrion(1991)이 개발한 FW/SM와 이를 발전시킨 Neustadter, et.al(1992)의 연구일 것이다. 이것은 통합 패키지인 Framework III로 구현한 프로토타입 시스템으로 20여 가지의 기능을 제공해 주고 있다. 이외에도 구조적 모델링의 특수한 일부분을 구현한 시스템들도 많이 있다. 이 중에서 특히 주목할 것은 GBMS를 결합한 Jones(1992)의 연구, 논리 시스템과의 결합을 시도한 Chari & krishnan(1990)의 연구, 그리고 관계형 DB와의 결합을 연구한 Lenard(1986)의 연구와 객체지향 데이터베이스관리시스템(ODBMS)와의 결합을 시도한 허순영(1994)의 연구를 들 수 있다. 또한 Dolk(1988b)는 IRDS (Information Resource Dictionary System)의 모델표현법으로 구조적 모델링을 채택하였다. 이외에도 많은 연구들이 구조적 모델링 개념을 채택하거나 언급하고 있다.

6. 객체지향 및 시스템 접근법

최근 들어 소프트웨어공학 분야에서 주된 관심을 끄는 객체지향개념(object-oriented concepts)을 모델관리에 도입하려는 일련의 연구들이 진행되고 있다(표 2-6). 여기서는 시스템 개념을 도입한 연구도 객체지향 MMS 연구의 한 부분으로 본다.

(표 2-6) 모델관리에 객체지향개념의 적용 연구

| 프레임웍 | MMS 또는 언어 | 연구자 |
|---------|-----------------------------|-----------------------------------------------------------------------------------------|
| 모델추상화 | GXMP GXMP | Dolk & Konsynski(1984) Dolk(1986a) |
| 객체지향KB | nn AIDSS PDM MIDAS | Mannino et.al.(1990) Lazimy(1988,1991) Krishnan(1991) Dempster & Ireland(1991) |
| 객체지향DB | nn | Huh(1993), 허순영(1994) |
| 모델가시화 | nn | Pracht(1990) |
| DB/KD통합 | (KDM/KDL) | Potter, et.al.(1992) |
| 시스템 | SYMMS(MDL) | Muhanna(1992,1993,1994) |

*nn: no named

그런데 이들 대부분의 연구들이 객체지향개념의 일부분만을 수용하고 있으며 의미데

이타모델의 확장이나 객체지향 데이터베이스 또는 지식베이스 개념을 일부 도입하고 있다. 따라서 진정한 객체지향 프레임워크라고는 볼 수 없으나 최소한 객체지향 MMS를 개발하는 데 있어 선구적 역할을 한 연구로 평가할 수 있다. 이 중 대표적인 연구를 소개하면 다음과 같다.

Dolk & Konsynski(1984)는 인공지능분야에서의 지식표현 문제와 DB 관리분야에서의 데이터 추상화 개념을 혼합한 모델추상화(model abstraction) 개념을 제시하였다. 모델추상화의 외부구조(개념적 구조)는 데이터추상화의 외부구조에 기초를 두고 있다. 즉, 모델추상화의 구조는 ①데이터객체(data object) ②절차(procedures) ③선언(assertions)으로 구성되며, 모두 일차술어연산으로 표현된다.

이러한 모델추상화 개념을 MMS 설계에 적용했을 때 다음과 같은 융통성 있고 강력한 장점을 지닌다(Dolk 1986a, p.102). ① 독립성 보장, ② 추상화 공간(Abstraction space)의 확립, ③데이터로서의 모델, ④ 추상화는 지식중심의 응용에 적용될 수 있다. 이외에도 모델추상화는 하나의 추상화가 또 다른 추상화의 데이터 객체나 절차가 될 수 있으므로 현재 있는 모델로부터 새로운 모델을 합성하거나 조합하기가 쉬우며, 모델관리에서 가장 중요한 모델들 간의 유사점, 차이점, 상호작용들 고려할 수 있는 메타추상화(meta-abstraction) 개념의 도입이 가능하다. 이러한 메타추상화 개념은 현재있는 모델로서 본다면 개념적 스키마로서의 메타추상화 개념은 프레임 개념과 일치한다.

Potter, et.al.(1992)는 데이터, 지식, 모델을 하나의 통일된 형태로 다룰 수 있는 KDM(Knowledge/Data Model)이라는 통합된 접근법을 위한 프레임워크를 제시했다. KDM은 의미데이터모델 접근법의 발전적 형태로 초의미모델(hyper-semantic model)이라 불리는 모델클래스에 속한다. KDM의 모델링 구성자로는 일반화, 분류화, 집합, 멤버십(membership), 제약식, 휴리스틱(heuristic), 템포럴(temporal) 등을 제시하고 있다. KDM은 KDL(Knowledge/Data Language)이라 불리는 스키마 명세 및 질의언어에 의해 실행되어진다.

KDM/KDL 프레임워크의 장점은 ① 계승개념에 의하여 기존 존재하는 모델의 재사용을 통하여 새로운 모델을 쉽게 만들 수 있다. ② 시스템 내의 각 객체타입들이 잘 정의된 지식중심의 구조로 구성되므로 사용자나 시스템에 의해서 보다 더 쉽게 관리되고 조작될 수 있다. ③ KDM/KDL 내에 있는 객체들을 보다 더 큰 시스템으로의 통합이 용이하다. ④ 메타수준의 활동이 시스템 내의 다른 객체로서 쉽게 구현될 수 있다.

시스템 개념은 현상 또는 프로세스를 이해하고 기술하는 데 있어 유용한 프레임워크이다. 시스템, 하위시스템, 인터페이스, 모듈성, 계층적 설계, 그리고 단계적 상세화 등은 시스템 분석 및 설계(소프트웨어 공학) 영역에서 일반적으로 사용되는 개념이다. Muhanna(1992, 1993, 1994)는 모델관리에 시스템 프레임워크를 도입하였다. 모델관리를 위한 시스템 프레임워크는 모델링환경에서 의미론(semantics)과 구조적 관계를 포착하는 데 기본이 되는 구조화 원리들과 풍부하고 직관적인 모델관점을 제공해준다. 또한 시스템 프레임워크는 모델합성과 해산출기 통합을 위한 그래프지향적, 비절차적, 계층적 접근법을 제공해준다.

Muhanna(1994a, pp.216-217)는 시스템 프레임워크를 위한 방법론의 기본이 되는 다섯 가지 원리를 다음과 같이 제시하고 있다. ①모델지식베이스, ②모델-데이터 독립, ③ 모델-해산출기 독립, ④모델스키마 명세는 외부(인터페이스)명세와 내부(구조적, 행위적)명세로 나누며, ⑤모델링을 소범위 모델링(Modeling-in-the-small)과 대범위 모델링(modeling-in-the-large)로 분리한다. 시스템 접근법의 기본 개념은 메타모델링

(meta-modeling)이며, 이 개념은 일반모델타입, 타입구체화, 원소 및 복합모델버전, 모델인스턴스, 그리고 파라메트화된 버전 등의 개념을 포함한다. 모델링환경에서 보다 나은 의미론을 포착하기 위해서 여러 가지의 개념과 구조화원리가 도입되었으며 이러한 메타모델링 개념들을 그가 제시한 MDL(Model Description Language)과 MMS인 SYMMS의 근간이 된다. MDL은 모델링언어의 차원을 넘어 모델통합언어의 역할을 한다.

이러한 시스템 프레임워크의 가장 큰 장점은 메타모델링 개념의 도입에 의해서 대규모의 모델베이스를 효과적으로 관리할 수 있는 매카니즘의 제공에 있을 것이다. 즉 ①모델타입과 클래스개념은 모델특성의 계승을 통하여 모델 재사용 매카니즘을 보여 주며, ②모델버전개념은 반복적, 잠정적 모델링환경에서 효과적으로 모델버전을 관리할 수 있게하며, ③모듈러적이고 계층적 관점은 모델의 재사용과 통합을 용이하게 하며, ④파라메트화된 버전은 모델클래스의 효과적인 저장과 공유를 가능하게 한다.

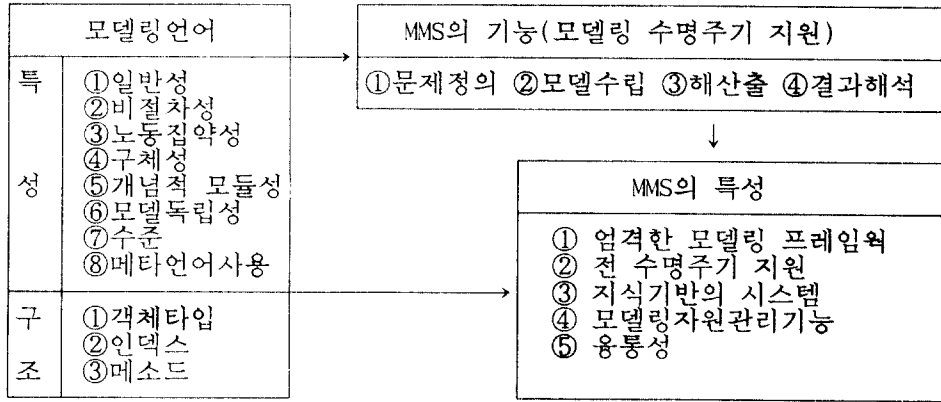
Huh(1993), 허순영(1994)는 다양한 모델링 패러다임의 모델을 표현하고 관리하기 위해서 객체지향 데이터베이스관리시스템(ODBMS) 개념에 기반을 둔 모델베이스 수립을 위한 프레임워크를 제시하였다. 모델베이스를 구축하는 데 있어 DSS의 구성요소인 모델베이스, DB, 해산출기 간의 논리적 독립성이 추구되어 왔으나 이들 요소들 간의 결합이라는 해결과제가 제기되고 있다. 그는 모델베이스 구축을 위한 개념적 기초로서 Muhanna의 시스템 프레임워크에서 제시한 일반모델개념을 채택하고 있다. 이 일반모델 개념은 단일 모델에 대한 다중 추상 관점을 제공해줄 뿐만 아니라 DSS 구성요소들 간의 통합을 지원할 수 있다. 또한 그는 모델표현을 위한 서버언어로서 모델링 패러다임-중립 언어인 SML(Geoffrion 1992)을 사용하였다. 모델베이스 구축에 있어 ODBMS는 융통성있는 DB 기술을 제공할 뿐만 아니라 강력한 프로그래밍 패러다임(객체지향 프로그래밍)을 수용할 수 있어 MMS의 구조적 운영적 요구를 지원할 수 있는 단일의 공식기구로 채택될 수 있다.

제 3 장 諸 接近法의 비교

좋은 모델링 시스템이 되기 위해서는 문제 모델링을 위한 엄격한 개념적 프레임워크를 제공해야 하며, 여러 가지의 모델 타입 클래스(class of model type)를 표현할 수 있어야 한다. 이러한 문제들은 모델을 어떻게 표현하느냐가 그 기본이 된다. 따라서 효과적인 모델표현방법의 개발이 시스템 성패를 좌우한다고 볼 수 있으므로 모델링언어의 표현방법의 우수성을 측정할 수 있는 기준의 도입과 기존의 MMS 프레임워크들이 이러한 기준에 어느 정도 부합하느냐를 분석하는 것이 필요하다. 또한 모델링언어가 컴퓨터 상에서 곧바로 실행되게 하기 위해서는 그 모델링언어 고유의 문법을 지니고 있어야 한다. 그리고 향후 개발되는 모델링언어의 문법은 기존에 개발된 모델링언어의 문법을 수용할 수 있는 매카니즘을 지녀야 만 이미 다른 문법으로 작성된 모델을 그 시스템내에서 서로 호환성있게 다룰 수 있다. 이를 위해서는 무엇보다도 기존의 모델링언어의 문법체계를 분석할 필요가 있다.

한편 MMS는 그 기능면에서 모델의 개발과 관리를 위한 매카니즘이 있어야 하며, 이를 평가할 수 있는 기준이 필요하다. 본 연구에서는 이를 위한 기준으로 모델링 수명주기의 지원형태와 정도를 사용하고 있다. 이러한 기능평가에 대한 기준은 개발된 시스템의 기능충족성을 판정하는데 유용하다. 또한 MMS의 기능명세와 설계를 위한 지침이 되기도 한다.

본 연구에서는 모델링언어 모델표현특성 및 언어구조와 MMS의 기능비교를 위해 (그림 3-1) 같은 프레임워크를 제시한다.



<그림 3-1> 기존 MMS연구 비교분석을 위한 프레임워크

1. 모델링언어의 비교

1) 모델링언어의 비교

모델링언어는 사용자가 표현한 모델을 컴퓨터 상에서 실행가능하도록 하는 공식적인 표기법이다. 즉 모델관리의 특정 프레임워크에서 제시한 추상적인 개념을 직접 또는 간접적으로 컴퓨터가 실행가능하도록 하는 언어이다. 이러한 모델링언어를 분류하기 위한 기준들이 몇 가지 제시되고 있다.

Fourer(1983)는 두 종류의 모델링언어 표현법(modeler's form / algorithm's form)을 제시하였다. 모델러 형태는 기호적이면서 일반적이고 간결하여 사용자가 쉽게 이해할 수 있는 언어인 반면 앨고리즘 형태는 해산출기(solver)가 이해할 수 있는 형태로 문제를 기술한다. 오늘날 대부분의 모델링언어는 모델러 형태에 속한다.

Maturana(1994)는 모델링언어의 분류기준으로 ①문자형과 그래픽적인 형태 ②절차형과 선언형 ③일반형과 특수형으로 구분하였다. 종래에는 컴퓨터상에 그래픽적인 정보를 표현하는 것이 어려웠기 때문에 대부분의 모델링언어는 문자형이다. 정보처리기술의 발달은 그래픽적인 모델링언어(예, GBMS)의 발전을 가져왔다. 그러나 컴퓨터 상에서의 문자정보처리는 그래픽적인 정보처리보다 훨씬 쉽고 효율적이다. 대부분의 그래픽적인 모델링언어는 문자표현에 대한 전방처리기(front-end) 역할을 한다. 즉, 그래픽적인 표현은 문자적 표현으로 전환된 후 모델링 시스템에 의해 처리된다. 문자형태가 아닌 또 다른 모델링언어로 일련의 메뉴를 이용하여 상호대화식으로 이루어지는 것도 있다(예 CAMPS[Locus & Mitra 1988]).

모델링언어는 절차적(procedural)인 것보다는 선언적(declarative)인 것이 많은 경향을 띤다. 하지만 이들 대부분이 완전히 선언적이거나 절차적인 것은 없고 두 가지를 혼합하거나 병용한다. 어떤 모델링언어들은 두 종류의 하위언어(sublanguage), 즉 모델정의언어와 모델조작언어를 지닌 것도 있다(대부분의 데이터베이스지향 프레임워크들이 여기에 해당한다). 특수형 언어는 특수한 구조를 지닌 문제를 지원하기 위해서

만들어진 언어이다(예, 네트워크모델을 위한 GNGEN[Forster 1988]). 이들 언어들은 보통 익히기 쉬우며 간결하지만 특수한 구조를 지닌 문제를 지원한다는 측면에서 일반성이 결여되어 있다.

Murphy, Stohr, and Asthana(1992)는 모델표현법의 비교기준으로 ①일반성(generality) ②비절차성(nonprocedurality) ③노동집약성(labor-intensiveness) ④구체성(concreteness) ⑤개념적 모듈성(conceptual modularity) ⑥모델독립성(model independence)을 제시하고 있다.

일반성은 표현법이 LP모델에 국한되지 않고 여러 가지의 경영과학 모델에 대한 적용가능성을 의미한다. 비절차성은 사용자가 '어떻게(how)' 명령문을 구성하는가는 알 필요없이 단순히 문제가 '무엇(what)'인가만을 정의하면 되도록 하는 것이다. 사용자가 모델의 문서화와 기록을 위해 요구되는 노동집약성은 표현의 복잡성 정도에 달려 있다. 구체성은 실세계의 객체와 이들간의 관계를 얼마나 유사하게 나타내는가에 대한 척도이다. 따라서 표현의 구체성이 약하다는 것은 추상적인 수학적 표현이 강함을 의미한다.

개념적 모듈성은 표현양식이 인지적 복잡성을 감소시키는 문제해결전략을 얼마나 잘 지원해줄 수 있는지에 대한 척도이다. 이를 위해서는 먼저 표현법은 계층적 추상화(Smith & Smith 1977)를 지원해야 한다. 다음으로 표현법은 하향식 분해를 지원해야 한다. 셋째로 표현법은 상향식 문제해결전략을 촉진시켜 이미 개발된 모델과 모델부분들을 재사용 할 수 있게 해야 한다. 구조적 모델링 표현은 이러한 형태의 모듈성을 제공해준다. 끝으로 사용자는 모델의 부분을 순서에 상관없이 정의할 수 있어야 하며, 문제의 요소중 결여된 부분을 추론하여 적절한 순서에 맞추어 일관성있게 내부적 표현으로 구성하는 데 필요한 작업을 컴퓨터가 할 수 있어야 한다.

모델독립성이란 모델을 하나의 논리적 실체로 보아 그 자체적으로 분리할 수 있음을 의미한다. 모델독립성에 관련된 개념으로는 모델-데이터 독립성과 모델-해산출기 독립성을 들 수 있다(Geoffrion 1987). 모델-데이터 독립성은 모델구조문과 모델에서 사용되는 데이터의 분리를 의미한다. 모델-해산출기 독립성은 모델문장이 특정 해산출기의 요구에 의존하지 않는 포맷으로 됨을 의미한다.

Bhargava & Kimbrough(1993)는 수리적 모델표현을 위해서 세가지 수준의 모델링언어 접근법을 제시하였다. 첫번째 수준의 언어는 자연언어와 수리적 표현이 혼합된 컴퓨터 상에서 곧바로 실행불가능한 준자연언어문장(quasi natural language statement)이다 (이것을 $L^*\downarrow$ 로 표시함). 두번째 수준의 언어는 GAMS, AMPL, 또는 SML과 같이 완전히 공식화되어 있으며 실행가능한 모델링언어(EML: Executable model languages)이다 (이것은 $L\downarrow$ 로 표시함). 그런데 이들 언어들은 첫번째 수준의 언어와 같이 자연언어 문장으로 표현되는 모델의 부가적인 수리적 특성과 정보를 표현하지 못한다. 세번째 수준의 언어($L\uparrow$)는 모델에 관한 이러한 종류의 정보를 표현할 수 있을 뿐만 아니라 $L\downarrow$ 로의 표현규칙과 $L\downarrow$ 에서 $L\uparrow$ 로 매핑(mapping)하는 규칙에 대한 정보를 갖고 있다.

이상에서 제시한 여러 기준들은 Murphy, Stohr, and Asthana(1992)의 경우를 제외하고는 약간 단편적이다. 따라서 본 연구에서는 모델링언어의 모델표현특성을 비교하기 위해서 Murphy, Stohr, and Asthana(1992)의 기준에 Bhargava & Kimbrough(1993)의 모델링언어의 수준과 메타언어의 사용여부를 추가한다. 왜냐하면 Murphy, Stohr, and Asthana(1992)의 기준은 Fourer(1983)와 Maturana(1994)의 기준을 수용하고 있으며, 메타모델링언어의 사용여부가 시스템의 모델링 효율성을 좌우하는 중요한 요소이기

때문이다. 즉, 메타언어는 매크로 수준(macro level)에서는 모델링을 가능하게 하며 모델의 재사용을 촉진시키며 모델링 노력을 현격히 절감시킬 수 있다.

모델링언어의 표현특성비교를 위해 앞에서 제시한 각 접근법별로 대표적인 2~3가지의 모델링언어를 선정하였으며 대부분의 프레임워크가 포함되도록 하였다. 각 접근법별로 선정된 모델링언어와 참고문헌을 제시하면 (표 3-1)과 같으며 이들 언어를 여덟가지의 표현특성에 따라 서로 종합적으로 비교하면 (표 3-2)와 같다.

여기서 주의할 것은 동일한 접근이라도 그 프레임워크에 따라 약간의 차이는 있을 수 있다. 그러나 전반적인 관점에서 다음과 같은 시사점을 찾을 수 있다. 구조적 접근법이나 객체지향 접근법이 패러다임 중립의 모델링방법을 택하므로 일반성이 높으며, 수리적 모델링언어 지향의 시스템들은 대부분 LP문제를 다루는 것이 많아 일반성이 낮다. 비절차성 면에서는 지식중심, 그래프중심, 구조적 접근법들이 높다고 볼 수 있으며, 노동집약성과 구체성은 수리적 모델링언어지향적 접근법이 낮아 모델의 추상성이 높다고 볼 수 있다. 개념적 모듈성과 모델독립성은 대수적 모델링언어를 제외한 모든 방법들이 높았다.

여기서 또 한가지 생각해 볼 수 있는 것은 모델링언어를 모델정의중심언어와 모델 조작중심언어, 그리고 이들이 혼합된 언어로 구분할 수 있다. 이것은 다음에 설명할 상위수준(high-level) 문법분석을 통해서 더욱 더 명백히 드러난다. 모델정의 중심의 언어로는 AMPL, PCL, SML 등을 들 수 있으며, 조작중심의 언어로는 MQL, ETQL, SQLMP 등이 있다. 그리고 나머지 대부분은 모델정의와 조작을 동시에 나타내는 혼합형이다.

<표 3-1> 비교를 위해 선정된 모델링언어

| 접근법 | 모델링언어 및 참고문헌 |
|--------------|----------------------------------------------------------------------------------------------------------------------|
| 수리적 모델링 언어지향 | OMNI(Harverly Systems 1976), GAMS(Bisschop & Meeraus 1982) AMPL(Fourer, et.al. 1990), MathPro(MathPro, Inc. 1990) |
| 데이터베이스 지향 | E-RM(Elam 1980), ERLMP(Choobineth 1991b), MQL, ETQL(Blanning1982, 1985, 1987), SQLMP(Choobineth 1991a) |
| 지식 중심 | PCL((predicate caculus language)(Dutta & Basu 1984) PM* (Krishnan 1989, 1990, 1991) |
| 그래프 중심 | Model-graph(Liang 1986, 1988), GBMS(Jones 1990, 1991) NETFORM(Glover 1990), gLPS(Collaud & Boltuck 1994) |
| 구조적 | SML(Geoffrion 1992b) |
| 객체지향/시스템 | Model Abstraction)(Dolk & Konsynski 1984), KDM/KDL(Potter, et.al. 1992)을 |

<표 3-2> 모델링언어의 표현특성 비교

| 언어 | 기준 | 일반성 | 비절차성 | 노동집약성 | 구체성 | 개념적 모듈성 | 모델 독립성 | 수준 | 메타언어 |
|-------------------|----|-----|------|-------|-----|---------|--------|-----|-------------------|
| OMNI | | ○ | ● | ○ | ○ | X | ○ | EML | no |
| GAMS | | ● | ● | ○ | ○ | ○ | ○ | " | no |
| AMPL | | ● | ● | ○ | ○ | ○ | ● | " | no |
| MathPro | | ○ | ● | ○ | ● | ● | ● | L3 | no |
| E-RM | | ● | ● | ● | ● | ● | nd | QNL | no |
| ERLMP | | ★ | ● | ● | ★ | ● | nd | " | no |
| MQL, ETQL | | ● | ● | ● | ● | ● | ● | EML | Virtual-relation |
| SQLMP | | ● | ● | ● | ● | ● | ● | " | " |
| PCL | | ● | ● | ● | ● | ● | ● | " | Many-sorted-logic |
| PM* | | ○ | ● | ● | ● | ● | ● | " | Meta-frame |
| Model-graph | | ● | ● | ● | ● | ● | nd | QNL | Model-graph |
| GBMS | | ● | ● | ★ | ★ | ● | nd | EML | Meta-graph |
| NETFORM | | ○ | ● | ★ | ★ | ● | nd | QNL | no |
| gLPS | | ○ | ● | ★ | ★ | ● | nd | EML | no |
| SML | | ★ | ● | ● | ● | ★ | ★ | " | Genus-graph |
| | | | | | | | | | Generic-rule |
| Model-Abstraction | | ★ | ● | ● | ● | ★ | ★ | " | Meta-abstraction |
| KDL | | ★ | ● | ● | ● | ★ | ★ | " | Hyper- |
| MDL | | ★ | ● | ● | ● | ★ | ★ | L3 | Metamodeling |

*범례: X 없음 ○ 낮음 ● 중간 ● 높음 ★ 매우높음 nd: no discussed

** Quasi Natural Language =QNL / Executable Model Languages = EML / Level 3 = L3

2) 모델링언어의 문법비교

모델링언어는 선언적 표기법, 대수적/절차적 표기법, 그리고 인덱스구조 등에 의해 모델표현의 문법적, 의미적 공통성을 찾을 수 있다. 따라서 기존의 여러가지 모델링언어의 문법에 내재된 공통된 특성을 서로 비교분석함으로써 모델링언어의 일반적인 구조를 도출할 수 있다. 일반적 구조는 개별 언어 특유의 문법과는 독립적으로 모델구성요소들을 단일적으로 포착할 수 있으며, 異種의 모델링언어를 하나의 모델베이스에 저장할 수 있게한다(허순영 1994).

모델링언어의 문법을 비교하기 위해서 본 연구에서는 (표 3-1)의 모델링언어 중에서 각 접근법에서 널리 알려진 여덟가지의 모델링언어를 선택하여 먼저 각 모델링언어의 상위수준 문법(high-level syntax)을 도출한다. 예를 들어 GAMS의 상위수준 문법은 (그림 3-2)과 같다. 여기서, 대괄호: []는 선택사항, 수직선: | 은 대체성, 그리고 괄호: < >는 반복을 표시한다. 명령문(statement)은 표현식(expression)과 구분된다. 명령문은 문법적으로 완전한 문장인 반면 표현식은 명령문의 일부분으로 불완전한 부분적 선언이다.

그리고 이들 언어들이 모델정의를 위해 사용한 객체타입과 인덱스구조, 그리고 모델조작과 오퍼레이션을 위해서 사용한 모델메소드(model method)를 찾아낸다(표 3-3).

이러한 모델링언어의 문법분석은 앞에서와 같이 모델링언어 분류를 위해 유용하게 사용될 수 있을 뿐만 아니라, 향후 모델링언어의 개발의 기초가 된다. 또한 개발된 모델링언어와 기존의 언어와의 비교기준이 된다. 좋은 모델링 언어가 되기 위해서는 (표 3-2)에서 제시되는 여러가지의 객체타입과 메소드를 수용할 수 있는 언어이어야 한다. 이러한 기준에서 볼 때, KDM/KDL은 아주 우수한 모델링언어로 평가할 수 있다.

<표 3-3> 각 모델링언어의 상위수준 문법의 분석표와 특징

| 언어 | 객체타입 | 인덱스(Index) | 메소드(method) | 특징 |
|-------------------|------------------------------------------------------------------------------------------------|-----------------|-----------------------------------------------------------------------------------------|-----------------------------------------------------------------------------|
| OMNI | CLASS, TABLES | class entities | FORM TABLE, VALUE, ROW, VECTOR, BOUNDS, VECTOR RHS, EXECUTE, etc. | FORM SECTION에 의해 명시적으로 MPS 파일을 만들. |
| GAMS | SET, TABLE, PARAMETER, SCALA, VARIABLE EQUATION | set entities | MODEL, SOLVE, DISPLAY, ABORT, OPTION, LOOP, etc. | *최초의 비절차적 선언문 형태 채택 *Domain cecking 기능 *모델-데이터 종속성. |
| AMPL | SET, PARAM, VAR, SUBJECT TO, MAX / MIN, DATA | set entities | | *인덱스집합정의를 위한 다양한 타입제공. *모델-데이터 독립성유지. *Dummy indices 사용가능. |
| SQLMP | CREATE TABLE, CONSTRAINT | | COMPUTE | *인덱스 사용 없음. *모델정의와 조작용을 위해 SQL형태의 문법제공. |
| PCL | Model_predicate, Domain_predicate, WFF, Input_list, Output_list, Argument_list, Term, Variable | | | * 모델술어와 영역술어 * 다중분류논리 사용 * Solver로 추론엔진 사용 * 추론효율을 높이기 위해 평가함수 사용. |
| SML | pe, ce, a, va, f, t | indexed genuses | | *데이터는 Elemental detail Table에 저장 --> 모델-데이터독립. *패러다임중립의 EML임. |
| Model Abstraction | Abstraction, Data object (OF, EXPR, IDXSET, VECTOR, PAR, DV, etc.) | | Procedures (ADDOF, ADDCON, DELOF, DELDV, Tn, etc.) ASSERTION (LINEAR, Tn, WFF, etc.) | * 모델-Solver 독립 * 추상화공간 확립 * 메타추상화개념 도입 |
| KDM/KDL | SUPERTYPE, SUBTYPE, ATTRIBUTES, MEMBERS | | CONSTRAINTS, HEURISTICS, METHOD | * 계승개념에 의한 모델 재사용 강화. * 절차적 지식과 비절차적 지식의 혼합. |

Entity declaration and assignment statement

```
SET entity_name [comment] [/assignment statement/]
PARAMETER entity_name[(set_entity_name <,set_entity_name>)] [comment]
[/assignment statement/]
TABLE entity_name(set_entity_name <,set_entity_name>) [comment]
[table statement]
SCALA entity_name [comment] [/assignment statement/]
[range] VARIABLE entity_name [comment] [/assignment statement/]
EQUATION entity_name[(set_entity_name <,set_entity_name>)] [comment]
[definition statement]
```

Model manipulation statements

```
MODEL model_name detail_statement
SOLVE detail_statement
DISPLAY detail_statement
ABORT detail_statement
OPTION detail_statement
LOOP detail_statement
```

〈그림 3-2〉 GAMS의 상위수준 문법

분석결과 수리적 모델링언어와 SML은 각자 독특한 인덱스 구조를 명시적으로 표현하고 있으며 다른 언어들은 인덱스구조가 없거나 묵시적 형태로 사용하고 있었다. 그리고 수리적 모델링언어들은 객체타입과 메소드를 엄격히 몇 가지로 한정함으로써 표현의 일반성을 희생하는 대신 컴퓨터 상의 실행가능성을 높이고 있다. 반면 데이터베이스나 지식중심의 모델링언어들은 객체의 선언에 치중하며, 한정된 메소드(모델/데이터 질의중심)를 지닌다. 객체지향언어들은 대한 다양한 타입과 메소드를 정의할 수 있게 하므로써 모델링의 융통성을 넓힐 수 있다.

2. 모델관리시스템의 기능비교

일반적으로 모델링활동은 적절한 수리적 모델을 수립하여 그 일련의 분석을 행하는 수명주기로 볼 수 있으며 크게 네 단계 ① 문제정의 ② 구조화 및 모델수립 ③ 해산출 ④ 민감도분석 및 결과해석 및 실행으로 나눌 수 있다. 모델링 수명주기 지원이란 이와 같은 일련이 작업을 수행하는 데 있어 MMS가 제공할 수 있는 기능과 업무(task)이다. MMS의 목적은 바로 모델링 수명주기의 전 단계를 지원하는 데 있다. 기존의 각 접근법별 대표적인 MMS의 모델링 수명주기 지원형태를 요약하면 (표 3-4)과 같다.

1) 문제정의

이 단계는 주로 문제해결자의 문제영역에 대한 지식에 의존한다. 따라서 MMS는 문제상황과 내용들에 대한 정보를 추출하고 수송문제, 생산믹스문제 등과 문제타입을 식별한다. MMS에서 문제에 대한 정보를 얻는 방법으로는 명령(commands), 메뉴(menus), 다이얼로그(dialog), 아이콘(icon) 등의 방법으로 이루어진다. 이것은 MMS의 사용자 인터페이스 설계에 달려있다.

<표 3-4> MMS의 모델링 수명주기 지원형태

| MMS | 문 제 인 식 | | 모델수립 | 해 산 출 | | 민감도분석 및 결과해석 |
|--------|----------------|-------------|----------------|-----------------|---------------------|--------------------|
| | 문제추출 | 모델타입 인 식 | | 데이터접근 및 모델전환 | Solver interface | |
| OMNI | 기호적언어 | 사용자 | 사용자 명세 | 사용자 프롬프트 | yes | yes |
| GAMS | 기호적언어 | 사용자 | 사용자 명세 | 사용자 프롬프트 | yes | yes |
| GMMS | na | 사용자 | 모델선택 | 사용자 명령 | yes | nd |
| GXMP | 메뉴선택 | 사용자 | 모델선택 | 사용자 명령 | yes | yes |
| ACS | na | 시스템 | 모델조작 | 사용자명령 | no | no |
| LPFORM | 그래픽 | 시스템 | 일차원리, 템플릿선택 | 사용자명령 | no | no |
| TIMMS | 다이얼로그 | 시스템 | 모델조작 | 자동 | yes | no |
| PDM | 문자 | 시스템 | 일차원리, 모델링규칙 | nd | | |
| AIMM | 다이얼로그, 다중선택 | 시스템 | 모델조작 | 사용자 프롬프트 | yes | yes |
| GBMS | 풀다운메뉴 및 아이콘 | 사용자 | 사용자 명세 | 자동 | yes | no |
| FW/SM | 풀다운메뉴 | 사용자 | 사용자 명세 | 자동 | yes | yes |
| SYMMS | 다이얼로그 | 시스템 | 모델조작 | 자동 | yes | no |

* na=not applicable, nd=no discussed.

2) 모델수립

이 단계는 문제정의 단계에서 정의된 모델을 수리적 모델로 구조화하는 단계이다. 즉 적절한 모델링 패러다임을 선정하고 공식적 모델명세를 하는 단계로서 많은 모델링 지식(영역지식, 문맥정보, 모델링 패러다임에 관한 지식 등)을 필요로 한다. 따라서 이 단계는 전 단계 중에서 가장 복잡하며 가장 많은 연구가능성을 지닌다. 이 분야에 대한 연구는 MMS의 성능을 향상시킬 뿐만 아니라 모델링과정 그 자체에 대한 이해를 촉진시켜 모델수립의 일반적인 이론을 개발하는 데도 중요한 역할을 할 것이다. 특히, 자동화된 모델링을 지원을 위해서는 지식중심의 시스템개발이 요구된다.

자동적 모델수립을 위한 기법으로는 ①일차원리(first principles) ②템플릿 채우기(template filling) ③모델조작(model manipulation) ④유추에 의한 모델링 기법 등이 있다. 일차원리기법은 스크래치(scratch)로부터 모델을 구축하는 것을 말한다. 이 기법은 먼저 문제정의문을 이해하고, 그 문제에 부합되는 특정 모델타입을 식별하고, 그리고 필요한 파라메타로서 적절한 모델을 구성한다. 이러한 방법을 택하고 있는 시스템 또는 연구로는 Murphy & Stohr(1986), Binbasioglu & Jarke(1986), Skar, et.al.(1987), LPFORM(Ma, et.al, 1989), PDM(Krishnan 1991) 등을 들 수 있다.

템플릿 채우기 기법은 주어진 문제를 모델베이스 내의 한 모델템프릿과 대응시켜 모델을 수립하는 방법으로 매번 완전히 새로운 모델을 수립해야 하는 번거러움을 든다.

이 기법을 사용하는 시스템으로는 LPFORM과 Mannino, et.al. (1990) 등이 있다.

모델조작 기법은 특정 문제에 대한 모델을 수립하는 데 있어 그 모델을 구성하는

하위모델(sub-model)을 자동적, 동적으로 선택하여 적절히 나열(sequencing)하거나 실행한다. 이 기법을 사용하는 시스템으로는 ACS(Sivasankaran & Jarke 1985), TIMMS(Liang 1988), AIMM(Liu, et.al. 1990), SYMMS(Muhanna 1994)등을 들 수 있다. 모델수립을 위해 최근 주목을 받는 방법으로 Liang(1993)의 유추에 의한 모델링기법이다.

자동적 모델수립을 지원하지 못하는 시스템에서는 사용자가 문제의 타입을 식별하고 모델베이스에서 적절한 모델을 선택한다. 물론 타당한 모델이 모델베이스에서 이용가능해야 하며, SELECT <modeltype>과 같은 선택문장이 아이콘을 이용하여 적절한 모델을 선택할 수 있어야 한다. 데이터베이스나 그래프중심의 프레임워크로 구현된 시스템들은 대부분 이 방법을 사용한다.

3) 모델 해산출

모델 해산출(Model solution) 단계의 주요 과업으로는 모델전환(model translation), 데이터접근(data access), 해산출기와의 인터페이스 및 해도출 등이다. 모델전환은 특정 해산출기가 인식할 수 있는 형태로 모델을 전환하거나 다른 타입의 문제로 전환하는 작업이다. 이것의 대표적인 형태가 매트릭스 생성기이다. 매트릭스 생성기를 이용하는 방법은 많은 프로그래밍능력과 수리계획법에 대한 지식이 요구된다. 따라서 MMS를 사용하여 내부적으로 필요한 전환과 적절한 해산출기와의 인터페이스를 자동적으로 수행하도록 하는 시스템의 개발이 필요하다.

또한 특정 타입의 문제를 다른 타입으로 바꾸는(예를 들어 네트워크문제를 LP문제로 전환) 매카니즘을 MMS 내에 갖출 필요가 있다. 이것은 동일한 문제를 여러 가지 모델 형태로 볼 수 있게 하며 다양한 형태의 해산출기를 이용할 수 있게 한다. Dolk & Konsynski(1984)가 제시하는 추상화공간(abstraction space)은 이러한 문제를 다루는데 적합하다.

데이터접근은 시스템에 따라 다양하다. 자동적 데이터접근이 가능한 시스템은 사용자는 데이터베이스 조작과 해산출기와의 연결방법을 몰라도 된다. 수리적 모델링언어 접근법을 채택하는 대부분의 시스템은 대화식으로 데이터를 입력이나 모델내에 데이터 부분을 별도로 선언하거나 모델과 같이 기록한다. 반면 데이터베이스지향적인 MMS들은 대부분 데이터베이스에서 필요한 데이터를 불러오기 위해서 SQL과 같은 언어를 이용하여 명령을 내린다. 그래프나 구조적 접근법을 택하는 대부분의 시스템들은 자동적 데이터접근 기법을 택한다. 지식중심의 시스템은 데이터접근방법에 상당한 차이를 보이고 있다.

모델 해산출 작업은 모델과 데이터 파일을 적절한 해산출기와 연결하는 것이다. 대부분의 MMS들이 이 기능을 가지고 있다. 그러나 지금까지 개발된 대부분의 시스템들이 단지 하나의 모델타입(예 LP)과 하나의 해산출기(예 Simplex Algorithm) 간의 자동적 접근만을 지원한다. 따라서 적절한 해산출기의 선택이란 말이 무의미했다. 이러한 문제를 극복하기 위해서는 객체지향개념에서 사용되는 다형성(polymorphism) 개념의 도입으로 해결될 수 있다. 이것은 하나의 모델객체에 대해 여러 가지의 해산출기를 선택할 수 있는 매카니즘을 제공해준다.

4) 모델해석

이 단계는 모델가정의 검토, 민감도 분석, 필요시 모델의 수정 등의 활동이 포함된다. 많은 대수적 모델링언어들이 민감도 분석을 위한 매카니즘을 제공하고 있다. 민

감도 분석이 가능한 시스템으로는 OMNI, GAMS, AMPL, LINDO, GXMP, FW/SM, AIMM 등이 있다. 한편 ANALYZE(Greenberg 1983)는 민감도 분석 뿐만 아니라 모델문서화, 타당성 검토, 그리고 결과해석 등을 수행할 수 있다.

기존의 MMS들의 대부분이 가정검토와 모델수정을 지원하지 못하고 있다. 이러한 문제를 다루는 데 있어 주목할 만한 연구는 AMPL의 도메인 체킹(domain checking) 기능으로 테이블의 열과 행의 이름이 그 테이블에 선언된 집합(SETS)인가를 체크한다. 또한 Dolk & Konsynski(1984)의 모델추상화에 사용한 선언문(예, LINEAR)도 가정검토를 자동화하는 데 사용될 수 있다. 모델수정을 직접적으로 지원하는 시스템으로 Muhanna(1994)의 SYMMS를 들 수 있다. 그는 모델버전개념을 도입하여 임시적, 잠정적 모델을 효과적으로 관리할 수 있는 매카니즘을 제시하고 있다.

끝으로 지식중심의 접근법은 모델의 결과를 해석할 수 있는 전문가시스템의 개발을 가능하게 하여 MS/OR의 지식이 없는 사용자도 결과를 쉽게 해석할 수 있도록 하며, 해산출과정을 요약하여 보여주거나 민감도 분석을 그래픽한 형태로 제시할 수 있는 연구들도 필요하다. 또한 계량적 모델의 해를 질적인 문제와 결합할 수 있는 모델사후분석(post model analysis)도 MMS의 중요 연구분야이다.

제 4 장 결론 및 연구과제

모델관리를 위해서는 다양한 원천의 지식(MS/OR, Database System, AI & Expert System, Software Engineering 등)이 필요하며, 효과적인 모델관리시스템을 개발하기 위해서 여러 가지의 연구가 진행되어 왔다. 즉 ① OR문제를 푸는 팻키지의 개발, ② 모델링언어의 개발, ③전통적 데이터모델의 개념을 도입하여 상용화된 DBMS에서의 구현, ④인공지능 개념을 도입하여 전문가시스템 셸(shells)을 이용한 구현 등 여러 가지 분야에서의 연구 결과들을 모델관리에 도입하려는 일련의 움직임이 있어 왔다. 또한 최근의 주류는 역시 모델관리에 객체지향개념을 도입하려는 연구가 주류를 이루고 있다.

그런데 이러한 노력들과 이상에서 제시한 기준이 얼마 만큼 잘 작동할지는 본질적으로 표현법 뿐만 아니라 시스템의 구현방법에도 관계된다. 그러나 표현법의 선택이 시스템 성과에 지대한 영향을 미치기 때문에 표현의 세부적인 설계시 필요한 정보를 고려해야 한다. 그리고 특정 모델표현법의 선택은 이상에서 제시한 기준에 비추어 볼 때 각각 장·단점이 있으므로 다분히 모델링 환경과 사용자의 선호도에 따라 달라질 수 있다. 사용자는 경영자, MS 전문가, 엔지니어, 학생 등 다양할 수 있으며 이들은 또한 모델의 개발과 사용시 동일한 모델에 대해 여러 표현법을 사용하기를 원할 수도 있다.

그러나 이상적인 모델링 시스템이 되기 위해서는 아마도 그래픽(아이콘), 네트웍, 그리고 대수적 표현기법이 포함된 한 가지 이상의 외부적 모델표현을 수용할 수 있어야 할 것이다. 또한 MMS는 특정 모델링패러다임 만을 지원하기 보다는 일반성이 높은 시스템을 지향하며, 사용자가 자신이 무엇을 할 것인가 만을 알면 시스템을 사용하여 원하는 결과를 얻을 수 있는 시스템이 되도록 하며, 사용자가 복잡한 문제를 보다 더 쉽고 정확하게 모델링 할 수 있게 하고, 모델의 유지보수를 용이하게 하는 표현법의 개발과 이를 잘 반영할 수 있는 시스템개발에 초점을 두어야 할 것이다.

따라서 어느 방법이 가장 이상적이라고 지적할 수는 없으나, 지금까지의 연구결

과를 종합하면 다음과 같은 특성들을 갖춘 모델링환경 및 시스템이 이상적이라 할 수 있다.

① 엄격한 이론적 기반을 가진 모델링 프레임워크와 사용자가 이해하기 쉽고 자신의 관점을 정확히 반영할 수 있는 모델표현기법 및 툴(tools)의 개발이 우선되어야 한다. 물론 모델표현의 일반성과 비절차성이 높고, 개념적 모듈성과 모델독립성이 강력히 지원될 수 있어야 하며, 표현된 모델의 자동적 문서화를 가능하게 해야 한다.

② 컴퓨터의 의한 모델링 수명주기의 전 단계를 지원할 수 있어야 하며, 모델관리의 주요 자원(모델, 데이터, 해산출기(solvers), 모델관리지식)의 관리기능이 있어야 한다. 또한 그 모델링 환경에서 만들어진 모델의 지속적인 발전이 용이해야 한다. 즉 모델의 버전관리 기능이 있어야 한다.

③ 모델정의를 위한 단일의 패러다임 중립 언어(paradigm-neutral language)가 있어야 하며, 이것은 자연언어에 가까우면서(記述的 성격이 강함) 컴퓨터 상에서 곧바로 실행가능한 언어이어야 한다. 또한 기존의 모델링언어 문법으로 표현된 것들을 수용할 수 있도록 그 문법이 충분히 일반적이어야 한다.

④ 모델활동은 많은 모델링 지식과 도메인 지식을 필요로 하므로 지식기반의 시스템이 되어야 한다. 즉 문제인식과 구조화 및 모델수립, 모델의 결과해석 등에 많은 전문적인 지식이 필요하므로 이를 지원할 수 있는 전문가시스템의 개발이 요구된다.

⑤ MMS는 융통성이 있어야 한다. 즉 MMS는 상이한 모델을 수용하고 단일 모델의 다중관점을 지원할 수 있도록 충분히 일반적이어야 한다. 또한 MMS는 상이한 응용영역에 대한 모델의 사용을 지원해야 한다. 더우기 모델에서 요구되는 다양한 원천(사용자, 데이터베이스, 또는 다른 모델의 출력)의 데이터를 결합할 수 있어야 한다.

이상에서 살펴본 모델관리에 대한 기존의 연구에 대한 비교분석 및 표현 상의 특성, 그리고 바람직한 모델링환경과 MMS의 요건에 대한 이해는 모델관리의 연구과제를 설정하는 기초가 된다. 모델관리의 필요성과 목적, 컴퓨터기반의 모델링환경과 관련하여 많은 연구과제가 제기될 수 있으나, 다음의 두 가지로 요약할 수 있다.

첫째, 효과적인 모델표현방법과 이의 명세를 위한 모델링언어의 개발이다. 효과적인 모델표현은 모델개발자와 모델사용자 간의 이해와 의사소통을 촉진시켜 모델개발의 신속성과 정확성을 제고시키고, 모델조작을 용이하게 하여 모델의 사용성을 높임으로써 의사결정 능력과 질을 향상시키는 것이 가장 중요한 요소이다. 모델표현의 향상을 위해서는 기존의 데이터베이스지향적인 접근법(E-R 접근법, 관계형 프레임워크 등)에 부가하여 인공지능접근법(논리중심, 의미네트워크중심, 프레임중심 등)과 그래프중심 접근법, 소프트웨어공학적인 접근법(구조적 방법, 객체지향적 방법 등)이 요구된다. 또한 구조적 모델링언어(Geoffrion 1992)와 같은 패러다임 독립적인 모델링언어의 개발과 모델베이스의 조직화를 위한 논리적 스키마의 연구가 필요하다. 특히 객체지향적 모델링 개념(분류화, 모델타입, 일반화와 구체화, 계승, 집합, 멤버십 등)의 도입은 모델의 의미론(semantic)를 더욱 더 증가시킬 것으로 기대된다. 그리고 메타프레임(Krishnan 1991), 메타추상화(Dolk & Konsynski 1984)과 같은 메타지식(meta-knowledge)의 도입과 메타그래프(Blanning 1994), 메타모델링(Muhanna 1994) 개념의 도입은 모델링의 효율성을 보다 향상시키고 보다 더 지적능력을 갖춘 모델관리시스템이 가능하게 한다.

둘째, 성공적인 모델관리시스템의 구축을 위한 방법론의 선택과 모델관리시스템 기능명세 및 설계이다. 우수한 모델관리시스템을 설계하고 구현하기 위해서는 먼저 시스템 개발을 위한 효과적인 방법론의 도입이 필요하다. 그리고 명확하고 다양한 기

능명세와 설계를 위한 지침 및 툴(tools)의 선정이 중요하다. 기존의 구조적 방법론의 문제점을 극복하기 위한 객체지향적 개발방법론의 도입은 성공적인 시스템개발에 많은 시사점을 줄 것으로 기대된다.

- 참고 문헌 -

허순영, "최적화 모델링 언어를 위한 객체지향 모형관리체계의 개발," *經營科學*, 제 11권, 제2호, 1994년 6월, pp.43-63.

Appelgate, L.M., B.R. Konsynski, and J.F. Nunamaker, "Model Management Systems : Design for Decision Support," *Decision Support Systems*, Vol.2, No.1, 1986, pp.81-91.

Bharadwaj, A., J. Choobineh, A. Lo, and B. Shetty, "Model Management Systems: A Survey," in: B. Shetty (ed.), *Annals of Operations Research, Model Management Systems*, Vol.38, 1992, pp.17-67.

Bhargava, H.K., and S.O. Kimbrough, "Model Management, An Embedded Approach," *Decision Support Systems*, Vol.10, 1993, pp.277-299.

Bhargava, H.K., R. Krishnan, and S. Mukherjee, "On the Intergration of Data and Mathematical Modeling Languages," in: B. Shetty (ed.), *Annals of Operations Research, Model Management Systems*, Vol.38, 1992, pp.69-95.

Bhargava, H.K., and R. Krishnan, "Computer-aided Model Construction," *Decision Support Systems*, Vol.9, No.1, 1993, pp.91-111.

Binbasiolu, M. , and M. Jarke, "Domain Specific DSS Tools for Knowledge-based Model Building," *Decision Support Systems*, Vol.2, No.1, 1986, pp.213-223.

Bisschop, J., and A. Meeraus, "On the Development of a General Algebraic Modeling Systems in a Strategic Planning Environment," *Mathematical Programming Study*, Vol.20, 1982, pp.1-29.

Blanning, R.W., "A Relational Framework for Model Management in Decision Support Systems," *DSS-82 Transactions*, 1982, pp.16-28.

Blanning, R.W., "A Relational Framework for Join Implementation in Model Management," *Decision Support Systems*, Vol.1, No.1, 1985, pp.69-82.

Blanning, R.W., "An Entity-relationship Approach to Model Management," *Decision Support Systems*, Vol.2, No.1, 1986, pp.65-72.

Blanning, R.W., "Metagraphs in Model Management," *Paper Presented at the TIMS/ORSA Joint National Meeting, Chicago, May 1993.*

Bonczek, R.H., C.W. Holsapple, and A.B. Whinston, "A Generalized Decision Support System Using Predicate Calculus and Network Database Management," *Operations Research*, Vol.29, NO.2, 1981, pp.263-281.

Chari, S., and R. Krishnan, "Toward a Logical Reconstruction of Structured Modeling," *Proceedings of the 23rd Hawaii International Conference of System Sciences*, Vol.3, (ed.) J. Nunamaker Jr., IEEE Computer Society Press, 1990, pp.524-533.

Chen, P.P.S., "The Entity-Relationship Model -- Toward a Unified View of Data," *ACM Transactions on Data Base Systems*, Vol.1, No.1, 1976, pp.9-36.

Chen, Y.S., "An Entity-relationship Approach to Decision Support and Expert Systems," *Decision Support Systems*, Vol.4, No.2, 1988, pp.225-234.

Chesapeake Decision Sciences, *MIMI/LP User Manual, Ver. 2.63*, New Providence, NJ, 1988.

Choobineh, J., "SQLMP: A Data Sublanguage for Representation and Formulation of Linear Mathematical Models," *ORSA Journal on Computing*, Vol.3, 1991a, pp.358-375.

Choobineh, J., "A Diagramming Technique for Representation of Linear Models," *Omega, International Journal of Management Science*, Vol.19, 1991b, pp.43-51.

Chung, Q.B., and R. O'Keefe, "A Formal Analysis of the Model Management Literature," in: B. Shetty (ed.), *Annals of Operations Research, Model Management Systems*, Vol.38, 1992, pp.137-176.

Collaud, G., and J. Pasquier-Boltuck, "gLPS: A Graphical Tool for the Definition and Manipulation of Linear Problems," *European Journal of Operational Research*, Vol.72, 1994, pp.277-286.

Dempster, M.A.H, and A.M.Ireland, "Object-oriented Model Integration in a Financial Decision Support System," *Decision Support Systems*, Vol.7, No.3, 1991, pp.329-340.

Dolk, D.R., "A Generalized Model Management System for Mathematical Programming," *ACM Transactions on Mathematical Software*, Vol.12, No.2, 1986a, pp.96-126.

Dolk, D.R., "Data as Models : An Approach to Implementing Model Management," *Decision Support Systems*, Vol.2, No.1, 1986b, pp.73-80.

Dolk, D.R., "Model Management Systems for Operational Research: A Prospectus," in: *Mathematical Models for Decision Support*, NATO ASI Series Vol. F48, ed. G. Mitra, Springer, Berlin, 1988a, pp.347-373.

Dolk, D.R., "Model Management and Structured Modeling: The Role of an Information Resource Dictionary System," *Communication of ACM*, Vol.31, 1988b, pp.704-718.

Dolk, D.R., and B.R. Konsynski, "Konwledge Representation for Model Management Systems," *IEEE Transactions on Software Engineering*, Vol.10, No.6, 1984, pp.619-628.

Dolk, D.R., and B.R. Konsynski, "Model Management in Organizations," *Information and Management*, Vol.9, 1985, pp.35-47.

Dolk, D.R., and J. Kottemann, "Model Integration and Theory of Models," *Decision Support Systems*, Vol.9, 1993, pp.51-63.

Dutta, A., and A. Basu, "An Artificial Intelligence Approach to Model Management in Decision Support Systems," *IEEE Computer*, Vol.17, No.9, 1984, pp.89-97.

Elam, J., "Model Management Systems: A Framework for Development," In *Proceedings of 1980 Southwest American Institute for Decision Science*, Decision Science Institute, Atlanta Georgia, 1980.

Elam, J., J. Henderson, and, L. Miller, "Model Management Systems : An Approach to Decision Support in Complex Organization," In *Proceedings of the First Conference on Information Systems*, Chicago, IL:Society for Management Information Systems, December 1980, pp.98-110.

Elam, J.J., and B. Konsynski, "Using Artificial Intelligence Techniques to Enhance the Capabilities of Model Management Systems," *Decision Science*, Vol.18, 1987, pp.487-501.

Fedorowicz, J., and G.B. Williams, "Representing Modeling Knowledge in an Intelligent Decision Support System," *Decision Support Systems*, Vol.2, No.1, 1986, pp.3-14.

Forster,M., "A General Network Generator," in G. Mitra (ed.), *Mathematical Model for Decision Support*, NOTO ASI Series F, Berlin, 1988, pp.207-215.

Fourer, R., "Modeling Language versus Matrix Generators for Linear Programming," *ACM Transactions on Mathematical Software*, Vol.9, 1983, pp.143-183.

Fourer, R., D.M. Gay, B.W. Kernighan, "A Modeling Language for Mathematical Programming," *Management Science*, Vol.36, No.5, 1990, pp.519-554.

Geoffrion, A.M., "An Introduction to Structured Modeling," *Management Science*, Vol.33, No.5, 1987, pp.547-588.

Geoffrion, A.M., "Computer-based Modeling Environments," *European Journal of Operational Research*, Vol.41, 1989, pp.33-43.

Geoffrion, A.M., "FW/SM: A Prototype Structured Modeling Environment," *Management Science*, Vol.37, No.12, 1991, pp.1513-1538.

Geoffrion, A.M., "The SML Language for Structured Modeling," *Operations Research*, Vol.40, No.1, 1992, pp.38-75.

Geoffrion, A.M., "Structured Modeling: A Survey and Future Research Directions," *ORSA CSTS Newsletter*, Vol.15, No.1, Spring 1994, pp.10-20.

Glover, F., D. Klingman, and N.V. Phillips, "Netform Modeling and Applications," *Interfaces*, Vol.20, No.4, 1990, pp.7-28.

Greenberg, H.J., "A Functional Description of ANALYZE: A Computer-assisted Analysis System for Linear Programming Models," *ACM Transactions on Mathematical Software*, Vol.9, 1983, pp.18-56.

Greenberg, H.J., "MODELER: Modeling by Object-driven Linear Elemental Relations," in: B. Shetty (ed.), *Annals of Operations Research, Model Management Systems*, Vol.38, 1992, pp.239-280.

Huh, S.Y., "Modelbase Construction with Object-oriented Constructs," *Decision Science*, Vol.24, No.2, 1993, pp.409-434.

Hürlimann, T., and J. Kohlas, "LPL: A Structured Language for Linear Programming Modeling," *OR Spektrum*, Vol.10, 1988, pp.55-63.

Jones, C., "An Introduction to Graph-based Modeling Systems, Part I: Overview," *ORSA Journal on Computing*, Vol.2, 1990, pp.136-151.

Jones, C., "An Introduction to Graph-based Modeling Systems, Part II:

Graph-grammars and the Implementation," *ORSA Journal on Computing*, Vol.3, 1991, pp.180-206.

Jones, C., "Attributed Graphs, Graph-grammars, Structured Modeling," in: B. Shetty (ed.), *Annals of Operations Research, Model Management Systems*, Vol.38, 1992, pp.281-324.

Ketron Management Science, *DATAFORM User Manual*, McClean, VA, 1987.

Kimbrough, S.O., "A Graph Representation for Management of Logic Models," *Decision Support Systems*, Vol.2, No.1, 1986, pp.27-37.

Kimbrough, S.O., and R.M. Lee, "A Logic Modeling: A Tool for Management Science," *Decision Support Systems*, Vol.4, No.1, 1988, pp.3-16.

Konsynski, B.R., "On the Structure of a Generalized Model Management System," in: *Proceedings of 14th Hawaii International Conference of System Science*, 1980.

Kottemann, J.E., and D.R. Dolk, "Model Integration and Modeling Languages: A Process Perspective," *Information Systems Research*, Vol.3, No.1., 1992, pp.1-16.

Krishnan, R., "Automated Model Construction: A Logic Based Approach," *Annals of Operations Research*, Vol.21, 1989, pp.195-226.

Krishnan, R., "A Logic Modeling Language for Automated Model Construction," *Decision Support Systems*, Vol.6, No.1, 1990, pp.123-152.

Krishnan, R., "PDM: A Knowledge-based Tool for Model Construction," *Decision Support Systems*, Vol.7, No.2, 1991, pp.1301-314.

Lazimy, R., "Knowledge Representation and Modeling Support in Knowledge-based Systems," in: C. Batini(ed.), *Entity-Relationship Approach*, Elsevier Science Pub., North-Holland, 1988, pp.133-161.

Lazimy, R., "A Deductive Approach for Problem Representation and Modeling Support: Conceptual Schema and Object-oriented Models," *Proceedings of the 24th Annual Hawaii International Conference on System Sciences*, Vol. III, IEEE Computer Society Press, Los Alamitos, CA, 1991, pp.291-305.

Lee, R.M., and L.W. Miller, "A Logic Programming Framework for Planning and Simulation," *Decision Support Systems*, Vol.2, No.1, 1986, pp.15-25.

Lenard, M.L., "Representing Models as Data," *Journal of Management Information Systems*, Vol.2, 1986, pp.36-48.

Liang, T.P., "Integrating Model Management with Data Management in Decision Support Systems," *Decision Support Systems*, Vol.1, No.2, 1985, pp.221-232.

Liang, T.P., "A Graph-Based Approach to Model Management," *Proceedings of Seventh International Conference of Information Systems*, 1986, pp.136-151.

Liang, T.P., "Development of a Knowledge-based Model Management System," *Operations Research*, Vol.36, No.6, 1988a, pp.849-863.

Liang, T.P., and C. Jones, "Meta-Design Considerations in Developing Model Management Systems," *Decision Sciences*, Vol.19, No.1, 1988b, pp.72-92.

Liang, T.P., "Analogical Reasoning and Case-based Learning in Model Management Systems," *Decision Support Systems*, Vol.10, 1993, pp.137-160.

Liu, J.I.C., D.Y.Y Yun, and G. Klein, "An Agent for Intelligent Model Management," *Journal of Management Information System*, Vol.7, 1990, pp.101-122.

Lucas, C., and G. Mitra, "Computer-Assisted Mathematical Programming Modeling System: CAMPS," *The Computer Journal*, Vol.31, 1988, pp.364-375.

Ma, P.C., F.H. Murphy, and E.A. Stohr, "Representing Knowledge about Linear Programming Formulation," *Annals of Operations Research*, Vol.21, 1989, pp.149-172.

Mannino, M.V., B.S. Greenberg, and S.N. Hong, "Model Libraries: Knowledge Representation and Reasoning," *ORSA Journal on Computing*, Vol.2, No.3, 1990, pp.287-301.

MathPro, Inc., *MathPro Usage Guide, Introduction and Reference*, Washington, DC, 1990

Maturana, S.V., "Issues in the Design of Modeling Languages for Mathematical Programming," *European Journal of Operational Research*, Vol.72, 1994, pp.243-261.

Minsky, M., "A Framework for Representing Knowledge," in: (eds.) R.J. Brachman, and H.J. Levesque, *Readings in Knowledge Representation*, Morgan Kaufman Pub., Los Altos, CA, 1985.

Mitra, G., C. Lucas, and S. Moody, "Tools for Reformulating Logical Forms into Zero-one Mixed Integer Programs," *European Journal of Operational Research*, Vol.72, 1994, pp.262-276.

Muhanna, W.A., "On the Organization of Large Shared Model Bases," in: B. Shetty(ed.), *Annals of Operations Research, Model Management*, Vol.38, 1992, pp.359-396.

Muhanna, W.A., "An Object-oriented Framework for Model Management and DSS Development," *Decision Support Systems*, Vol.9, No.2., 1993, pp.217-229.

Muhanna, W.A., "SYMMS: A Model Management System That Supports Model Reuse, Sharing, and Integration," *European Journal of Operational Research*, Vol.72, 1994, pp.214-243.

Murphy, F.H., and E.A. Stohr, "An Intelligent Systems for Formulating Linear Programming," *Decision Support Systems*, Vol.2, No.1, 1986, pp.39-47.

Murphy, F.H., E.A. Stohr. and A. Asthana, "Representation Schemes for Linear Programming Models," *Management Science*, Vol.38, No.7, 1992, pp.964-991.

Neustadter, L., A. Geoffrion, S. Maturana, Y. Tsai, and F. Vicuna, "The Design and Implementation of a Prototype Structured Modeling Environment," in: B. Shetty (ed.), *Annals of Operations Research, Model Management Systems*, Vol.38, 1992, pp.453-484.

Potter, W.D., T.A. Byrd, J.A. Miller, and K.J. Kochut, "Extending Decision Support Systems: The Integration of Data, Knowledge, and Model Management," in: B. Shetty (ed.), *Annals of Operations Research, Model Management Systems*, Vol.38, 1992, pp.501-527.

Pracht, W.E., "Model Visualization : Graphical Support for DSS Problem Structuring and Knowledge Organization," *Decision Support Systems*, Vol.6, No.1, 1990, pp.13-27.

Schrage, L., *Linear, Integer and Quadratic Programming with LINDO*, The Scientific Press, Redwood City, CA, 1987.

Shaw, M.J., P.L. Tu, and P. De, "Applying Machine Learning to Model Management," *Decision Support Systems*, Vol.4, 1988, pp.285-305.

Simmons, R.V., "Mathematical Programming Modeling Using MGG," *IMA Journal of Mathematics and Management*, Vol.1, 1986, pp.267-276.

Sivasanakaran, T., and M. Jarke, "Logic-based Formula Management Strategies in Actuarial Consulting System," *Decision Support Systems*, Vol.1, 1985, pp.251-262.

Sklar, M.M., R.A. Pick, and J.R. Evans, "On the Automatic Formulation of LP Problems," *Paper Presented at the ORSA/TIMS Joint National Meeting*, St. Louis, October 1987.

Smith, J.H., and D.C.P. Smith, "Database Abstractions: Aggregation and Generalization," *ACM Transactions on Database Systems*, Vol.2, No.2, 1977, pp.105-133.

Stohr, E.A., and M. Tanniru, "A Database for Operations Research Models," *International Journal of Policy Analysis and Information Systems*, Vol.4, No.1, 1980, pp.105-121.

Tung, L., R.G. Ramirez, and R.D. St. Louis, "Model Integration in an Object-oriented Model Management System," *Proceedings of the 24th Annual Hawaii International Conference on System Sciences*, Vol.III, IEEE Computer Society Press, Los Alamitos, CA, 1991, pp.284-290.

Welch, Jr., J.S., "PAM- A Practitioner's Approach to Modeling," *Management Science*, Vol.33, No.5, 1987, pp.610-625.