

통합생산을 위한 생산정보 모델에 관한 연구†

김철한* · 김광수**

An Information Modeling Methodolgy for CIM

Cheolhan Kim and Kwangsoo Kim

Abstract

An economically competitive automated manufacturing system integrates the various control processes and data used in design, manufacturing, sale and service of products. CIM is a way to achieve such integration through computers and computational techniques in design, planning, and manufacturing. Developing effective CIM architectures is hampered by integration problems. The key to resolving these problems lies in a better understanding of manufacturing function and how it is related to other manufacturing functions. Integration of CIM environment requires coordinated solutions to data management problems for individual application system as well as for exchange of data between these applications. This requires a common framework for data management throughout the CIM environment. This paper discusses the design paradigm as a framework for this purpose. Designing an organizational structure to meet those goals involves 1) analyzing the functions through functional decomposition, 2) developing a data model to coordinate functions. As a result, we propose an object-oriented design methodology for manufacturing information system.

1. 서 론

컴퓨터를 이용한 생산방식인 CIM(Computer Integrated Manufacturing)에 대한 관심이 생산라인의

자동화에 박차를 가하고 있는 국내 제조업체들 사이에 폭넓게 확산되고 있다. CIM은 컴퓨터 hardware 및 software, network기술, 생산기술 등을 database를 근간으로 하는 information technology와

† 본 연구는 포항공과대학의 연구비 지원에 의해 수행되었음.

* 포항공과대학 산업공학과

유기적으로 결합하여 기업전체의 경쟁력을 높이려는 기업전략으로 적시에 정확한 결정을 내릴 수 있도록 요구되는 장소, 사람, 또는 장비에 정확한 정보를 제공하는 것을 목적으로 하고 있다.[1] 그러나 일반적으로 CIM을 추구하는 국내외의 많은 기업들이 느끼는 점은 단위기계 및 단위시스템의 자동화에 따르는 효과가 기대에 미치지 못한다는 점이다. 이는 단위기계나 단위시스템의 설계시에 통합된 생산시스템을 위한 고려가 선행되지 못하고, 각 기능에 따라서 필요에 따라서 개별시스템을 개발하였으므로 전체를 통합하여 운용할 때 여러 가지 문제점이 발생하기 때문이다. 따라서, 개별시스템은 통합시스템을 고려한 common database의 기초위에서 설계하여야 한다. 이를 위해서는 CIM 시스템을 구성하는 많은 하부시스템에 대한 분석이 행해져야 하는데 대표적인 시스템 분석방법으로는 ICAM의 IDEF[2, 3, 4, 5], GRAI의 GRAI[6], Nijssen의 NIAM[7], Canada NRC의 M*[8] 등이 있다. 그러나 이들 분석방법은 대부분 사무용데이터 처리를 위한 분석방법(DFD, E-R Model) 등에 근거를 두고 있기 때문에 생산시스템을 분석하는데는 제각기의 제한점들이 있다[9].

IDEF 경우에는 시스템을 functional, information, dynamics model 등 세단계로 구분하여 시스템을 분석하고 있어 각 model에 의한 시스템의 설명이 용이하나, 이를 바탕으로 새로운 시스템의 설계시에 조직도나 interface를 고려하지 않았기 때문에 전체를 하나의 view로 파악하여 database를 설계하기에는 한계가 있고, 데이터를 단순한 entity로 정의하기 때문에 데이터가 가지고 있는 모든 특징을 표현하여 실제의 생산정보 시스템을 구축하기에는 한계가 있다. GRAI 경우에는 생산시스템 전체를 파악하기 보다는 생산시스템내에서 사용되고 있는 decision making에 중점을 두고 분석할 수 있는 tool이다. N*는 database 설계를 목적으로하여 생산시스템을 분석하기 위한 방법으로 각 단계에서 사용될 수 있는 tool들을 제시하고 이를 바탕으로

생산시스템을 분석하고 있으나 복잡한 생산시스템을 하나의 database로 설계하기 위하여 DATAID라는 사무용 database설계 방법에 그 기본을 두고 있고 있기 때문에 실시간을 표현하여 수행하는 생산시스템의 database설계에는 한계가 있다. 또한 Baudin[11]은 생산시스템을 DFD(data flow diagram)으로 분석하였으나, Hsu와 Rottner[10]가 언급하였듯이 DFD는 process-oriented의 계층구조이기 때문에 view의 생성이나 의미의 한정(semantic constraints) 등의 면에서 한계가 있다.

따라서 본 연구에서는 이들 대표적인 시스템 분석방법들이 갖고 있는 한계들을 보완하는 새로운 생산시스템 분석방법을 제시하고자 한다. 이를 위하여 본 연구에서는 제 2장에서 기존의 시스템 분석방법 중에서 대표적인 방법들을 소개하고, 제 3장에서는 제 2장에서 소개한 방법들이 갖고 있는 한계를 보완할 수 있는 새로운 생산시스템 분석방법을 제안하였다. 제 4장에서는 이 방법에 따라서 생산시스템을 분석한 예를 제시하였고, 제 5장에서는 연구결과를 요약하고 앞으로의 연구 방향을 제시하였다.

2. 시스템 분석방법

여기서는 대표적인 시스템 분석기법인 ICAM program의 IDEF와 GRAI 연구소의 GRAI, 그리고 Canada NRC의 M*등을 분석하여 이들 기법들이 가지고 있는 특징을 분석하고자 한다.

2-1. IDEF

IDEF(ICAM definition)는 미국 공군에서 생산 시스템을 구조적으로 분석하기 위하여 개발한 분석방법이다. 이 방법은 대상시스템의 activity들과 그들간의 상호관계를 규정하는 functional model인 IDEF₀, 대상 시스템에 의해서 요구되는 정보의 구조를 규정하는 information model인 IDEF₁, 시

간에 따른 function과 information의 behavior를 규정하는 dynamics model인 IDEF₂로 나뉘어진다.

IDEF₀는 일종의 decomposition 방법으로 대상시스템 전체를 하나의 box로 표시하며, 이에 대한 입력, 출력, 제어 및 기구를 설정하여 시스템을 설명하고 다시 이 box의 내부를 기능에 의해서 3-6개의 하위 box로 나누어 분석하는 계층구조를 지닌 방법으로 생산시스템내의 data flow를 정의하기 위하여 사용된다.

IDEF₁은 수행되는 function을 지지하는 information의 구조를 E-R Model을 이용하여 표현하는 것이다. 이 방법의 장점은 생산시스템을 설명하는 정보모델이 생산정보의 요구변화에 따른 database나 그 적용에 관계없는 데이터 구조를 제공한다는 것이다.

IDEF₂은 시간의 변화에 따라서 반응하는 function이나 information의 behavior를 분석하고 설명하기 위하여 생산시스템을 시간의 특성에 따른 state-transition으로 나타낸다.

2-2. GRAI

GRAI 연구소에서 생산시스템을 구축하기 위한 개념으로 제안한 것으로 physical system과 production control system으로 구성되어 있다. Physical system이란 manufacturing process plan과 engineering specification에 따라서 raw materials, parts, components 등을 이용하여 가공하거나 조립하는 등의 공정을 통하여 제품을 생산하는 기능을 가진 생산 단위로서 사람, 기계, 공구 등과 같은 생산 환경을 의미한다. Control system이란 information system과 decision system으로 구성되어 order, resources, energy 등과 같은 information으로 부터 필요한 decision을 내려서 physical system으로 하여금 그 기능을 수행하게 하는 것으로 decision making이 주 임무이다. GRAI conceptual model은 information system과 decision system 그리고 physi-

cal system로 구성되어 있다. Decision과 information은 각 단계로 세분되며, 따라서 information은 각 단계의 decision에 맞도록 구축되어야 한다. 즉, planning horizon은 짧을수록, physical system에 가까이 갈수록 information들이 세분되어야 한다. 한편 이들 GRAI method를 표현하기 위해서는 GRAI Nets와 GRAI grid라는 두가지 graphic tool이 사용되는데, GRAI grid는 production management system의 decision center의 전체구조를 계층적으로 표현하기 위한 tool이며, GRAI Nets는 각 decision center의 다양한 activity를 설명하는 tool이다.

2-3. M*

위에서 언급한 IDEF나 GRAI가 database 구축면을 고려하지 않은 반면에 M*는 database 구축을 고려한 생산정보를 분석방법이다. Italy의 DATAID [4] project의 아이디어를 이용하여 엔지니어에게 CIM 환경하에서의 database 개발을 위한 design tool들을 제공하기 위한 것으로 enterprise modeling and analysis, conceptual design, implementation design 등과 같은 3단계로 구성되어 있다.

Enterprise Modeling 및 Analysis 단계에서는 functional, information, dynamic model 등을 이용하여 AS_BE 분석과 TO_BE 분석을 통하여 기업을 구조적으로 분석한다. AS_BE 분석은 기업의 current-state를 알기위한 방법으로 interview/ modeling/ validation/ integration 분석의 사이클을 통하여 top_down 방식으로 기업을 분석하고 이를 bottom_up 방식을 통하여 검증하고 평가하는 것이다. TO_BE 분석은 현 시스템의 문제점을 AS_IS 분석의 결과인 function diagnostics를 통하여 각 문제점을 structural, operational, information-based으로 구분하고 이를 해결할 수 있는 새로운 시스템을 제안하는 것이다.

Conceptual design은 대상기업의 local view와

global view를 결정하는 것으로 data schema, operation schema, functional net 등을 이용하여 data model, operation model, process model을 정의하는 것이다. Data model로는 extended entity-relationship (EER) model을 대상기업의 information을 표현하는데 사용하였는데, entity, relationship, attribute, abstraction hierarchy라는 4개의 추상화 메카니즘을 제공한다. Operation model은 operation을 EER model 내에서 표현된 data schema로서 설명하며, 이를 operation schema라 하는데, operation에 필요한 data의 이름 및 종류를 정의한다. Process model은 object function의 수행조건에 따른 일련의 operation 들로서 process net을 정의하는데 이때, Petri-net[9]이 그 표현 및 수행의 근간이 된다. 이들의 수행조건은 object function의 외부에서는 오는 message에 따라서 결정된다. 이들 model로서 만들어진 local view들은 conflict 분석을 통하여 서로의 모순점들을 해결하여 통합한 global view를 만든다.

Implementation design 단계에서는 conceptual design 단계에서 생성한 model을 사용하고자 하는 특정 DBMS의 data structure로 변환시키는 것으로 global view의 결과로 얻어진 global data schema와 global operation schema를 이용하여 실제 사용할 RDBMS의 relational schema와 application program를 구축하고 이를 바탕으로 physical schema를 설계하는 것이다.

3. 새로운 생산시스템 분석방법

앞에서 언급한 바와 같이 기존의 시스템 분석방법으로는 CIM시스템을 효과적으로 분석하는 데는 한계가 있으므로 본 연구에서는 object-oriented paradigm에 의한 새로운 생산정보시스템 설계방법을 제안한다. 효과적인 생산정보시스템 설계를 위해서는 먼저 functional level에서 top-down 분석을

통해서 생산시스템이 어떤 기능들로 구성되어 있는가를 분석한 다음에, operational level에서 bottom-up 분석을 통하여 구성된 기능 내에서 또는 기능들 간의 data가 어떻게 표현되고 그 의미가 변화하는 가를 파악해야 한다.

이를 위하여 본 연구에서는 functional decomposition 방법을 이용하여 시스템내의 기능들을 평형 구조 또는 계층구조로 분석하여 각 기능내에서 사용되고 있는 데이터를 주변 기능과의 관계를 고려하여 IDEF₀에서 정의한 것과 같이 input, output, control, mechanism data로 구분하였다. 이렇게

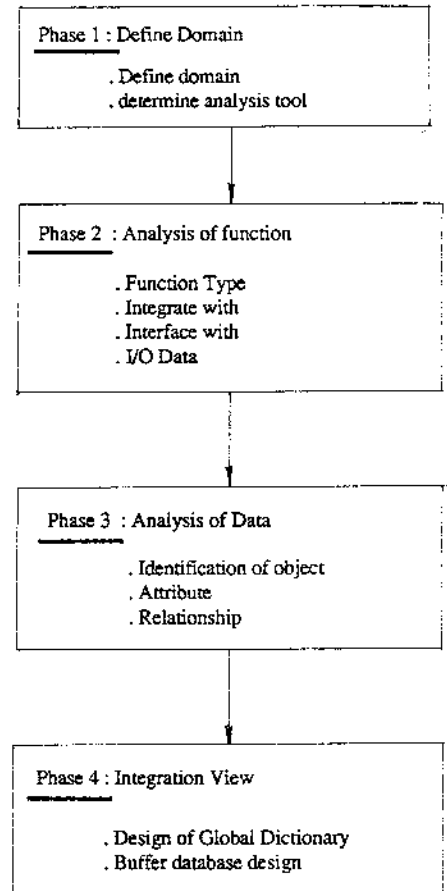


그림 1. Information modeling methodology for CIM

정의된 데이터는 각 기능에 따라서 그 의미가 달라질 수 있으므로 생산시스템 내의 데이터를 각 기능에서의 의미를 포함하면서 일관성있게 표현하기 위하여 본 연구에서는 이러한 데이터를 object-oriented data model을 이용하여 표현하였다. object-oriented data model은 생산시스템의 데이터가 갖는 복잡성과 이질성들을 표현할 수 있으며, object-oriented 방법이 갖고 있는 특징들 즉, abstraction, polymorphism, inheritance, encapsulation 등을 이용하여 각 기능에 따라 달라지는 데이터의 의미를 쉽게 구현할 수 있다.

이러한 functional decomposition 방법과 object-oriented 방법을 이용하여 생산시스템을 분석하고 표현하는 절차는 그림 1과 같다. 즉, 1) 생산시스템의 분석범위(domain)를 정의하고 2) 기능을 세분화하여 이를 분석하고 3) 기능에 사용된 데이터를 분석하여 이를 바탕으로 생산시스템의 데이터샤양을 정의하고, 4) 전체를 하나의 관점에서 통합하는 것이다.

1) 분석범위의 정의

생산시스템은 원자재로부터 최종상품이 만들어지는 과정인 제품설계, 공정설계, 생산계획, 생산관리, 생산공정을 포함하는 시스템으로 본 연구에서는 CIM 구축시에 요구되는 생산시스템의 정보시스템 즉, 생산정보 시스템(manufacturing information system)을 분석범위로 정의하였다.

2) 기능의 분석

기능의 분석이란 시스템이 갖고 있는 기능을 어떠한 관점에서 어떠한 기준으로 분석하는가를 뜻하는 것으로 본 연구에서는 자료수집을 단순화할 수 있고, 하부시스템을 분석하기 쉽고 표준화하기 용이한 functional decomposition 방법[4]을 이용하여 다음과 같은 기준으로 하부시스템을 분류하고자 한다.

- 정보흐름(information flow)의 최소화
- 동일공정 또는 연속공정의 집합으로서 일관성을 유지
- 데이터의 투명성(transparency)을 유지
- 자재나 장비에 영향을 미치는 지 여부
- 의사 결정기구가 존재하는 지 여부

위의 기준중에서 처음의 2가지는 기능의 단위를 정의하는 것으로 동일공정이나 연속공정을 하나의 기능으로 간주함으로써 정보의 흐름이 최소화할 수 있도록 기능을 black box화하여 모든설계에 의한 생산시스템 구축의 생산성과 신뢰성을 높일 수 있다. 생산시스템내의 정보망(information network)은 형태상 분산데이터베이스를 기준으로 하기 때문에 [12, 13] 데이터의 투명성을 유지할 수 있도록 기능들이 구분되어야 한다. 또한 나머지 두가지는 이렇게 나뉘어진 기능을 그 활동 영역에 따라 분류하는 기준이다.

위의 개념으로 기능을 분류할 경우, 기능은 공통기능(common function), 실행 기능(execution function), 준비기능(planning function)으로 구분할 수 있다. 공통기능이란 의사 결정기구(decision making mechanism)을 갖지 않는 기능들로 다른 기능들을 지원하는 master file의 생성 및 유지보수를 담당하는 part master 또는 part number logging 등과 같은 기능들이며, 실행기능이란 생산시스템의 작동을 수행하는 기능들로서 외부의 사건(event)에 의해서 작동되는 shop floor monitoring 등과 같은 event-driven function 들로서 이들 기능들은 그 기능이 갖고 있는 의사결정기구를 정의하여야 하며 이들 의사 결정구조가 다른 기능들과 갖는 관계를 제층적 구조로서 뿐만 아니라 평행구조로서도 파악해야 한다. 한편, 준비기능이란 실행기능이 수행될 수 있도록 준비하는 기능으로 자재나 장비 등에 물리적인 영향이 미치지 않는 forecasting 등과 같은 기능들로 실행기능과 병행하여 또는 실행기능과 관계없이 발생할 수 있다. 이처럼

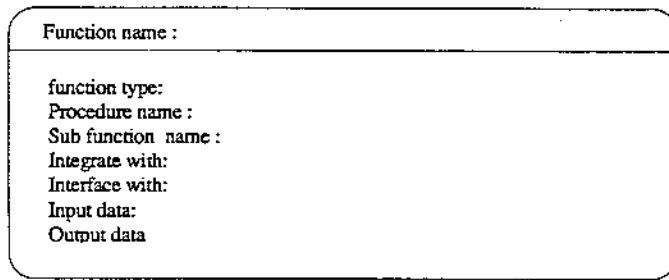
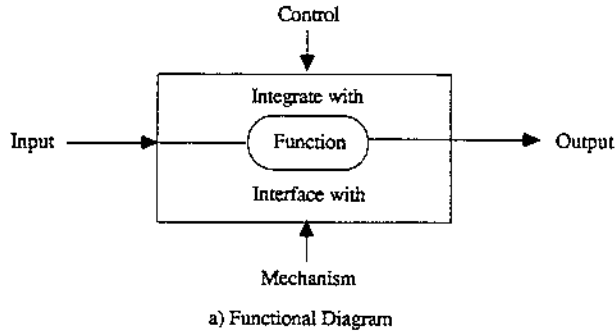


그림 2. Functional diagram and Functional chart

정의된 기능들은 각각의 procedure와 기능들의 하부기능, 기능의 입출력을 담당하는 데이터가 정의되어야 하며 functional decomposition의 결과로 그림 2와 같은 기능도(function diagram)과 기능표(function chart)를 정의할 수 있다. 본 연구에서는 기능사이의 정보 흐름을 알기 위하여 제 2장에서 언급한 IDEF₀와 같은 형태의 기호를 사용하였으나, 그림 2에서 처럼 box 안의 기능을 기능들 간의 평형구조를 고려하여 관계있는 주변 기능들을 'Interface with'와 'Integrate with'로 나누어 기술하였다. 'Interface with'의 관계는 단순히 데이터를 참조하는 관계에 있는 기능들이고, 'Integrate with'는 관련된 기능들에 데이터가 서로에게 영향을 미치는 기능들이다. 후자의 경우, 이들 사이에 있는 데이터의 생성 및 수정들은 연관된 모든 기능들을 고려해야 한다. 즉, 시스템분석 후에 이를 바탕으로

하여 분산정보시스템을 구축할 때 필요한 global dictionary 내에서 이들 기능들 간의 관계를 정의하기 위하여 'Integrate with'와 'Interface with'의 구분이 필요하다. 이들 기능들을 데이터의 입출력에 맞추어 연결하면 정보흐름에 따른 기능들의 역할 및 관계를 알 수 있게 된다. 이에 따른 예를 제 4장에서 설명하였다.

3) 데이터의 분석

위에서 정의한 기능에 따르는 생산시스템에 데이터는 일반적으로 복잡한 구조를 갖고 있으며 시간, 공간의 의미를 갖기도 하고, 자체내에서 계층 구조를 갖기도 하는 등 일반적인 형태로는 그 표현에 한계가 있다[14]. 따라서, 생산시스템에서 사용되고 있는 데이터를 표현한다는 것은 간단한 것이 아니다. 우선 생산시스템에 사용되고 있는

데이터는 성격상 공정데이터(process data), 제품 데이터(product data), 설비데이터(facility data)로 구분할 수 있는데, 여기서 공정데이터는 주로 실행 기능에 사용되는 데이터로서 의사 결정구조에 따라서 데이터가 변화하며, 의미가 달라질 수 있는 데이터이고, 제품데이터는 주로 준비기능 및 공통 기능에 사용되는 데이터로서 생산시스템의 최종출력인 제품에 관한 데이터이다. 설비데이터는 제품 데이터처럼 준비기능 및 공통기능에 사용되는 데이터로서 생산시스템을 구성하는 장비나, 사람, 공구 등에 관계된 데이터이다.

위의 정의에서 구분된 데이터는 그 특성에 따라서 표현방식이 달라지게 된다. 즉, 데이터는 속성에 따라서 동적특성을 갖고 시간에 따라서 변화하는 경우가 있고, 정적특성을 갖는 경우도 있고, 상황에 따라서 정보나 지식을 표현할 경우도 있다. 그러므로 데이터를 entity로 표현하고 이를 처리하는 ROBMS로는 생산시스템 데이터가 갖고 있는 다음과 같은 특성을 포함하기에는 한계가 있다[12].

- 이질(nonhomogeneous)의 객체들의 집합체로서의 데이터를 표현
 - vectors, matrices 또는 시간이나 공간의 데이터를 표현
 - 설계데이터의 변화를 지원하는 versioning 기능
 - 복합구조(complex object structure)를 표현
- 따라서 생산시스템에 쓰이는 데이터는 위에서 언급한 특징을 만족하기 위하여 여러가지 의미(semantic) 및 형태를 포괄적으로 표현할 수 있는 object-oriented data modeling 또는 semantic data modeling 방법 등이 필요하다[15, 16, 17, 18]. 이 방법들은 object(객체)를 정의하고 이 객체의 의미를 어떻게 표현하는가 하는 방법들로서 생산시스템에서 요구되는 데이터의 표현에 적합하고[10, 14, 19, 20, 21], semantic data modeling은 풍부한 의미(semantic) 표현이 가능하지만 동적특성을 표현하는데는 한계가 있다[22, 23]. 따라서 본 연구에서는

frame을 이용한 object-oriented data modeling method를 사용하여 생산시스템의 데이터를 표현하였는데, 이 방법은 다음과 같은 몇가지 특징을 가지고 있다[24, 25].

1. 생산시스템의 객체를 데이터베이스 객체로 1:1 mapping이 가능하다.
2. 추상화(abstraction)의 개념을 이용하여 객체간의 관계를 설명할 수 있다.
3. 객체내에 외부 객체와의 interface의 기능을 담당하는 method나 operation 등을 포함할 수 있다(encapsulation).
4. 클래스 사이에서 속성들을 상속할 수 있다(inheritance)
5. 복잡한 객체(complex object)의 표현이 가능
6. 모듈화가 가능하고, 유연성을 높으며, 언제나 확장이 가능하다.
7. method를 이용하여 시스템내의 동적특성을 데이터 내에 포함시킬 수 있다.

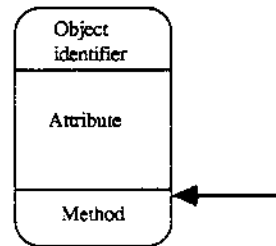


그림 3. Object representation

그림 3은 이러한 성격을 포함하고 있는 객체를 표현한 그림으로 object identifier는 객체를 나타내는 명칭이며, attribute는 그 객체가 갖는 성질을 의미하며 method는 객체와 객체사이의 message 전달에 의해서 행해지는 procedure나 algorithm 등을 의미하고 굵은 화살표는 객체사이의 message connection이 있음을 의미한다. 이들의 사용 예는 제 4 장에서 설명하였다.

이들 특징 중에서 객체, 객체 사이의 관계, insta-

nce 그리고 method 등을 체계적으로 정의할 수 있는 추상화의 개념은 generalization, aggregation, interaction 등의 세가지의 형태로 구분할 수 있다.

(i) Generalization

Generalization은 개개의 객체(subtype)를 일반적인 하나의 객체(type)로 표현하는 것으로, 이때 type은 subtype 사이의 공통성질(commonality)를 갖게 되는데, 각 subtype은 고유의 성질(intrinsic property)만 갖고 나머지 공통성질은 type으로부터 상속받는다. 그림 4에서 fastener와 screw는 서로간에 specialize 또는 generalize라는 의미를 갖게 되는데, 이들 관계를 그림 5의 (a)에서와 같은 generalization(G)의 관계로 정의한다.

(ii) Aggregation

일반적으로 assembly-subassembly 등과 부품구조에 유용한 것으로 여러개의 하위 클래스들이 하나의 상위 클래스를 구성하게 될 때 사용되며, 모든 클래스들은 각자의 instance를 갖는다는 점에서 하위 클래스에서만 instance를 갖는 generalization과는 구분이 되며 동일 객체의 set으로 묶을 수 없다는 점에서 interaction과도 구분이 된다. 그림 4에서 bolt assembly와 screw는 서로간에 part-of 또는 consist-of라는 의미를 갖게 되는데, 이들 의미는 그림

5의 (b)에서와 같은 aggregation(A)의 관계로 정의된다.

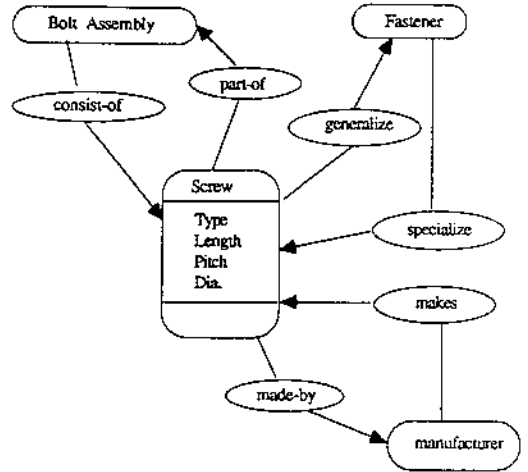


그림 4. Example of object relationship

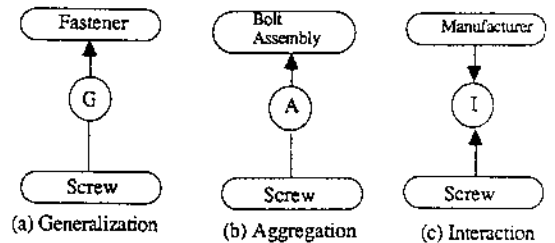


그림 5. Relationship representation of object

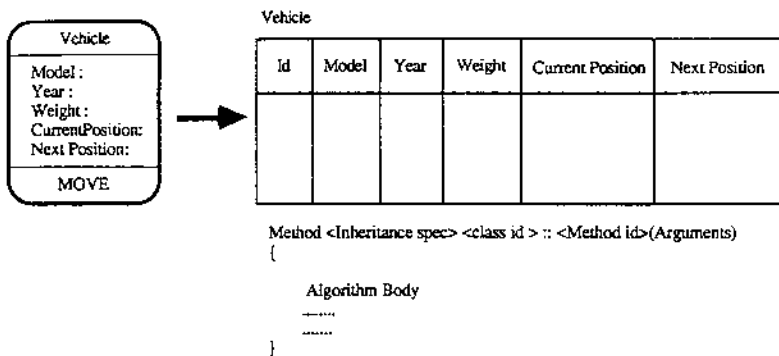


그림 6. Relational table of object

(iii) Interaction

interaction의 경우는 계층적으로 관련되어 있지 않은 클래스 사이의 관련성(relationship)을 나타내는데, 동일 set으로 나타낼 수 있는 클래스들의 관계를 나타낸다. 그림 4에서 screw와 manufacturer 사이에는 made-by와 makes라는 의미가 존재하는데, 이들의 의미는 그림 5의 (c)에서와 같은 interaction(I)의 관계로 정의된다.

이러한 특징들을 가진 object는 관계형 데이터베이스로 구축이 가능하다[24]. 즉, 기존의 object-oriented programming language(OOPL)와 RD-BMS를 사용하여 object-oriented 방법에 의한 생산정보시스템을 구현할 수 있다. 그림 6에서와 같이 객체의 속성을 DBMS의 table로 mapping하여 표현할 수 있으며[25], 객체내의 method 또는 객체가 갖고 있는 지식의 표현 및 추론기구를 OOPL을 사용하여 구현할 수 있다[26].

(4) Integration view

위에서 언급한 기능들과 이들 기능에 관련된 데

이타가 통합시스템 내에서 운용되려면 데이터의 투명성(transparancy)을 유지하면서 각 기능들의 view가 통합되어야 한다[27]. 미국의 NBS에서 제안한 생산시스템의 계층구조[28]를 이용하여 위에서 분류한 기능들이 각각 어느 계층들에서 사용되는지를 정의하는 그림 7과 같은 functional cross table을 작성하여 이를 바탕으로 global dictionary가 구축되어야 하며, 이들 기능들이 사용되는 데이터베이스간의 데이터 처리를 위한 버퍼 데이터베이스가 정의되어야 한다. Global dictionary는 local view들이 갖는 local dictionary의 통합으로서 분산 데이터베이스 구축시에 각 local database의 영역을 할당하며, 각 기능에서의 data의 생성, 유지, 보수 및 검색의 범위를 명확하게 구분하여 각 data base의 접근권한이 정의되어야 한다. 그림 8과 같은 버퍼 데이터베이스는 각 데이터베이스에 발생하는 데이터 중에서 다른 데이터베이스에서 필요로 하는 데이터를 저장하여 필요로 하는 데이터베이스에 필요한 형태의 데이터로 보내주고 각 데이터베이스에서의 통신을 담당한다.

	F1	F2	F3	F4	F5	F6	F7	F8
L1	*	*	*	*	*	*	*	*
L2	*	*		*	*	*	*	
L3	*	*		*		*	*	
L4	*					*	*	
L5						*		

LEVEL

- L1 : Plant Level
- L2 : Area Level
- L3 : Cell Level
- L4 : Machine Level
- L5 : Sensor Level

Function

- F1 : part master
- F2 : BOM
- F3 : forecasting
- F4 : master schedule
- F5 : cost planning
- F6 : facility management
- F7 : process planning
- F8 : accounting

그림 7. Functional cross table

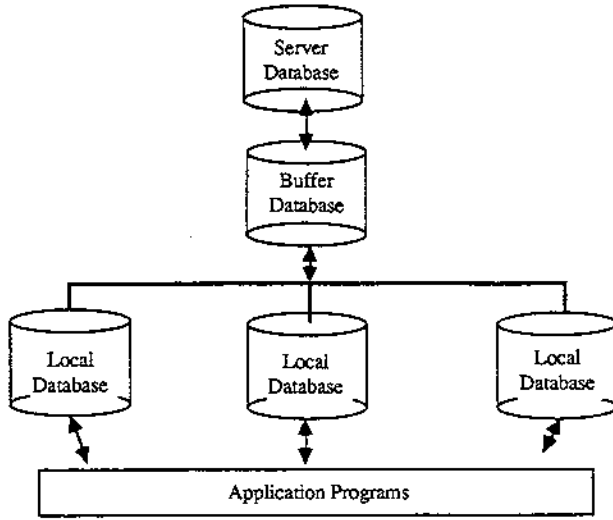


그림 8. Buffer database of distributed database

4. 응용 예

본 장에서는 material management에 관계 되어

있는 기능의 정보흐름을 제 3장에서 정의한 기호를 사용하여 나타내었다. 또한 이들 데이터 중에서 product structure에 관계된 데이터의 구조를 제 3

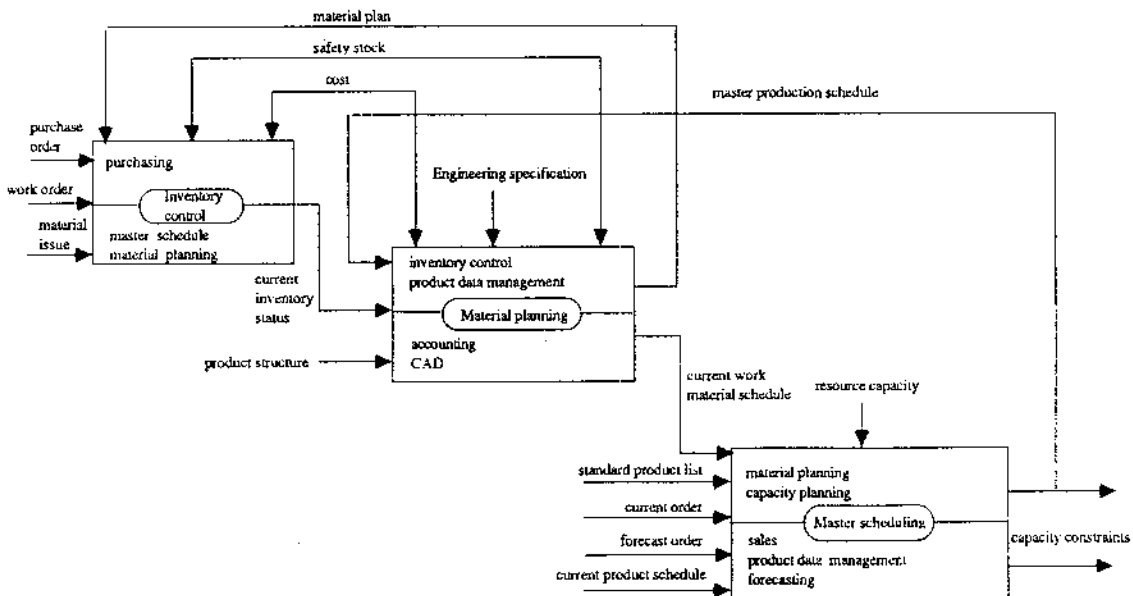


그림 9. Information Flow of Material Management

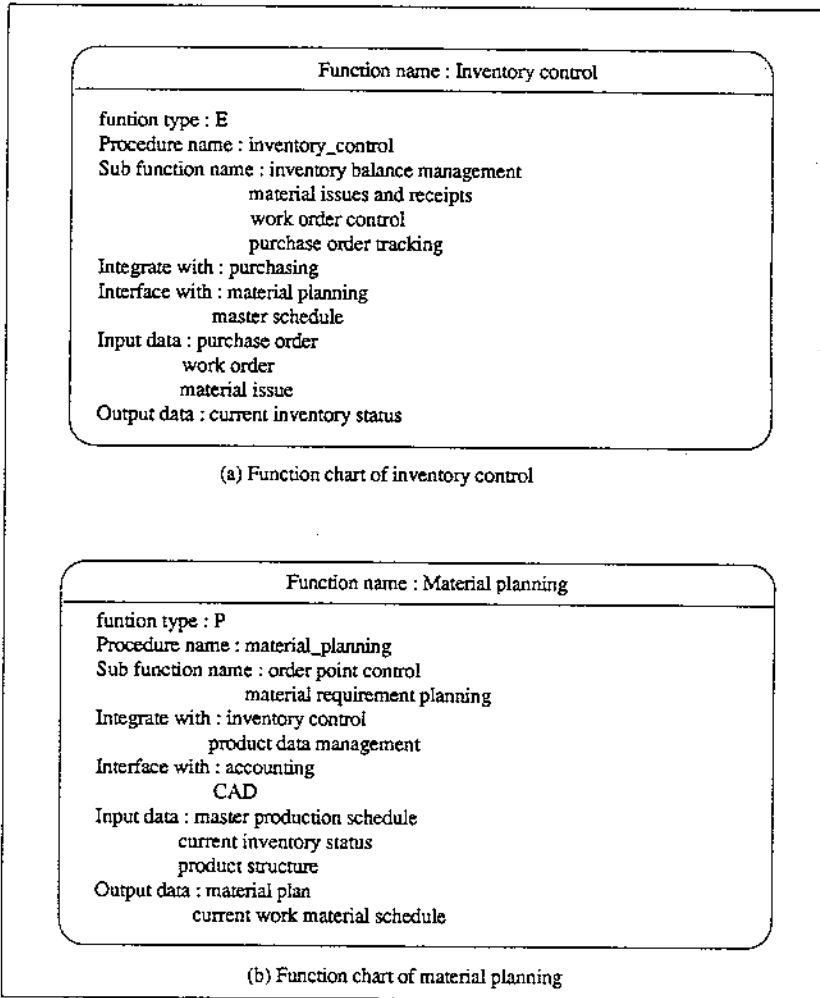


그림 10. Example of function chart

장에서 정의한 객체간의 관계를 이용하여 나타내었고, 이들 객체의 일부를 객체를 나타내는 기호로서 그 내용을 설명하였다.

그림 9의 material planning의 경우 기능도 내부의 'Integrate with'나 'Interface with'에 나타난 기능은 기능의 입출력 또는 제어데이터에 관련되어 있음을 알 수 있다. 이렇게 기능도의 내부에 이들 관계를 나타낼 경우, 기능도에 나타난 데이터 중

에서 이 기능도에 의해서 생성되지 않은 데이터의 출처를 알 수 있으므로 IDEF₀가 나타낸 것보다 더 많은 정보를 알 수 있다. 그림 10은 그림 9에 나타난 기능 중에서 Inventory Control의 기능을 나타낸 기능표(function chart)로서 그림 9에 나타난 기능을 정리하고 이들 기능이 포함하고 있는 하위기능을 알 수 있게하여 기능들 사이의 계층구조를 이해할 수 있다. 한편 그림 11은 그림 9에 나타난 product

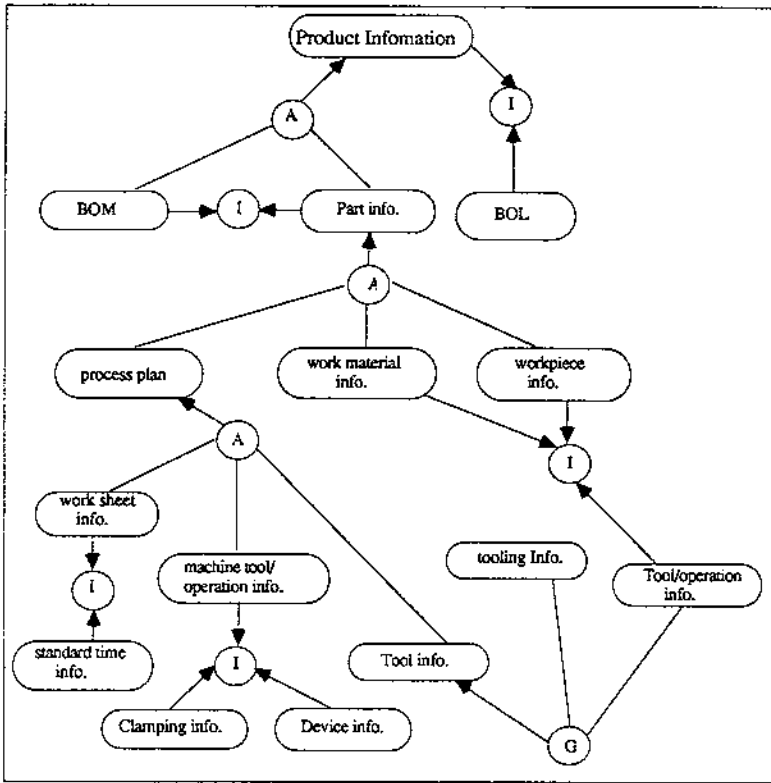


그림 11. Relationship between part structure and other related objects

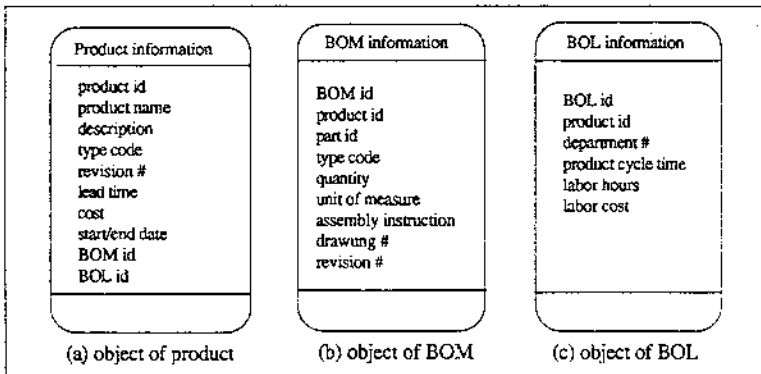


그림 12. Representations of Object

structure의 정보를 나타내는 그림으로 product structure의 정보 구성 및 다른 정보와의 상관관계를

제 4 장에서 제안한 방법으로 나타내고 있다. 그림 11에서처럼 생산시스템의 데이터는 그 특성에 따

라서 다른 데이터와의 결합으로 인하여 상세한 정보를 나타낼 수 있으므로 이들 데이터의 관계가 정의되어야 한다. 또한 그림 12는 그림 11에 나타난 데이터 중에서 그 일부의 내부 구조를 객체로 표현한 것으로 두 그림을 이용하면 객체 사이의 관계를 보다 명확하게 정의할 수 있다.

5. 결 론

CIM의 구축목적은 경영관리, 연구개발, 설계, 생산, 물류등 생산전반에 영향을 미치는 요인들을 고려하여 생산시스템의 총체적 최적해를 이루하는데 있다. 이를 위해서는 생산시스템 설계시에 전체를 하나의 관점에서 볼 수 있는 통합화 측면이 고려되어야 하며, 따라서 common database에 바탕을 통합 생산시스템이 정의되어야 한다. 이러한 통합 생산시스템을 정의하기 위해서는 먼저 시스템을 구성하고 있는 기능 및 데이터들을 일관성있게 분석하고 표현하여 이를 바탕으로 통합 생산시스템에 맞는 정보시스템을 구축하여야 한다.

본 연구에서는 이러한 통합 생산시스템내의 생산정보시스템을 구축하는데, 필요한 시스템분석 및 설계에 관한 것으로 기존의 생산시스템 분석방법인 IDEF, M* GRAI 등을 분석하여 이들 방법이 가지고 있는 시스템 분석상의 문제, 데이터베이스 설계상의 문제, 데이터 표현의 문제 등을 보완할 수 있는 새로운 분석방법 및 표현방법을 중점으로 다루었다. 그 결과 본 연구에서는 생산시스템내의 기능을 기존의 방법인 계층구조 뿐만 아니라 평형구조로 파악하여 이를 나타낼 수 있는 새로운 기능 분석 방법을 제안하였다. 또한 기존의 방법들이 데이터를 단순한 entity로 취급하여 이를 바탕으로 한 relational database의 구축을 목적으로 하였기에 발생되었던 데이터의 의미 표현의 한계를 극복할 수 있는 새로운 data modeling 방법을 제안하였다.

기능에 대한 분석방법으로는 생산시스템의 기능

을 functional decomposition 방법에 의하여 공통 기능, 준비기능, 실행기능으로 세분화하였는데, 이는 각 기능에 따라서 그 기능이 갖고 있는 성격에 따른 고려사항이 다르기 때문이다.

생산시스템을 구축하고 있는 이종의 시스템(heterogeneous system)을 통합할 경우 데이터가 갖고 있는 의미를 통일된 형태로 일관성있게 표현할 수 있는 방법이 필요하다. 따라서 본 연구에서는 생산시스템의 데이터를 그 특성에 따라서 facility data, product data, process data로 구분하였으며, 이들 데이터가 갖는 특성(dynamic, static, spatial)들을 일관성 있게 표현할 수 있는 data model로서 생산시스템과 1:1 mapping이 가능하고 확장이 자유로운 object-oriented data model을 제시하였고, 이들 object 사이의 관계를 aggregation, generalization, interaction 등으로 정의하여 이들 관계 사이의 의미를 파악할 수 있도록 하였다. 또한 분산데이터베이스내에서 complex object의 처리 및 각 데이터베이스간의 데이터 처리를 위하여 버퍼 데이터베이스를 제안하고자 한다.

이상과 같이 정의한 기능 및 데이터를 표현하기 위한 방법으로 각각에 맞는 기호를 정의하였으며, 이들 기호를 사용하여 생산시스템 내에서의 정보 흐름을 파악하는 방법을 설명하였다. 이렇게 시스템을 분석할 경우 다음과 같은 장점이 있다.

- 계층별구조 및 평형구조로 생산시스템의 내부를 파악할 수 있다.
- 통일된 형태로의 데이터의 의미표현이 가능하다.

통합생산을 고려한 생산정보시스템은 생산정보시스템은 다음과 같은 3층 구조를 지니고 있는데, 이질의 하부시스템과 데이터를 어떻게 효율적으로 표현하고 이를 이용하는가에 성공여부가 달려있다.

- 객체지향적 데이터 모델(object-oriented data model)

- 분산데이터베이스 시스템(distributed database management system)
- 전문가시스템(knowledge based expert System)

본 연구는 위의 성격중에서 생산시스템의 기능 분석을 통한 객체지향적 데이터 모델에 관한 것으로 나머지 두가지 성격에 대하여는 추후의 연구가 필요하다. 또한 위의 모델로 정의된 데이터를 바탕으로 생산시스템을 구축할 경우, object-oriented data model로 정의된 데이터를 object-oriented database나 relational database로 mapping하는 문제, 이들 데이터를 knowledge로 사용하여 전문가 시스템을 구축하는 문제, 분산 데이터베이스 구축시 전체의 성능향상을 위한 data allocation 문제 등 실제 시스템 구축시에 필요한 여러가지 문제에 대한 연구도 필요하다.

References

- [1] SME/CASA, CIM A Working Defintion, SME/CASA, 1990.
- [2] Bravoco, R. R. and Yadav, S. B., "Requirement Definition ARchitecture-An Overview," Computers in Industry, Vol. 6, pp.237-251, 1985.
- [3] Bravoco, R. R. and Yadav, S. B., "A Methodology to Model the Functional Structure of an Organization," Computers in Industry, Vol. 6, pp. 345-361, 1985.
- [4] Bravoco, R. R. and Yadav, S. B., "A Methodology to Model the Information Structure of an Organization," Journal of Systems and Software, Vol. 5, pp.59-71, 1985.
- [5] Bravoco, R. R. and Yadav, S. B., "A Methodology to Model the Dynamic Structure of an Organization," Information Systems, Vol. 10, No. 3, pp.299-317, 1985.
- [6] Doumeingt, G., "Design Methodology for Advanced Manufacturing System," Computers in Industry, Vol. 9, pp.271-296, 1987.
- [7] Bray, O. H., CIM : The Data Management Strategy, Digital Press, 1988.
- [8] Di Leva, A. and Vernadat, F., "Information System Analysis and Conceptual Database Design in Production Environments M*," Computers in Industry, Vol. 9, pp.183-217, 1987.
- [9] Peterson, J. L., Petri Net Theory and the Modeling of systems, Prentice-Hall, 1981.
- [10] Hsu, C. and Rattner, L., "Information Modeling for Computerized Manufacturing," IEEE Trnas. on systems, Man and Cybernetics, Vol. 20, No. 4, July, 1990.
- [11] Baudin, M., Manufacturing System Analysis with Application to Production Schedule, Yourdon Press, 1990.
- [12] Weber, D. M. and Moodie, C. L., "Distributed, Intelligent Information System for Automated, Integrated Manufacturing Systems," in advanced Information Technologies for Industrial Material Flow systems edited by Nof, S. Y. and Moodies, C. L., pp.57-80, Springer-Verlag, 1989.
- [13] Barkmeyer, E. and Mitchell, M. and Su, S. Y. W., An Architecture for Distributed Data Management in CIM, U. S. Department of Commerce Report Number NBSIR 86-3312, 1986.
- [14] Su, S. Y. W., "Modeling Integrated Manufacturing Data With SAM*, IEEE Computer, pp.34-49, Jan, 1986.
- [15] Seidewitz, E., "General Object-Oriented Software Development Background and Experience," Journal of Systems and Software, Vol. 9, pp.95-108, 1989.
- [16] Moiaffari, M., "ODM : An Object-Oriented

ted Data Model," *New generation Computing*, Vol. 7, pp.3-35, 1989.

[17] Emod, J. C. and Herman, t., "CIM-OSA Concepts Demonstrated with an Object-Oriented Language," *Proc. 4th CIM Europe Conf.*, pp.265-282.

[18] Jarvinen, O. and kanrad, H., "Object-Oriented Modeling of Production Activity Cotrol Systems," *Proc. 4th CIM Europe Conf.*, pp.353-362, May 1988.

[19] Bu-hulaiga, M. I. and chakravarty. A. K., "An Object-Oriented Knowledge Representation for Hierarchical Real-Time Control of Flexible Manufacturing," *Int. J. Prod. Res.*, Vol. 26, No. 25, pp. 777-793, 1988.

[20] Su, S. Y. W., "An Object-Oriented Semantic Association Model(OSAM*)," in *AI in Industrial engineering and Manufacturing* Edit by Kumara. American Institute of Industrial Engineer, 1988.

[21] Menga, g., "A Framework for Object-Oriented Design and Prototyping of Manufacturing systems," *Proc. Technology on Object-Oriented Languages and Systems '89*, pp.59-68, Nov, 1989.

[22] Peckham, J. and Maryamski, F., "Semantic Data Model," *ACM Computing Surveys*, Vol. 20, No. 3, Sep. 1988.

[23] Hull, R. and King, R., "Semantic Database Modeling: Survey, Applications, and research Issues," *ACM Computing Surveys*, Vol. 19, No. 3, Sep, 1987.

[24] Blaha, M. R., "Relational Database Design Using an Object-Oriented Methodolgy," *Communication of the ACM*, Vol. 31, No. 4, pp.414-427, April 1988.

[25] Coad, P. and Yourdon, E., *Object-Oriented Analysis*, Prentic-Hall, 1990.

[26] suk-in, Yoo, "Design and Implementation of An Object-Oriented Knowledge-Based System," *International Workshop on New Database Technology*, pp. 172-184, 1991.

[27] Navathe, S. and Elmasri, R., "Integrating User Views in Database Design," *IEEE Computer*, pp.50-62, 1986.

[28] Mclean, C. r., "An AMRF Architecture Systems Overview," *Proc. on Factory Standards Model conference*, Nov. 1985.