

객체지향적 모델관리시스템을 위한 기능요구분석

정 대 울 (동명전문대학 경영정보과)

제 1 장 연구동기

모델링의 영역은 매우 넓고, 다양한 패러다임이 존재하기 때문에 각 영역마다 다양한 형태의 모델이 개발되고 사용되어 왔으며, 이를 관리할 수 있는 매카니즘이나 컴퓨터지원 도구의 개발이 중요시 되고 있다. 이 중 의사결정지원을 위한 모델링 자원을 효과적으로 관리하려는 모델관리시스템(Model Management Systems : MMS)의 연구가 최근 MIS(Management Information Systems) / DSS(Decision Support Systems)의 중요 연구분야로 부상하고 있다. MMS는 모델베이스 내의 모델을 효과적으로 관리하는 기능 즉, 모델의 개발 및 지장, 조작 및 접근, 통제하는 것을 용이하게 하는 소프트웨어이다. 여기서 ①모델개발기능은 문제인식과 문제구조화, 모델링 패러다임의 선정, 모델수립, 모델타당성 검토 등이다. ②모델지장은 모델표현, 모델의 논리적 관점, 물리적 저장을 의미한다. ③모델조작기능은 모델검색, 모델의 선택, 모델의 합성 및 통합, 모델의 구체적인 실현(instantiation), 모델의 순서배열 및 실행 등을 의미한다. ④모델통제기능은 모델베이스의 접근통제, 모델비전관리, 일치성 및 무결성유지 등을 포함한다.

모델관리를 위해서는 다양한 원천의 지식 즉, 경영과학 및 OR(Operations Research), 데이터베이스 시스템, 인공지능 및 전문가 시스템, 소프트웨어공학 등의 지식이 필요하다. 따라서 효과적인 모델관리시스템을 개발하기 위해 여러 분야에서의 연구가 진행되어 왔다. 즉 ① OR문제를 푸는 패키지의 개발, ② 수리적 모델링언어의 개발, ③ 상용화된 DBMS을 환용한 구현, ④ 전문가시스템 셸(shells)을 이용한 구현 등 여러 가지 분야에서의 연구 결과들을 모델관리에 도입하려는 일련의 움직임이 있어 왔다. 하지만 아직도 어떠한 접근법 또는 방법이 성공적인 모델관리를 가능하게 할 것인지는 연구과제로 남아 있다[1].

그런데 지금까지의 모델관리에 대한 대부분의 연구들은 단일 모델의 논리적 스키마 표현(즉, 단일 모델의 논리적 표현을 통해서 이를 컴퓨터 상에 실행하여 그 모델의 해를 구하는 것)에 대한 연구에 주안점을 두어 왔으며, 이 역시 특정 모델링 패러다임(예, LP 모델링)에 한정된 모델을 수립하고 이를 해산출기와 연결시키려는 연구에 그

초점이 맞추어져 있었다. 따라서 특정 모델링 패러다임에 독립적인 모델관리 프레임워크를 개발하는 것이 어려웠으며, 이 프레임워크에 기초를 둔 범용의 모델관리시스템을 개발하는 것에 대한 연구가 미진하였다.

효과적인 모델관리 프레임워크를 개발하고, 이러한 프레임워크를 실현할 수 있는 모델관리시스템을 개발하기 위해서는 무엇보다도 모델관리의 필요성과 컴퓨터 기반의 모델링환경을 이해해야만 한다. 또한 엄격한 모델관리의 원리와 이상적인 모델관리시스템의 특성을 추출하여야 한다. 성공적인 모델관리시스템을 개발하기 위해서는 이러한 기초위에서 모델관리시스템의 기능요구명세가 선행되어야 한다.

따라서 본 연구에서는 기존의 모델관리시스템에 대한 문헌연구들을 기반으로 하여 가장 이상적인 모델관리의 원리와 MMS의 특성을 도출하고, 이를 바탕으로 오늘날 연구의 초점이 되고 있는 객체지향적 모델관리시스템(Object-Oriented MMS)의 아키텍처와 기능요구사항명세를 제시하는 데 그 연구의 목적이 있다.

객체지향적 모델관리시스템은 MMS의 여러 구성요소들, 즉, 의사결정모델, 데이터, 계산출기, 그리고 모델관리에 필요한 각종 지식들을 객체라는 단일화된 방법으로 관리할 수 있게 하므로써 효과적인 모델관리를 가능하게 한다.

제 2 장 모델관리시스템에 대한 선행 연구의 고찰

모델관리시스템이란 용어를 최초로 사용한 사람은 Will[]로 알려져 있다. 그 이후 약 20여년 동안 경영과학 및 정보시스템 분야에서 많은 연구가 이루어져와 이제는 의사결정지원시스템 분야의 중요한 연구영역으로 자리를 차지하고 있다.

Chung & O'Keefe[13]는 모델관리에 대한 기본적인 접근법으로 ①모델/모델링의 인식 접근법 ②데이터-모델 유추 접근법 ③지식중심의 접근법 ④모델표현 접근법 ⑤기타 접근법으로 분류하고 있으며, Bharadwaj 등[4]은 ①대수적 모델링 언어 ②데이터베이스 접근법 ③그래프중심의 접근법 ④지식중심 접근법으로 분류하고 있다. 또한 Blanning[10]은 ①모델베이스구조(네트워크 또는 관계형 데이터베이스 구조와 대비됨) ②모델베이스처리와 인공지능응용 ③MMS의 조직적 환경과 기여라는 측면에서 모델관리의 주요 주제를 분류하고 있다.

모델관리에 대한 연구들을 이들 학자들의 분류에 근거하여 본 연구에서는 다음과 같이 분류한다. 즉, ① 수리적 모델(특히 LP모델)을 계산출 알고리즘(solver)과 연결하여 모델의 해를 구하려는 연구[6][22][51], ② 모델을 데이터로 보고 데이터베이스 기술과 통합하려는 연구[7][8][12][15], ③모델을 지식의 집합체로 보고 지식표현의 기법을 모델

관리에 도입하려는 연구[3][5][11][17][19][20][21][31][38][48][42], ④ 수리적 모델을 그래프로 이용하여 표현하고자 하는 연구[30][37][28][23] 등이 있어 왔다. 그리고 최근에 소프트웨어공학에서 연구의 초점이 되는 객체지향개념을 모델관리에 도입하려는 일련의 움직임이 일고 있다(Mannino 등[43], Lazimy[34], Dempster & Ireland[14], 허순영[2], Potter 등[50], Muhanna[45]).

이들 연구들의 대부분은 이론적 프레임워크만을 제시하고 있을 뿐 실제로 구현된 시스템은 많지 않다. 진정한 모델관리시스템이 되기 위해서는 모델링활동의 전 과정을 지원해야 한다. 일반적으로 모델링활동은 직진한 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 과 실행으로 나눌 수 있다[4]. 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 2-1>과 같다.

<표 2-1> MMS의 모델링 수명주기 지원형태

접근법	MMS	문 제 인 식		모델수립	해 산 출		민감도분석 및 결과해석
		문제추출	모델타입인식		데이터접근 및 모델전환	해산출기 인터페이스	
수리모델링 언어지향	AMPL[22]	기호적인어	사용자	사용자 명세	사용자 프롬프트	yes	yes
	GAMS[6]	기호적인어	사용자	사용자 명세	사용자 프롬프트	yes	yes
데이터베이스 지향	GXMP[15]	메뉴선택	사용자	모델선택	사용자 명령	yes	yes
지식베이스 지향	ACS[52]	na 그래픽	시스템 시스템	모델조작 일차원리	사용자명령 사용자명령	no no	no no
	LIFORM[42]	다이얼로그 문자	시스템 시스템	템플릿선택 모델조작 일차원리,	자동 nd	yes	no
	TIMMS[38]	다이얼로그 다중선택	시스템	모델링규칙 모델조작	사용자 프롬프트	yes	yes
	PDM[33]						
AIMM[41]							
그래프지향	GBMS[28]	폴다운메뉴 및 아이콘	사용자	사용자 명세	자동	yes	no
	FW/SM[25]	폴다운메뉴	사용자	사용자 명세	자동	yes	yes
객체/시스템 지향	SYMMS[46]	다이얼로그	시스템	모델조작	자동	yes	no

* na=not applicable, nd=no discussed.

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.

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[48], Binbasioglu & Jarke[5], LPFORM(Ma 등)[42], PDM(Krishnan)[33] 등을 들 수 있다.

템플릿 채우기 기법은 주어진 문제를 모델베이스 내의 한 모델템플릿과 대응시켜 모델을 수립하는 방법으로 메넌 완전히 새로운 모델을 수립해야 하는 번거러움을 든다. 이 기법을 사용하는 시스템으로는 LPFORM과 Mannino 등[43] 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[52], TIMMS[38], AIMM[41], SYMMS[46]등을 들 수 있다.

모델수립을 위해 최근 주목을 받는 방법으로 Liang[39]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 계산출

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

또한 특정 타입의 문제를 다른 타입으로 바꾸는(예를 들어 네트워크문제를 LP문제로 전환) 메카니즘을 MMS 내에 갖출 필요가 있다. 이것은 동일한 문제를 여러 가지 모델 형태로 볼 수 있게 하며 다양한 형태의 계산출기를 이용할 수 있게 한다. Dolk & Kongsynski[17]가 제시하는 추상화공간(abstraction space)은 이러한 문제를 다루는 데 적합하다.

데이터접근은 시스템에 따라 다양하다. 자동적 데이터접근이 가능한 시스템은 사용자는 데이터베이스 조작과 계산출기와의 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. 그래프 접근법을 택하는 대부분의 시스템들은 자동적 데이터접근 기법을 택한다. 지식중심의 시스템은 데이터접근방법에 상당한 차이를 보이고 있다.

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

4) 모델 해석

이 단계는 모델가정의 검토, 민감도 분석, 필요시 모델의 수정 등의 활동이 포함된

다. 많은 대수적 모델링언어들이 민감도 분석을 위한 메카니즘을 제공하고 있다. ANALYZE[22]는 민감도 분석뿐만 아니라 모델문서화, 타당성검토, 그리고 결과해석 등을 수행할 수 있다. 기존의 MMS들의 대부분이 가정검토와 모델수정을 지원하지 못하고 있다. 이러한 문제를 다루는 데 있어 주목할 만한 연구는 AMPL의 도메인 체크링(domain checking) 기능으로 테이블의 열과 행의 이름이 그 테이블에 선언된 집합(SETS)인가를 체크한다. 모델수정을 직접적으로 지원하는 시스템으로 Muhanna[46]의 SYMMS를 들 수 있다. 그는 모델비전개념을 도입하여 임시적, 잠정적 모델을 효과적으로 관리할 수 있는 메카니즘을 제시하고 있다.

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

제 3 장 모델관리의 필요성 모델링 환경

1. 모델관리의 필요성

모든 의사결정자의 공통적인 특성은 데이터를 수집하고, 이 데이터를 분석하여 최종적으로 선택을 하는데 있어 “모델”을 사용한다는 점이다. 이러한 의사결정 모델은 직관적일 수도 있으며, 또는 기호로 정식화되어 구체적일 수도 있다.

문제해결과정에서 적절한 의사결정지원모델의 사용은 주어진 문제에 대한 이해를 높여주고, 많은 대체안들을 체계적으로 검토할 수 있는 수단을 제공하므로써 의사결정자의 성과를 높여준다[26]. 여기서 모델이란 실세계(real world)에 대한 추상화(abstraction)로서, 사용자들로 하여금 실세계에 대한 추론을 가능하게 하여 의사결정을 원활하게 해준다.

조직에서 데이터관리의 중요성이 부각되고, 데이터가 조직의 자원으로 인식됨에 따라 효과적인 데이터베이스관리시스템(Data-base Management Systems: DBMS)의 개발이 기업의 정보시스템개발에 결정적인 역할을 하였다. 이와 마찬가지로 합리적, 과학적 의사결정을 효과적으로 수행하기 위해서는 모델의 사용이 필수적이며, 이러한 모델을 기업의 자원으로 인식하여 이를 통제하고 관리할 필요가 있다. D.R. Dolk[16]는 기업 내에서 모델이 안정적으로 사용되고, 또한 공유가능한 자원이라는 인식이 점차 확산되자

기업 전체의 정보관리라는 입장에서 총체적 모델링자원의 통제 중요성을 강조하고 모델관리가 그 핵심적인 요소라고 주장하였다.

데이터관리가 데이터자원의 통제와 사용에 있어서 여러 가지의 비효율성에 대처하기 위해서 필요하듯이 모델링환경에서도 모델의 ①중복성, ②불일치성과 무결성(integrity) ③공유의 결여, ④표준화, ⑤물리적/논리적 독립성 등의 문제가 발생할 수 있으므로 이러한 문제를 효과적으로 처리하기 위해서는 체계적인 모델베이스 조직화가 필요하다.

모델링환경에서 중복성(redundancy)은 동일한 모델이 여러 사람에 의해 서로 독립적으로 개발되어 여러 개의 버전(versions)이 존재하는 경우이다. 이 경우 사전에 미리 계획되어 상이한 모델이 특정 문제에 대해서 여러 시각에서 볼 수 있게 한 것이 아니라면, 대개의 경우 문제를 분명히 하기보다는 혼돈을 가져올 수 있다. 유사한 모델에 상이한 가정을 했을 경우 이들 모델로부터 나온 결과를 해석하는 방법에 지대한 영향을 미칠 수 있다.

모델간의 불일치(inconsistency)는 모델의 정확성, 타당성, 무결성에 문제를 일으킬 수 있다. 동일한 데이터를 사용한다는 보장이 없는 상황에서 모델을 평가·비교하는 문제라든가 또는 더 나아가 특정 모델의 기본가정을 밝히거나 분석하는 일은 잘 정립된 모델링환경에서조차도 힘든 것이다. 그러나 PC를 이용한 모델링활동이 조직 내에서 널리 흩어져있는 경우 특히 심각한 문제를 일으킬 수 있다. 또한 이 경우 적절한 통제 메카니즘을 강구하기도 어렵다.

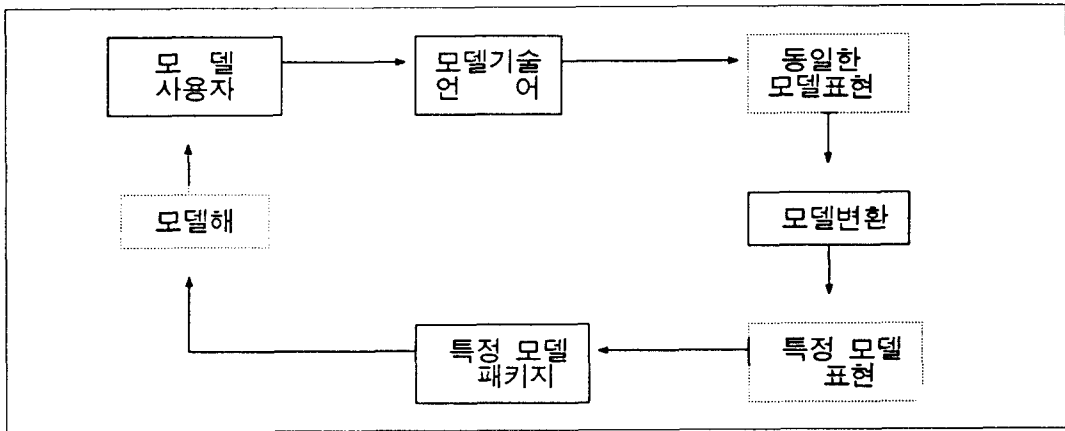
자원공유의 결여(lack of resource sharing)는 현재 모델링환경의 주된 허점이다. 통제되지 않은 최종사용자에 의한 개인별 모델링활동은 각 개인의 생산성을 증가시키는데는 기여하지만 조직 전체적인 입장에서 본다면 상당한 기회비용이 숨어있다. 개인별 생산성을 향상시키기 위해 개발된 의사결정 모델들은 만들어져 한번 사용되고는 버리는 일회용품인 것이 많다. 이것은 조직 전체적인 입장에서 볼 때 분명히 최적일 수가 없다. 어떤 의사결정상황에서 어떤 한 모델이 유용하다고 입증되면, 이 모델은 유사한 상황에서 반복적으로 활용될 수 있다. 따라서 유사한 의사결정상황에서 가능하다면 현존하는 유용한 모델을 개조하는 것이 처음부터 다시 만드는 것보다는 나은 것이다. 또 하나 중요한 사실은, 모델은 이전가치(transfer value)를 가질 수 있어 어떤 한 상황에서 개발된 모델이 역시 다른 상황에 적용될 수 있다.

표준화(standardization)는 중복성 및 불일치성과 직결된 문제이다. 데이터베이스 관리분야에서 데이터요소의 표준화는 모든 사용자가 동일한 방법으로 데이터에 접근하도록 해준다.

모델관리를 데이터관리와 동일 선상에 놓고 볼 때, 또 하나 중요한 문제는 논리적

관점 對 물리적 관점의 독립성(independence of logical model and physical model)이다. 데이터베이스환경에서 독립성이란 데이터 실체와 관계(data entity and relationship)를 데이터의 물리적 저장구조에 상관없이 논리적으로 정의할 수 있음을 의미한다. 모델독립성은 모델의 논리적 표현과 물리적 표현간의 독립성 또는 모델기술(model description)과 모델의 解(model solution)를 구하는 작업이 분리된 기능이다. 이러한 모델독립성 개념을 도식하면 <그림 3-1>과 같다. 모든 모델 사용자는 모델을 동일한 記述言語(description language)를 사용하여 표현하고, 이렇게 표현된 모델은 해를 구하기 위해서 특정 모델링 틀에 맞도록 내부적으로 변환하는 능력을 모델관리시스템이 가져야 한다.

<그림 3-1> 모델독립성



이상에서 언급한 것 외에도 모델과 다른 정보자원, 특히 데이터와의 통합문제는 매우 중요하다. 모델은 데이터를 사용하기 때문에 모델에 필요한 데이터자원에 접근하기 위해서는 기존의 데이터베이스와의 통합이 중요한 연구과제이다[36].

따라서 체계적인 모델베이스 조직화는 조직의 모델링 자원의 중복성을 줄이고 모델의 일관적 사용과 통제를 가능하게 할 뿐 만 아니라 의사결정 문제영역이 확대 될 경우(이는 주로 상위계층으로의 의사결정 요구가 확대될 경우 나타난다) 이미 모델베이스에 존재하는 기존 모델을 이용하여 새로운 모델을 통합하거나, 문제영역의 축소에 따른 모델의 분리활동을 돕는다. 또한 유사한 의사결정환경에서는 비슷한 유형의 의사결정 모델이 필요한 경우가 있으므로 모델의 재사용을 가능하게 한다.

이상의 모델베이스 조직화의 필요성 비추어 볼 때, 모델베이스 조직화의 주된 목적은 의사결정자의 의사결정능력과 질을 향상시키고, 모델링비용을 절감시키기 위해서

조직 내에서 사용되는 모델들을 효과적으로 통합하여 모델베이스 내에 저장하는 데 있다.

2. 컴퓨터기반의 모델링 환경

의사결정모델을 효과적으로 활용하기 위해서는 컴퓨터 기반의 모델링 툴(tools)의 개발이 무엇보다도 중요하다. 컴퓨터 기반의 모델링환경을 잘 구축함으로써 의사결정의 생산성을 향상시킬 수 있으며, 의사결정의 질을 높일 수 있다. 그리고 MS/OR 전문가와 의사결정자와의 연결을 자연스럽게 해주어 의사결정시 모델사용을 촉진시킬 수 있다. 이러한 잇점을 얻기 위해서는 다음과 같은 바람직한 모델링환경의 특성들이 요구된다 [24].

- ① 컴퓨터의 의한 모델링 수명주기의 전 단계를 지원할 수 있어야 한다.
- ② MS/OR 전문가가 사용할 수 있는 것이 아니라, 의사결정자 또는 정책결정자에게 사용이 용이해야 한다.
- ③ 그 모델링 환경에서 만들어진 모델이나 시스템의 지속적인 발전이 용이해야 한다.
- ④ 모델정의를 위한 단일의 패러다임 중립 언어(paradigm-neutral language)가 있어야 한다.
- ⑤ 모델관리의 주요 자원(모델, 데이터, 해산출기(solvers), 모델관리지식)의 관리기능이 있어야 한다.

지금까지의 수리적 모델링분야에서 주된 연구는 효율적인 해산출 기법의 개발과 구현에 집착한 나머지 조직내의 전체적인 모델링과정을 관리하는 것에 대한 연구가 미진하였다[15]. 모델링 활동은 주로 수작업에 의해 단편적으로 이루어졌기 때문에 모델링 생산성의 저하와 의사결정자(사용자)의 모델 수용이 낮았다[23]. 이에 모델링과정의 전 단계를 컴퓨터를 통한 지원의 중요성이 부각된다. 또한 모델링환경은 고차원의 소프트웨어 통합이 필요하다. 모델링 환경은 데이터 원천, 모델, 해산출기, 그리고 이들로부터 나온 결과들을 저장한 라이브러리(libraries)들 간의 연결기능을 지녀야 한다.

진정한 모델링환경이 되기 위해서는 모델링 전문가에게 뿐만 아니라 모델링 작업에 관여하는 모든 사람들이 그것을 사용하는 것이 용이해야 한다. 이를 위해서는 모델 표현의 간결성과 직관적인 조직화, 모델링언어를 익히고 사용하는 것이 용이해야 하며, 비전문가나 자주 사용하지 않는 사용자들을 위한 특별하고 완전한 접근경로를 제공할 수 있어야 한다. 또한 사용자가 작성한 모델링언어는 컴퓨터 상에서 곧바로 실행가능해야 한다. 즉, 모델링 전문가가 아닌 사람도 약간의 훈련으로 모델링언어를 쉽게 이해

할 수 있어야 하며, 컴퓨터가 이해할 수 있는 형태로 프로그램 될 수 있도록 공식적 구조를 가져야 한다. 이를 위해서는 모델링언어 자체에서 현실문제에 내포된 충분한 정보를 표현할 수 있는 강력한 표기법과 컴퓨터 내에서의 강력한 완전성 및 일관성 체크 기능이 제공될 수 있도록 모델관리시스템이 설계되어야 한다.

모델링환경은 또한 발전적 융통성(evolutionary flexibility)을 지녀야 한다. 즉, 그 모델링환경에서 만들어진 것을 변경하기가 용이해야 하며, 심지어는 모델링환경 그 자체도 변경이 쉬워야 한다. 왜냐하면, 아무리 모델링 전문가라도 단번에 100% 완전한 모델시스템 또는 모델기반의 시스템을 만들 수 있는 사람은 거의 없다. 이것이 만일 가능하더라도 의사결정 요구사항이 시간이 갈수록 변하므로 변경은 불가피하다. 이러한 모델링환경의 특성이 또 한편으로는 실행가능 모델링언어를 필요하게 하며, 모델관리에서 버전관리(version management)의 중요성을 시사한다.

모델정의를 위해서는 특정 문제영역이나 모델링 패러다임, 또는 해산출기에 관계없이 폭넓게 적용될 수 있는 공통된 언어가 있어야 한다. 왜냐하면, 현재의 모델링 상황에 비추어 볼 때 모델표현에 있어 패러다임에 국한된 스타일(paradigm-specific style)이 만연되고 있어 여러 가지의 모델표현 스타일(예, 의사결정 트리, 네트워크, 마야코브체인, 수리계획, 대기행렬 등)이 무수히 많이 존재함에 따라 모델링 전문가와 사용자간의 의사소통은 물론이고 심지어는 다른 하위필드에 속한 전문가들 사이에도 의사소통에 크다란 장애요인이 된다. 물론 그 원인은 일면에선 표현의 명확성과 효율성을 위한 이유도 있으나, 주된 원인은 모델링 전문가의 임의적인 선택, 그 사람의 과거 경험에 의한 우연적 선택, 표준화의 결여 등에 있다고 볼 수 있다. 이러한 표현 스타일의 다양성은 모델과 시스템의 통합에 있어 기술적 장애물이다. 따라서 패러다임 중립 언어의 기초가 되는 개념적 모델링의 일반적인 프레임워크를 모델링 환경은 가져야 하며, 언어 자체는 최소한의 특수훈련을 받은 사람이면 이해할 수 있어야 하며 컴퓨터 상에서 실행 가능하도록 공식적 구조를 지녀야 한다.

따라서 모델링환경은 데이터, 모델, 해산출기, 그리고 모델관리지식 등과 같은 모델링자원의 축적, 공유, 재사용을 위한 관리기능이 제공되어야 한다.

제 4 장 모델관리시스템의 특성과 모델관리원리

1. 이상적인 모델관리시스템의 특성

이상에서 모델관리의 필요성과 바람직한 모델링환경에 비추어 볼 때 이상적인 모

델관리시스템은 다음과 같은 요건들이 갖추어져야 한다.

① 엄격한 이론적 기반을 가진 모델링 프레임워크를 가지고 사용자가 쉽게 자신의 관점을 정확히 반영할 수 있는 모델표현기법 및 도구를 제공해야 한다. 물론 모델표현의 일반성과 비절차성을 높이고, 개념적 모듈성과 모델독립성을 강력히 지원할 수 있어야 하며, 표현된 모델의 자동적 문서화를 가능하도록 해야 한다.

② 컴퓨터의 의한 모델링 수명주기의 전 단계를 지원할 수 있어야 하며, 모델관리의 주요 자원(모델, 데이터, 해산출기, 모델관리지식)을 효과적으로 관리할 수 있어야 한다. 또한 그 모델링 환경에서 만들어진 모델의 지속적인 발전을 용이하게 해야 한다.

③ 모델정의를 위한 단일의 패러다임 중립 언어(paradigm-neutral language)가 있어야 하며, 그 언어는 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 모델링활동은 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.

따라서 이상적인 모델링 시스템이 되기 위해서는 아마도 그래픽(아이콘), 네트워크, 그리고 대수적 표현기법이 포함된 한 가지 이상의 외부적 모델표현을 수용할 수 있어야 할 것이다. 또한 MMS는 특정 모델링 패러다임만을 지원하기 보다는 일반성이 높은 시스템을 지향해야 하며, 사용자가 복잡한 문제를 보다 더 쉽고 정확하게 모델링 할 수 있게 하고, 모델의 유지보수를 용이하게 하는 표현법을 가져야 할 것이다.

2. 모델관리의 원리

모델관리시스템의 목적은 인간의 의사결정과정에 필요한 모델을 개발하고 이용하는 것을 지원하는데 있다. 따라서 모델관리시스템을 개발하는데 있어 인간의 인지과정을 고려해야 한다. 인간의 인지과정에서 주된 관심거리는 ①인간의 한계, ②인간의 정보저장과 검색, ③인간의 문제해결전략이다. 인간은 완전한 정보처리자가 아니므로 이러한 인간의 한계를 극복하기 위해서 많은 전략들이 사용되고 있다. 예를 들면 복잡한 문제를 하위문제로 분해한다거나 만족할만한 해를 찾는데 있어 경험법칙(rules of

thumb)을 이용하기도 한다. 인간의 의사결정을 지원하기 위해서 모델관리시스템은 이러한 한계를 극복해야만 한다. 예를 들어 모델관리시스템은 모델의 체계적인 저장과 검색을 위해서 대규모의 모델을 여러 개의 작은 구성요소모델로서 나눌 수 있어야 한다. 그리고 여러 개의 작은 모델들은 하나의 큰 모델을 만드는데 통합될 수 있어야 한다. 즉, “모델분해”와 “모델통합”을 지원해야 한다. 본 연구에서 제시하는 객체지향적 모델관리는 이러한 개념을 자연스럽게 지원해 줄 수 있는 메카니즘을 지니며, 모델관리시스템 설계시 이러한 개념이 반영되어진다.

효과적인 모델관리를 위해서는 입직한 모델링 방법에 기초한 모델관리의 원리가 필요하다. 따라서 본 연구에서는 지금까지 제시된 여러 가지 모델관리 접근법의 문제점들을 극복하고, 이상적인 모델관리시스템의 특성들을 실현하기 위해서 다음과 같은 모델관리의 원리를 제시한다. 본 연구에서 제시하는 모델관리의 원리는 객체지향적 방법론과 시스템이론에 그 기초를 두고 있다.

(1) 사용자 지향적 모델링

모델링의 출발점은 의사결정자의 문제해결을 위한 시도에서 시작된다. 문제해결을 위해서는 의사결정자가 당면한 문제를 정확히 정의하고 이를 구조화해야 한다. 문제 구조화는 특정 문제영역에서 그 문제의 구성요소, 즉 의사결정변수 및 제약사항 등을 식별하고 이들 구성요소들이 어떻게 서로 상호작용하는지를 명백히 밝히는 작업이다. 이러한 문제구조화를 위해서 의사결정자의 문제에 대한 의미(semantics)를 정확히 표현할 수 있는 툴(tools)의 사용이 매우 중요하다.

(2) 두 가지 수준의 모델링

모델링은 개념적 모델링(conceptual modeling)과 논리적 모델링(logical modeling)으로 나눈다. 개념적 모델링은 사용자 관점에서의 모델링으로 MMS의 모델명세 요구와는 독립적이다. 이 때 만들어진 모델은 개념적 스키마(conceptual schema)라 부른다. 논리적 모델링은 특정의 MMS가 곧바로 실행가능한 모델포맷을 생성하는 과정으로 종래 대부분의 모델링언어들이 이 범주에 속한다. 이 때 만들어진 모델을 논리적 스키마(logical schema)라 부른다. 여기서 중요한 것은 개념적 스키마와 논리적 스키마를 자동적으로 연결하기 위한 메카니즘이 필요하다.

(3) 여러 수준의 추상화

모델베이스를 구성하는 모델은 그 추상화수준에 따라 여러 수준으로 표현되어진다. 각 수준의 추상화는 특정 관점에 상응된다. 즉, 모델수립자는 일반적 문제클래스를 모

델화한 일반적 범용모델 스키마(general model schema)에 관심을 가질 것이며, 사용자(의 사결정자)는 특정문제를 위해 제작된 특수모델 스키마(customized model schema)에 관심을 가질 것이다.

또한 모델스키마 명세는 그 관점에 따라 외부 명세와 내부 명세로 나눈다. 즉, 외부(인터페이스)명세는 모델타입(model-type)으로 구성되며, 내부(구조적이고 행위적)명세는 모델버전(model-version)으로 구성된다. 이것은 하나의 시스템을 두 가지 수준의 추상화(외부 및 내부)로 보는 관점과 일치한다. 따라서 하나의 주어진 모델타입에 대해 상이한 내부구조와 행위를 관련 지을 수 있다. 하나의 공통된 인터페이스를 공유(계승)하는 여러 가지의 내부명세들은 그 모델타입을 실현한 버전(version)들이다.

(4) 모델독립성(model independence)

모델독립성에는 모델-데이터 독립성(model-data independence)과 모델-해산출기 독립성(model-solver independence)이 있다. 모델-데이터 독립은 모델스키마(모델클래스의 일반적 구조)를 그 스키마의 변수와 계수를 인스턴스한 데이터 값의 집합으로부터 완전히 분리한다. 이것은 모델스키마를 변경하지 않고도 데이터 값, 크기, 저장구조, 또는 접근방법 등을 변경할 수 있음을 의미한다. 이러한 특성은 동일한 모델스키마를 다양한 원천으로부터 올 수 있는 데이터집합에 재사용이 가능하게 한다. 또한 하나의 데이터집합에 여러 개의 모델이 재사용되게 하며 저장상의 중복성을 줄인다. 따라서 모델에 사용되는 데이터는 독립된 데이터베이스에 저장하여 관리할 수 있도록 한다.

모델-해산출기 독립은 모델명세(모델스키마)를 해산출기와 분리시킨다. 이것은 하나의 모델이 여러 개의 해산출기를 가지거나 또는 하나의 해산출기가 여러 개의 모델인스턴스를 풀 수 있게 한다.

(5) 통합된 객체관리

모델관리에 필요한 의사결정모델, 데이터, 지식, 해산출기 등을 각각 객체타입으로 보고 이들에 대해서 동일한 객체표현법을 적용하므로써 저장 및 관리 메카니즘의 통일과 이들 간의 통합을 용이하게 한다.

(6) 지식중심의 모델링 및 모델관리

모델수립과 모델적용을 위해서는 MS/OR 전문가의 전문지식을 필요로 한다. 또한 대규모의 모델베이스에 저장된 모델 및 모델관련정보를 찾는 데도 많은 모델관리 지식이 필요하다. 이를 위해서 지식중심의 모델링시스템과 모델상담시스템을 개발할 필요가 있다.

(7) 소범위 모델링과 대범위 모델링의 분리

소범위 모델링(modeling- in-the-large)은 문제-모델 연결, 즉 주어진 의사결정목표에 대한 개별 모델의 개념화, 모델수립, 다당성에 초점을 맞춘다. 구조적 모델링을 제외한 대부분의 이 분야에 대한 연구들이 특정 페리다임이나 특정 영역에 한정된 연구들이다 (예, 모델링 언어, 모델분석시스템, LP모델수립시스템 등). 반면 대범위 모델링 (Modeling-in-the-small)은 광범위하고 장기적 안목의 모델관리문제에 초점을 둔다. 즉 모델을 조직의 자원으로 보고 이에 대한 계속적인 사용과 통제에 주된 관심을 갖는다. 특히, ① 대규모의 공유된 모델의 조직화와 관리, ②현재 제사용가능한 모델요소들로부터 모델의 합성(모델-모델 연결), ③ 모델-데이터 연결, 해산출기 통합, 모델 진화와 구성관리에 초점을 둔다.

이상에서 제시된 ①모델관리의 필요성, ②바람직한 모델링환경, ③이상적인 모델관리시스템의 특성, ④모델관리의 원리는 효과적인 모델관리 프레임워크의 개발을 위한 기초가 되며, 모델관리시스템의 기능요구사항명세와 성공적인 모델관리시스템 개발의 기반이 된다. 또한 개발된 모델관리 프레임워크와 모델관리시스템을 평가하는 기준이 된다.

제 5 장 모델관리시스템의 아키텍처와 기능요구사항 명세

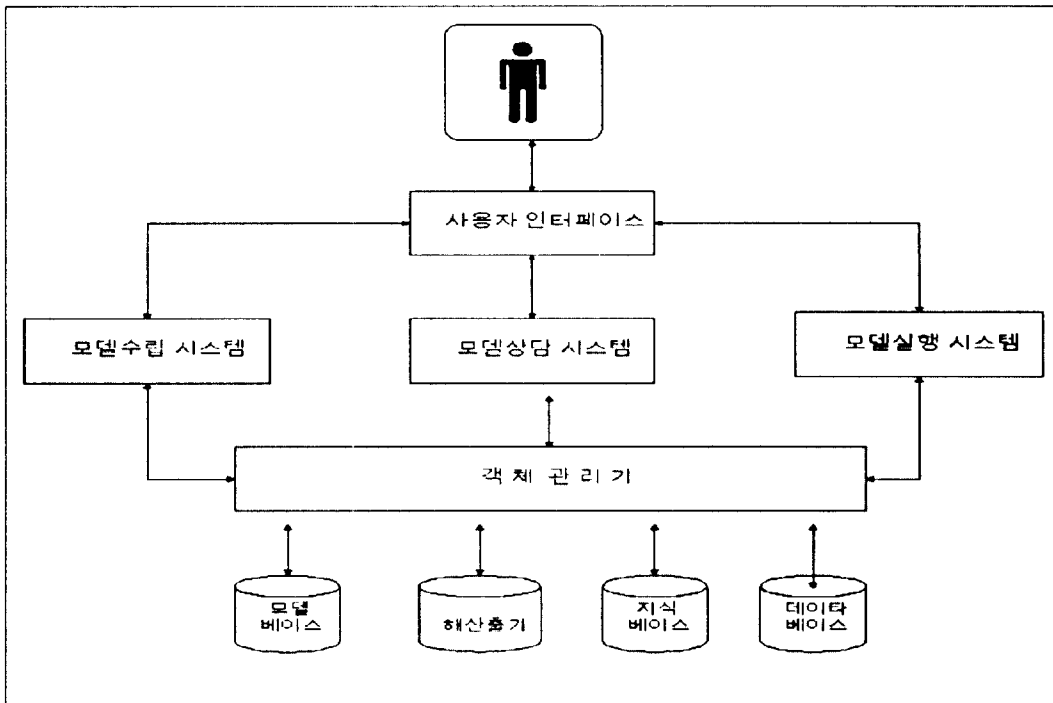
1. 시스템 아키텍처

앞에서 제시한 여러 가지의 모델관리원리를 실현하기 위해서 <그림5-1>과 같은 객체지향적 모델관리시스템(Object-Oriented Model Management System : OOMMS) 아키텍처를 제시한다. OOMMS는 모델베이스(MB), 해산출기(solver), 지식베이스(KB), 데이터베이스(DB)를 관리하는 객체관리기(object manager), 모델수립활동을 돕는 모델수립 하위시스템(model building subsystem), 모델수립과 모델실행을 위해 적절한 모델을 선택하는 것을 돕는 모델상담 하위시스템(model consultation subsystem), 선택된 모델의 인스턴스화와 실행(최적화, 민감도분석, 목표탐색 등)을 가능하게 하는 모델실행 하위시스템(model execution subsystem), 그리고 사용자 인터페이스(user interface)로 구성된다.

객체관리기는 모델관련정보에 대한 관리기능과 모델링 관리기 및 모델사용 관리기의 요구에 대한 지원기능을 수행한다. 사용자 인터페이스는 사용자와 시스템 간의 상호작용을 매개한다. 이것은 시스템에 대한 통합된 편집만을 사용자에게 보이며 복잡한 내부구조는 숨긴다.

일반적으로 MMS의 사용자는 모델사용자, 모델개발자, 모델구현자(model implementor)를 지원할 수 있다. 모델사용자는 의사결정자 또는 그 중개자로서 자신의 문제해결을 위한 도구로서 모델을 수립한다. 따라서 이들은 주로 모델인스턴스를 만들고 모델실행시스템을 통하여 그 모델의 해를 구하고 민감도분석과 같은 작업을 수행한다. 따라서 모델사용자는 모델의 구현보다는 모델의 가정, 기대되는 행위, 적절한 결과 등에 관심을 가질 것이다. 모델개발자는 경영과학자 또는 OR 전문가로서 의사결정자의 문제를 분석하고 적절한 모델링 방법론을 사용하여 타당한 모델을 결합한다. 모델구현자는 일종의 프로그래머로서 모델개발자가 제공한 모델명세 및 계산출 알고리즘을 컴퓨터 상에서 실행가능한 코드로 전환하고 그것의 정확성을 검토한다. 이들 외에도 데이터베이스나 모델 및 지식베이스를 정의하고 관리하는 이들도 있다. 그런데 이들이 별개의 사람일 수도 있고 또는 한 사람이 여러 가지의 역할을 수행하기도 한다.

<그림 5-1> OOMMS 아키텍처



2. 모델수립 및 상담시스템

모델수립시스템은 모델개발자가 새로운 모델타입을 정의하고, 새로운 원소 및 복합

모델버전을 만들고 현재의 모델타입과 버전을 갱신하는 활동, 즉 모델수립과 유지활동을 지원하는 시스템이다. 현존 모델베이스 내에 있는 모델타입이나 모델버전을 검색하고 저장하기 위해서는 객체관리기의 도움을 받는다. 이를 구현하기 위해서는 표준 문서 편집기 뿐만 아니라 그래픽 편집기가 필요하다. 왜냐하면 개념적 수준의 모델을 만들고 편집하기 위해서는 그래프지향적인 방법이 좋으므로 이를 지원하기 위해서는 그래픽 편집기가 필요하며, 논리적 수준의 모델을 작성하고 편집하기 위해서는 문서편집기가 효과적이다. 따라서 효과적인 모델수립 시스템이 되기 위해서는 이러한 두 종류의 편집기가 필요하다.

한편 모델개발자의 모델수립 활동을 돕고, 모델사용자가 적절한 모델을 선택하는 것을 돕기 위해서는 모델상담시스템이 필요하다. 즉 모델개발자가 새로운 모델을 수립하기 전에 그가 모델링하고자 하는 문제에 적합한 모델이 현재의 모델베이스에 있는지를 찾아볼 수 있게 하며, 새로운 복합모델을 만들고자 할 경우 이에 이용될 수 있는 원소모델이 어떤 것이 있는지를 조회할 수 있어야 한다. 또한 모델사용자가 자신의 문제를 해결하기 위해서 이용가능한 모델이 어떤 것이 있는지를 조회하고 최선의 모델을 선택할 수 있게해야 한다. 이를 위해서는 문제영역별, 모델타입별로 각각 이용가능한 모델을 카탈로그 할 수 있게 하거나, 모델개발자 또는 사용자가 기술한 문제내용을 분석하여 그 문제상황에 가장 적절한 모델타입을 제시하는 것이다. 또한 여러 가지 방법으로 모델수립자가 원하는 모델 또는 관련성이 있는 모델을 찾기 위해 필요한 정보를 받아들이고 현재의 모델베이스와 지식베이스를 탐색하는 작업과 이들을 서로 연결할 수 있는 매카니즘이 갖추어져야 한다.

이러한 모델수립과 모델링 상담활동을 지원하기 위해서는 인공지능기법의 도입이 필요하다. 만일 적절한 모델이 없는 경우에는 모델수립자가 처음부터 새로운 모델을 만들어야 한다. 이 경우 지식베이스 내에 있는 특징의 모델링 패러다임이나 문제영역에 대한 모델링 지식이 환용될 수 있다. 또한 모델베이스 내에 있는 원소모델을 결합하여 복합모델을 만들 경우 그 원소모델에 대한 정보와 모델결합에 대한 지식이 이용될 수 있다. 그런데 현실 세계의 모델은 매우 복잡하며 현재의 인공지능기술에는 한계가 있으므로 이러한 개념을 완전히 실현하기란 어렵다. 특히 모델-영역독립의 대규모 범용 MMS를 구현하는 데는 더욱 더 어렵다. 하지만 특정 분야에 한정된 MMS를 구현하는 데는 인공지능개념의 적용이 매우 성공적인 경험을 가지고 있다. 이런 점에서 Liang[38]의 연구는 자동적 모델수립을 위한 좋은 출발점이 된다.

따라서 모델수립시스템은 다음의 기능을 갖추어야 한다. ① 개념적 모델링 활동을 지원하는 그래픽 편집기는 모델링에서 사용되는 여러 가지의 객체표현 아이콘을 제공해야 하며, 모델수립자는 특정 모델을 구성하는 객체를 표현하기 위해서 그에 상응하는

아이콘을 선택하고 필요한 문서화를 수행할 수 있어야 한다. ② 또한 특정의 실행가능한 모델링언어로 만들어진 논리적 스키마를 직접 작성하거나 필요시 편집할 수 있어야 한다. ③ 그래픽 편집기를 통해서 작성된 개념적 모델스키마는 특정 모델링언어의 논리적 모델스키마로 전환할 수 있어야 한다. 이를 위해서는 스키마 전환기를 갖추어야 한다.

또한 모델상답시스템은 다음의 기능을 갖추어야 한다. ① 문제타입, 모델링 방법론 및 이론, 또는 조직상의 기능과 계층에 따라 이용가능한 모델을 카탈로그 할 수 있는 다양한 계층적 인덱스를 통해서 상호대화식으로 열거해 볼 수 있는 메뉴형태의 인터페이스를 제공할 수 있어야 한다. ② 계승그래프(inheritance graph)에 나타난 이용가능한 모델타입과 모델버전 간의 관계를 볼 수 있어야 한다. 즉, 특정 일반모델타입을 구체화한 모든 의사결정모델타입을 리스트하고, 주어진 모델타입에 상응하는 이용가능한 모델버전들을 기술하고, 그리고 어떤 모델버전이나 모델타입을 직접적으로나 또는 간접적으로 구성요소로 하여 만들어진 모든 복합모델버전들을 찾을 수 있어야 한다. ③ 주어진 입력들로부터 어떠한 모델을 사용하여 어떤 출력이 나올 수 있는가를 제시할 수 있어야 하며, 주어진 출력을 내기 위해서 어떤 입력이 필요하고 어떠한 모델이 필요한지를 제시해야 할 수 있어야 한다. 이를 위해서는 진방 또는 후방추론 매카니즘이 있어야 한다.

3. 모델실행시스템

모델실행시스템은 모델을 사용하여 사용자의 의사결정과정을 향상키는데 주목적이 있다. 모델상답시스템을 통하여 의사결정자 또는 사용자가 원하는 적절한 모델을 선택하고 나면, 사용자는 모델을 실행시키기 위해서 이 시스템을 사용한다. 따라서 모델실행시스템은 ① 데이터베이스나 사용자로부터 데이터를 받아들여 모델인스턴스를 만들고, ② 이것의 해를 구하기 위해서 해산출기와 인터페이스하며, ③ 최적 또는 최선의 대안(optimization or best alternative)을 제시하며, ④ 구해진 해에 대해 "what-if"분석(민감도분석)을 수행할 수 있게 하며, ⑤ 주어진 출력 값을 얻기 위해 의사결정변수의 값이 어떻게 바뀌어야 하는 가를 탐색하는 목표탐색(goal-seeking) 기능이 있어야 한다.

모델에 필요한 계수와 파라메트 값들을 얻기 위해서 객체관리를 통하여 데이터베이스를 먼저 검색하도록 한 후, 원하는 데이터가 탐색되어지면 이를 추출하여 모델인스턴스를 만들거나 또는 사용자로부터 직접 데이터를 받아들여 모델인스턴스를 만든다. 물론 사용자는 모델인스턴스를 볼 수 있어야 하며, 모델의 데이터를 여러 가지 시각에서 볼 수 있게 하며, 필요시 편집할 수 있게 한다. 예를 들어 LP모델의 경우 매트릭스 형태로 볼 수도 있으며, 대수식 형태, 또는 네트워크 형태로 볼 수 있게 한다. 완성된 모델인스턴스의 해를 구하기 위해서는 해산출기가 요구하는 포맷으로 데이터를 변환한다.

또한 복합모델버전을 실행하기 위해서 각 원소모델에 나온 해를 자동적으로 연결해 준다. 즉 각 원소모델버전으로 구성된 복합모델의 입출력관계를 이용하여 한 원소모델에서 나온 출력 값을 다른 원소모델의 입력 값으로 연결시켜준다.

이러한 단순한 입력값에 대한 모델실행 뿐만 아니라 사용자는 하나 또는 여러 가지의 입력 값의 변화에 따른 출력 값의 변화효과를 분석하는 “what-if” 분석을 할 수 있어야 한다. 이러한 유형의 분석에는 사용자가 몇 개의 입력변수의 값을 하나씩 바꾸어가며 그 결과를 하나씩 관찰하는 단순한 형태의 민감도분석과 여러 가지의 시나리오를 열거하고, 각 시나리오 하에서 모델을 실행한 결과를 종합하여 보여주는 복잡한 형태를 모두 지원한다.

또한 사용자는 그 모델을 이용하여 특정의 결과치를 얻기 위해서 입력변수를 얼마로 조정해야 하는가를 찾는 목표탐색질의 처리를 할 수 있다. 이러한 목표탐색질의 처리는 “what-if” 질의처리보다 훨씬 어렵다. 왜냐하면 목표수준의 값으로부터 입력 값(의사결정변수 값)을 찾기 위해서는 모델을 역으로 풀어야 하기 때문이다. 이를 위해서는 Newton-Raphson 법과 같이 하나 또는 여러 개의 의사결정변수에 대한 초기 추정치에서 출발하여 특정의 허용치 내에 목표 출력 값이 도달할 때까지 모델을 반복적으로 푸는 연속적 근사원리를 이용해야 한다.

그런데 사용하는 모델이 최적화모델이 아닌 경우(예, 시뮬레이션모델)에는 최적해(optimal solution)를 구할 수 없으므로 최선의 해(best solution)를 찾기 위한 탐색매카니즘이 요구된다. 그런데 이것은 탐색공간의 크기와 탐색알고리즘의 설계에 따라 산출된 해의 효율성을 좌우된다.

4. 객체관리기

객체관리기는 다음과 같은 기능을 가진다.

① 모델, 지식, 데이터, 해산출 알고리즘을 각각 하나의 객체타입으로 보고 이들을 보조기억장치에 저장, 검색, 갱신하는 작업을 수행하는 지장관리기능을 가진다. 객체관리기는 모델관련 정보(모델스키마, 모델 입력데이터, 모델 출력결과, 해산출 알고리즘 코드 등)와 모델링 지식(규칙 또는 일차논리, 프레임)을 보조기억장치에 저장관리한다. 이것은 전통적인 파일형태나 또는 DBMS의 릴레이션과 같은 형태로 저장된다.

② 각 객체타입들에 대한 정보를 담은 객체사전(object dictionary)을 유지하는 기능을 가진다. 사용자가 다수일 경우 관리해야하는 모델스키마, 모델데이터, 해산출기 등은 매우 많을 수 있다. 이런 경우 특정한 문제에 직면한 사용자가 자신이 원하는 적절한 모델을 편리하게 찾을 수 있도록 해야 한다. 이를 위해서는 도서관에서 책을 분류하여

분류번호를 매겨 원하는 도서를 신속하게 찾는 것과 같이, 모델의 다중분류를 지원함으로써 사용자가 적절한 모델을 신속하게 찾을 수 있게 한다. 이를 위해서는 모델을 문제 중심으로 분류하거나 또는 조직의 기능, 계층, 복잡성 등에 따라 분류하는 기법이 필요하다. 각 분류기법들은 가장 일반적인 범주에서부터 보다 더 특정의 하위범주로 계층적 트리구조로 구성할 수 있다. 또한 특정 모델인스턴스와 모델데이터 해산출기에 대한 정보와 이들 간의 상호관련성에 대한 정보를 담은 사전이 있어야 한다. 따라서 객체사전은 모델, 데이터, 해산출기 등에 대한 일종의 메타지식을 저장하고 관리하는 기능을 갖는다.

③ 각 객체에 대한 동시적 접근을 조정하는 기능, 즉 객체들에 대한 사서(librarian) 역할을 한다. 즉, 하나의 객체(예 모델비전)가 만들어지면 이것을 등록하고, 저장된 객체에 대한 “check-out” 요구가 들어오면 이것이 수정될 수 있는 객체인지 또는 수정이 인가되지 않는 객체인지를 식별하여 “check-out”을 하고 적절한 “lock”을 걸어두어 중복 “check-out”이 일어나지 않도록 한다.

④ 완전한 객체명세(객체들 간의 일관성과 무결성조건 등)를 검사하는 무결성 검사(integrity check)기능을 갖는다. 무결성 검사의 예로는 복합모델을 구성하는 원소모델 간의 순환현상(cycling)을 체크한다든지, 객체들 간의 참조적 무결성(referential integrity)를 검사한다든지, 일반화와 구체화에 의해 계승이 존재하는 경우 상위 모델클래스의 애트리뷰트 수정이 하위클래스나 인스턴스에 반영되는가를 체크한다.

⑤ 객체관리기에 대한 모든 요구를 모니터하는 활동감독(activity supervisor)기능이 필요하다. 오늘날 모델링환경에서 가장 중요한 문제점 중의 하나는 모델유지보수와 타당성평가 기능이다. 하지만 이러한 문제는 모델변경의 파문효과와 모델의 모델사용 환경변화에 따른 모델타당성을 재평가해야 하는 문제로 인하여 모델관리를 더욱 어렵게 하고 있다. 예를 들어 하나의 수요예측모델이 과거의 데이터로 평가하여 타당성이 인정되었으나 시간이 지남에 따라 새로운 데이터가 들어왔을 경우 그 타당성을 보장하기란 어렵다. 모델타당성은 모델수립자의 주관적인 판단이 앞서므로 완전히 자동화하기란 어렵지만 타당성평가 프로그램을 개발하여 이러한 평가노력들을 어느 정도는 완화할 수 있다. 이러한 문제들을 극복하기 위해서는 객체(모델)들 간의 종속관계를 명시함으로써, 특정 객체의 변경을 모니터하여 관련 객체에 미치는 파급효과가 반영되도록 한다. 또한 타당성평가를 위해서는 타당성평가 지식(규칙)을 이용한다.

⑥ 질의에 응답하기 위해 데이터베이스를 검색하거나 또는 지식베이스를 탐색하여 적절한 추론을 할 수 있도록 질의처리기나 추론엔진을 갖추고 있어야 한다.

제 6 장 결 론

모델관리시스템의 필요성과 목적, 그리고 바람직한 모델링환경의 특성에 비추어 볼 때, 좋은 모델링 또는 모델관리시스템이 되기 위해서는 다음의 여러 가지의 조건들을 갖추어야 할 것으로 결론을 내릴 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.

② 모델표현의 일반성과 비절차성이 높고, 개념적 모듈성과 모델독립성이 강력히 지원될 수 있어야 하며, 표현된 모델의 자동적 문서화를 가능하게 해야한다.

③ 컴퓨터기반의 전 모델링 수명주기 지원능력이 갖추어져야 한다.

④ 모델관리의 주요 자원(모델, 데이터, 계산출기(solvers), 모델관리지식)의 관리기능이 있어야 한다.

⑤ 모델정의를 위한 단일의 패러다임 중립 언어가 있어야 하며, 이것은 자연언어에 가까우면서(記述的 성격이 강함) 컴퓨터 상에서 곧바로 실행가능한 언어이어야 한다.

⑥ 기존의 모델링언어 문법으로 표현된 것들을 수용할 수 있도록 그 문법이 충분히 일반적이어야 한다.

⑦ 모델활동은 많은 모델링 지식과 도메인 지식을 필요로 하므로 지식기반의 시스템이 되어야 한다.

⑧ MMS는 상이한 모델을 수용하고 단일 모델의 다중관점을 지원할 수 있도록 충분히 일반적이어야 한다.

⑨ MMS는 상이한 응용영역에 대한 모델의 사용을 지원해야 한다. 더욱이 모델에서 요구되는 다양한 원천(사용자, 데이터베이스, 또는 다른 모델의 출력)의 데이터를 결합할 수 있어야 한다.

이상에서 제시된 특성들을 가장 잘 지원해 줄 수 있는 모델관리시스템을 개발하기 위한 프레임워크로 본 연구에서는 객체지향적 접근법을 채택하고 있다. 모델관리에 객체지향적 접근법을 도입함으로써 다음과 같은 잇점을 얻을 수 있다.

① 모델베이스설계를 위한 다양한 모델링 구성자(constructs)를 제공해준다.

② 강력한 모델분류기능과 모델타입화를 지원할 수 있어 효과적인 모델베이스 조직화를 가능하게 한다.

③ 계승개념은 모델베이스 내의 모델의 재사용을 촉진시키고 새로운 모델을 쉽게 만들 수 있다.

④ 시스템 내의 각 객체타입들이 잘 정의된 지식중심의 구조로 구성되므로 사용자나 시스템에 의해서 보다 더 쉽게 관리되고 조작될 수 있다.

⑤ 모델베이스 내에 있는 객체들을 보다 더 큰 시스템으로의 통합이 용이하다.

⑥ 일반모델타입과 포트타입은 모델의 단순화된 관점을 제공해주므로써 매우 복잡해 보이는 모델의 세부적인 사항을 사용자가 알아야 하는 부담을 줄여준다.

이러한 장점을 지닌 객체지향적 모델관리시스템을 개발하는데 있어 가장 중요한 열쇠는 역시 모델관리시스템을 구성하는 요소들(의사결정모델, 해산출기, 모델데이터, 모델관리지식)을 하나의 객체타입으로 보고, 이 객체들을 효과적으로 조직화하고 설계하는 데 있다.

본 연구에서는 모델에 대한 이러한 객체지향적 관점을 지원하고, 효과적인 모델관리를 수행할 수 있는 객체지향적 모델관리시스템(OOMMS)의 아키텍처를 제시하고 있으며, 각 하위시스템의 기능명세를 제공하고 있다. OOMMS는 모델베이스, 해산출기, 지식베이스, 데이터베이스를 구성하는 여러 가지의 요소들을 하나의 객체로 보아 그 관리기능을 일원화할 수 있는 객체관리기 개념을 도입하고 있다.

훌륭한 모델관리시스템은 그래픽(아이콘), 네트워크, 그리고 대수적 표현기법이 포함된 한 가지 이상의 외부적 모델표현을 수용할 수 있어야 할 것이다. 또한 MMS는 특정 모델링패러다임만을 지원하기보다는 일반성이 높은 시스템을 지향하며, 사용자가 자신이 무엇을 할 것인가만을 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.

이러한 문제를 해결하기 위해서는 메타프레임[33], 메타추상화[17]과 같은 메타지식(meta-knowledge)의 도입과 메타그래프[9], 메타모델링[44] 개념의 도입이 필요하다. 이러한 메타 모델링 개념의 도입은 모델링의 효율성을 향상시키고 보다 더 지적능력을 갖춘 모델관리시스템을 가능하게 할 것이다.

참 고 문 헌

1. 김유일, 김진수, 정내윤, "모델관리시스템의 세 접근법에 대한 비교연구," 정보시스템연구, 제3권, 1994년, pp.101-132.
2. 허순영, "최적화 모델링 인어를 위한 객체지향 모형관리체계의 개발," 經營科學, 제11권, 제2호, 1994년 6월, pp.43-63.
3. Applegate, 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.

4. 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.
5. 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.
6. 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.
7. Blanning, R.W., "A Relational Framework for Join Implementation in Model Management," *Decision Support Systems*, Vol.1, No.1, 1985, pp.69-82.
8. Blanning, R.W., "An Entity-relationship Approach to Model Management," *Decision Support Systems*, Vol.2, No.1, 1986a, pp.65-72.
9. Blanning, R.W., "Metagraphs in Model Management," *Paper Presented at the TIMS/ORSA Joint National Meeting*, Chicago, May 1993.
10. Blanning, R.W., "Model Management Systems : An Overview," *Decision Support Systems*, Vol.9, No.1, 1993, pp.9-18.
11. 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, N0.2, 1981, pp.263-281.
12. Choobineh, J., "SQLMP: A Data Sublanguage for Representation and Formulation of Linear Mathematical Models," *ORSA Journal on Computing*, Vol.3, 1991a, pp.358-375.
13. Chung, Q.B., and R.M. 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.
14. 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.
15. Dolk, D.R., "A Generalized Model Management System for Mathematical Programming," *ACM Transactions on Mathematical Software*, Vol.12, No.2, 1986a, pp.96-126.
16. Dolk, D.R., "Data as Models : An Approach to Implementing Model Management," *Decision Support Systems*, Vol.2, No.1, 1986b, pp.73-80.
17. 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.
18. Dolk, D.R., and B.R. Konsynski, "Model Management in Organizations," *Information*

- and Management*, Vol.9, 1985, pp.35-47.
19. 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.
 20. 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.
 21. 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.
 22. Fourer, R., D.M. Gay, B.W. Kernighan, "A Modeling Language for Mathematical Programming," *Management Science*, Vol.36, No.5, 1990, pp.519-554.
 23. Geoffrion, A.M., "An Introduction to Structured Modeling," *Management Science*, Vol.33, No.5, 1987, pp.547-588.
 24. Geoffrion, A.M., "Computer-based Modeling Environments," *European Journal of Operational Research*, Vol.41, 1989, pp.33-43.
 25. Geoffrion, A.M., "FW/SM: A Prototype Structured Modeling Environment," *Management Science*, Vol.37, No.12, 1991, pp.1513-1538.
 26. Greenberg, H.J., "A Way of Thinking about Model Analysis," *Interface*, Vol.10, No.2, 1980, pp.91-96.
 27. Huh, S.Y., "Modelbase Construction with Object-oriented Constructs," *Decision Science*, Vol.24, No.2, 1993, pp.409-434.
 28. Jones, C., "An Introduction to Graph-based Modeling Systems, Part I: Overview," *ORSA Journal on Computing*, Vol.2, 1990, pp.136-151.
 29. 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.
 30. Kimbrough, S.O., "A Graph Representation for Management of Logic Models," *Decision Support Systems*, Vol.2, No.1, 1986, pp.27-37.
 31. Krishnan, R., "Automated Model Construction: A Logic Based Approach," *Annals of Operations Research*, Vol.21, 1989, pp.195-226.
 32. Krishnan, R., "A Logic Modeling Language for Automated Model Construction," *Decision Support Systems*, Vol.6, No.1, 1990, pp.123-152.
 33. Krishnan, R., "PDM: A Knowledge-based Tool for Model Construction," *Decision Support Systems*, Vol.7, No.2, 1991, pp.301-314.
 34. 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.
35. Lenard, M.L., "An Object-oriented Approach to Model Management," *Decision Support Systems*, Vol.9, No.1, 1993, pp.67-73.
 36. Liang, T.P., "Integrating Model Management with Data Management in Decision Support Systems," *Decision Support Systems*, Vol.1, No.2, 1985, pp.221-232.
 37. Liang, T.P., "A Graph-Based Approach to Model Management," *Proceedings of Seventh International Conference of Information Systems*, 1986b, pp.136-151.
 38. Liang, T.P., "Development of a Knowledge-based Model Management System," *Operations Research*, Vol.36. No.6, 1988a, pp.849-863.
 39. Liang, T.P., "Analogical Reasoning and Case-based Learning in Model Management Systems," *Decision Support Systems*, Vol.10, 1993b, pp.137-160.
 40. Liang, T.P., and C. Jones, "Meta-Design Considerations in Developing Model Management Systems," *Decision Sciences*, Vol.19, No.1, 1988, pp.72-92.
 41. 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.
 42. Ma, P.C., F.H.. Murphy, and E.A. Stohr, "Representing Knowledge about Linear Programming Formulation," *Annals of Operations Research*, Vol.21, 1989a, pp.149-172.
 43. 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.
 44. 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.
 45. Muhanna, W.A., "An Object-oriented Framework for Model Management and DSS Development," *Decision Support Systems*, Vol.9, No.2., 1993, pp.217-229.
 46. Muhanna, W.A., "SYMMS: A Model Management System That Supports Model Reuse, Sharing, and Integration," *European Journal of Operational Research*, Vol.72, 1994a, pp.214-243.
 47. Muhanna, W.A., and R.A. Pick, "Meta-modeling Concepts and Tools for Model Management: A Systems Approach," *Management Science*, Vol.40, No.9, 1994, pp.1093-1123..

48. 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.
49. 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.
50. 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.
51. Schrage, L., *Linear, Integer and Quadratic Programming with LINDO*, The Scientific Press, Redwood City, CA, 1987.
52. Sivasanakaran, T., and M. Jarke, "Logic-based Formula Management Strategies in Actuarial Consulting System," *Decision Support Systems*, Vol.1, 1985, pp.251-262.
53. Will, H.J., "Model Management Systems," in: (eds.) E. Grochla, and N. Szyperski, *Information Systems and Organization Structure*, Walter De Gruyter, Berlin, 1975, pp.468-482.