'91 韓國自動制御學術會議論文集 1991. 10.22~24

객체지향적 접근방법에 의한 생산정보시스템 설계방법

김 철한 김 광수

포항공과대학 산업공학과

An Object-Oriented Design Methodology for Manufacturing Information System

Cheolhan Kim and Kwangsoo Kim
Industrial Engineering
Pohang Institute of Science and Technology

ABSTRACT

A competitive automated manufacturing system integrates the various control processes and data used in service of products, design, manufacturing, and, sale. CIM is a way to achieve such integration through computers and computational techniques in manufacturing, planning, and design. 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 invloves 1) analyzing the functions through functional decompostion, 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 CIM에 대한 관심이 제조업체들 사이에 폭넓게 확산되고 있다. 그러나 일반적으로 CIM을 추구하는 기업들이 느끼는 점은 단위기계 및 시스템의 자동화에 따르는 효과가 기대에 미치지 못한다는점이다. 이는 단위기계나 시스템의 설계시에 통합생산시스템을 위한 고려가 선행되지 못하고 각 기능에 따라서 필요에 따라서 개별시스템을 개발하였으므로 전체를 통합하여 운용할 때 여러가지 문제점이 발생하기 때문이다. 즉, 개별시스템은 통합시스템을 고려한 공통데이타베이스의 기초위에서 설계되어야한다. 이를 위해서는 CIM 시스템을 구성하는 하부시스템에 대한 분석이 행해져야 하는데는 대표적인 시스템 분석방법으로 IDEF[1], GRAI[2], NIAM[3], M* [4] 등이 있다. 그러나 이들 분석방법은 사무용 데이타처리를 위한 분석방법에 근거를 두고있어 생산시스템을 분석하는데는 제한점들이 있다.[5]

IDEF경우에는 시스템을 functional, information, dynamics model등으로 시스템을 분석하고 있어 각 모델에 의한 시스템의 설명이 용이하나, 이를 바탕으로 새로운 시스템의 설계시에 조직도나 인터페이스를 고려하지 않았기 때문에 전체를 하나의 view로 파악하여 데이타베이스를 설계하기에는 한계가 있다. GRAI경우에는 생산시스템내에서 사용되고 있는 의사결정구조에 중점을 두고 분석할 수 있는 tool이며 M*는 데이타베이스설계를 목적으로하여 생산시스템을 분석하기 위한 방법이지만 하나의 데이타베이스로 설계하기위한 것으로 생산시스템이 갖고 있는 분산의 성격을 고려하지 않았으므로 한계가 있다. 또한 Baudin[6]은 생산시스템을 DFD로 분석하였으나, Hsu와 Rottner[5]가 언급하였듯이 DFD는 process-oriented의 계층구조이기 때문에 view의 생성이나 semantic constraints등의 면에서 한계가 있다.

따라서 본 연구에서는 이들 대표적인 시스템 분석방법들이 갖고 있는 한계들을 보완하는 새로운 생산시스템 분석방법을 제시하고자 한다.

2. 새로운 생산시스템 분석방법

본 연구에서는 객체지향적 방법에 의한 새로운 생산정보시스템 설계방법을 제안한다. 이를 위하여 func- tional decomposition 방법을 이용하여 시스템내의 기능들을 평형구조 또는 계층구조로 분석하여 각 기능내에서 사용되고 있는 데이타를 주변 기능과의 관계를 고려하여 IDEF^{0}에서 정의한 것과 같이 input, output, control, mechanism data로 구분하였다. 각 기능에 따라서 그 의미가 달라지는 생산시스템내의 데이타를 각 기능에서의 의미를 포함하면서 일관성있게 나타내기 위하여 객체지향적 데이타모델 (object- oriented

data model)[7,8]을 이용하여 표현하였다. 객체지향적 데이타모델은 생산시스템의 데이타가 갖는 복잡성과 이질성들을 표현할 수있으며 객체지향적방법이 갖고있는 특징들 즉, encapsulation, polymorphism, inheritance, abs- traction, 등을 이용하여 각 기능에 따라 달라지는 데이타의 의미를 쉽게 구현할 수 있다.

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

1) 분석범위의 정의

생산시스템은 원자재로부터 최종상품이 만들어지는 과정인 제품설계, 공정설계, 생산계획, 생산관리, 생산공정을 포함하는 시스템으로 본 연구에서는 CIM구축시에 요구되는 생산정보시스템을 분석범위로 정의하였다.

2) 기능의 분석

생산시스템을 functional decomposition 방법을 이용하여 하부시스템으로 분석할 경우 다음과 같은 기준으로 분류하고자 한다.

. 정보흐름 (information flow)의 최소화
. 동일공정 또는 연속공정의 집합으로서 일관성을 유지
. 데이타의 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 동일기능으로 간주함으로써 정보의 흐름이 최소화할 수 있도록 기능을 black box화하였다. 생산시스템내의 정보망은 형태상 분산데이타베이스를 기준으로 하기 때문에[9,10] 데이타의 투명성을 유지할 수 있도록 기능들이 구분 되어야 한다. 나머지 두가지는 나뉘어진 기능을 그 활동 영역에 따라 분류하는 기준이다.

이러한 개념으로 기능을 공통기능, 실행기능, 준비기능으로 구분할 수 있다. 공통기능이란 의사 결정기구를 갖지않는 기능들로 다른 기능들을 지원하는 master file의 생성 및 유지보수를 담당하는 part master와 같은 기능들이며, 실행기능이란 생산시스템의 작동을 수행하는 기능들로서 외부의 사건에 의해서 작동되는 production control등과 같은 event-driven function들로서 이들 기능들은 그 기능이 갖고있는 의사결정기구를 정의하여야하며 이들 의사결정구조가 다른 기능들과 갖는 관계를 계층적구조로서 뿐만 아니라 평행구조로서도 파악해야한다. 준비기능이란 실행기능이 수행될 수 있도록 준비하는 기능으로 자재나 장비등에 물리적인 영향이 미치지않는 수요예측등과 같은 기능들로 실행기능과 병행하여 또는 실행기능과 관계없이 발생할 수 있다. 이처럼 정의된 기능들은 각각의 procedure와 기능들의 하부기능, 기능의 입출력을 담당하는 데이타가 정의되어야 하며 functional decomposition의 결과로 그림 1와같은 기능도와 기능표를 정의할 수 있다. 본 연구에서는 기능사이의 정보흐름을 알기위하여 IDEF0와 같은 형태의 기호를 사용하였으나, 그림 1에서처럼 box 안의 기능을 기능들 간의 평형구조를 고려하여 관계있는 주변기능들을 'Interface with'와 'Integrate with'로 나누어 기술하였다. 'Interface with'의 관계는 단순히 데이타를 참조하는 기능들이고, 'Integrate with'는 관련된 기능들의 데이타가 서로에게 영향을 미치는 기능들이다. 후자의 경우, 이들 사이에 있는 데이타의 생성 및 수정들은 연관된 모든 기능들을 고려해야 한다. 즉, 시스템분석 후에 이를 바탕으로하여 분산정보시스템을 구축할 때 필요한 global dictionary내에서 이들 기능들 간의 관계를 정의하기 위하여 'Integrate with'와 'Interface with'의 구분이 필요하다. 이들 기능들을 데이타의 입출력에 맞추어 연결하면 정보흐름에 따른 기능들의 역할 및 관계를 알 수 있게된다.

3) 데이타의 분석

위에서 정의한 기능에 따르는 생산시스템의 데이타는 시간, 공간의 의미를 갖기도 하고, 자체내에서 계층구조를 갖기도 하는등 일반적인 형태로는 그 표현에 한계가 있다.[10] 따라서, 생산시스템에서 사용되고 있는 데이타를 표현한다는 것은 간단한 것이 아니다. 우선 생산시스템에 사용되고 있는 데이타는 성격상 공정데이타, 제품데이타, 설비데이타로 구분할 수 있는데, 여기서 공정데이타는 주로 실행기능에 사용되는 데이타로서 의사결정구조에 따라서 데이타가 변화하며, 의미가 달라질 수 있는 데이타이고, 제품데이타는 주로 준비기능 및 공통기능에 사용되는 데이타로서 생산시스템의 최종출력인 제품에 관한 데이타이다. 설비데이타는 제품데이타처럼 준비기능 및 공통기능에 사용되는 데이타로서 생산시스템을 구성하는 장비나, 사람, 공구등에 관계된 데이타이다.

이렇게 정의된 데이타는 그 특성에 따라서 표현방식이 달라지게 되므로 이들 데이타를 entity로 표현하고 이를 처리하는 RDBMS로는 생산시스템 데이타가 갖고있는 다음과 같은 특성을 포함하기에는 한계가 있다.[10]

. 이질의 객체들의 집합체로서의 데이타를 표현
. matrics 또는 시간이나 공간의 데이타를 표현
. 설계데이타의 변화를 지원하는 versioning기능
. 복합구조(complex object structure)를 표현

따라서 생산시스템의 데이타가 위에서 언급한 특징을 만족하기위하여 여러가지 의미 및 형태를 포괄적으로 표현할 수 있는 객체지향적 데이타모델을 사용하였다. 그림 2는 객

체를 표현한 그림으로 object identifier는 객체를 나타내는 명칭이며, attribute는 그 객체가 갖는 성질을 의미하며 method는 객체와 객체사이의 메세지 전달에 의해서 행해지는 procedure나 알고리즘등을 의미하고 굵은 화살표는 객체사이의 메세지 교환이 있음을 의미한다. 이들 특징중에서 객체, 객체 사이의 관계, instance 그리고 method등을 체계적으로 정의할 수 있는 추상화의 개념은 generalization, aggregation, interaction 등의 세가지의 형태로 구분할 수 있다.

Generalization은 개개의 객체(subtype)를 일반적인 하나의 객체(type)로 표현하는 것으로, 이때 type은 subtype사이의 공통성질를 갖게되는데, 각 subtype은 고유의 성질만 갖고 나머지 공통성질은 type으로부터 상속받는다.

Aggregation은 assembly-subassembly등의 부품구조에 유용한 것으로 여러개의 하위클래스들이 하나의 상위클래스를 구성하게 될 때 사용된다.

Interaction의 경우는 계층적으로 관련되어 있지 않은 클래스 사이의 관련성 (relationship)을 나타내는데 사용되며, 동일 set으로 나타낼 수 있는 클래스들의 관계를 나타낸다.

이러한 특징들을 가진 객체는 관계형 데이타베이스로 구축이 가능하다.[11] 즉, 기존의 객체지향적 프로그래밍언어(OOPL)와 RDBMS를 사용하여 객체지향적 방법에 의한 생산정보 시스템을 구현할 수 있다.

(4) Integration view

위에서 언급한 기능들과 이들 기능에 관련된 데이타가 통합시스템 내에서 운용되려면 데이타의 투명성을 유지하면서 각 기능들의 view가 통합되어야 한다.[12] 미국의 NBS에서 제안한 생산시스템의 계층구조[13]를 이용하여 위에서 분류한 기능들이 각각 어느 계층들에서 사용되는지를 정의하는 그림 3과 같은 functional cross table을 작성하여 이를 바탕으로 global dictionary가 구축되어야 한다. Global dictionary는 local view들이 갖는 local dictionary의 통합으로서 분산 데이타베이스 구축시에 각 local 데이타베이스의 영역을 할당하며, 각 기능에서의 data의 생성, 유지, 보수 및 검색의 범위를 명확하게 구분하여 각 data base의 접근권한이 정의되어야 한다.

3. 응용 예

본 장에서는 material management에 관계되어 있는 기능의 정보흐름을 제 2장에서 정의한 기호를 사용하여 나타내었다. 이들 데이타 중에서 제품구조에 관계된 데이타의 구조를 제 2장에서 정의한 객체 간의 관계를 이용하여 나타내고 이들 객체의 일부를 객체를 나타내는 기호로서 그 내용을 설명하였다.

그림 4의 material planning의 경우 기능도 내부의 'integrate with'나 'Interface with'에 나타난 기능은 기능의 입출력 또는 제어데이타에 관련되어 있음을 알 수 있다. 그림 5은 그림 4에 나타난 기능중에서 재고관리의 기능을 나타낸 기능표로서 그림 4에 나타난 기능을 정리하고 이들 기능이 포함하고 있는 하위기능을 알 수 있게하여 기능들간의 계층구조를 이해할 수 있다. 한편 그림 6은 그림 4에 나타난 제품구조의 정보를 나타내는 그림으로 제품구조의 정보구성 및 다른 정보와의 상관 관계를 제 2장에서 제안한 방법으로 나타내고 있다. 그림 6에서 처럼 생산시스템의 데이타는 그 특성에 따라서 다른 데이타와의 결합으로 인하여 상세한 정보를 나타낼 수 있으므로 이들 데이타의 관계가 정의되어야 한다. 또한 그림 7은 그림 6에 나타난 데이타 중에서 그 일부의 내부 구조를 객체로 표현한 것으로 두 그림을 이용하면 객체 사이의 관계를 보다 명확하게 정의할 수 있다.

4. 결 론

CIM의 구축을 위해서는 생산시스템 설계시에 전체를 하나의 관점에서 볼 수 있는 통합화 측면이 고려되어야 하며, 따라서 공통 데이타베이스에 바탕을 통합 생산시스템이 정의되어야 한다. 이러한 통합 생산시스템을 정의하기 위해서는 먼저 시스템을 구성하고 있는 기능 및 데이타들을 일관성있게 분석하고 표현하여 이를 바탕으로 통합생산시스템에 맞는 정보시스템을 구축하여야한다.

본 연구에서는 이러한 통합생산시스템의 생산정보시스템을 구축하는데 필요한 시스템분석 및 설계에 관한 것으로 기존의 생산시스템 분석방법인 IDEF, M*, GRAI등을 분석하여 이들 방법이 가지고 있는 시스템 분석상의 문제, 데이타베이스 설계상의 문제, 데이타 표현의 문제등을 보완할 수 있는 새로운 분석 방법 및 표현방법을 중점으로 다루었다. 그 결과 본 연구에서는 생산시스템 내의 기능을 계층구조와 평형구조로 파악하여 이를 나타낼 수 있는 새로운 기능 분석방법을 제안하였다. 기존의 방법들이 데이타를 단순한 entity로 취급하여 이를 바탕으로 한 관계형 데이타베이스의 구축을 목적으로 하였기에 발생되었던 데이타의 의미 표현의 한계를 극복할 수 있는 새로운 데이타 모델링 방법을 제안하였다.

기능에 대한 분석방법으로는 생산시스템의 기능을 그 기능이 갖고 있는 성격을 고려하여 functional decomposition 방법에 의하여 공통기능, 준비기능, 실행기능으로 세분화하였다. 생산시스템을 구축하고 있는 이종의 시스템을 통합할 경우 데이타가 갖고있는 의미를 통일된 형태로 일관성있게 표현할 수 있는 방법이 필요하므로 생산시스템의 데이타를 그 특성에 따라서 설비데이타, 제품데이타, 공정데이타로 구분하였으며, 이들 데이타가 갖는 특성들을 일관성 있게 표현

할 수 있는 데이타모델로서 생산시스템과 1:1mapping이 가능하고 확장이 자유로운 객체지향적 데이타모델을 제시하였고, 객체사이의 관계를 aggregation, generalization, interaction 등으로 정의하여 객체들사이의 관계를 파악할 수 있도록하였다. 이상과 같이 정의한 기능 및 데이타를 표현하기 위한 방법으로 각각에 맞는 기호를 정의하였으며, 이들 기호를 사용하여 생산시스템 내에서의 정보흐름을 파악하는 방법을 설명하였는데 이 경우 다음과 같은 장점이 있다.

. 계층별구조 및 평형구조로 생산시스템의 내부를 파악할 수 있다.

. 통일된 형태로의 데이타의 의미표현이 가능하다.

통합생산을 고려한 생산정보시스템은 다음과 같은 3중구조를 지녀고있는데 이질의 하부시스템과 데이타를 어떻게 효율적으로 표현하고 이를 이용하는가에 성공여부가 달려있다.

. 객체지향적 데이타 모델

. 분산데이타베이스 시스템

. 전문가시스템

본 연구는 위의 성격중에서 생산시스템의 기능분석을 통한 객체지향적 데이타 모델에 관한 것으로 나머지 두가지 성격에 대하여는 추후의 연구가 필요하다. 또한 위의 모델로 정의된 데이타를 바탕으로 생산시스템을 구축할 경우, 객체지향적 데이타모델로 정의된 데이타를 객체지향적 데이타베이스나 관계형 데이타베이스로 mapping하는 문제, 이들 데이타를 knowledge로 사용하여 전문가 시스템을 구축하는 문제, 분산데이타베이스 구축시 전체의 성능향상을 위한 데이타 할당 문제 등 실제 시스템 구축시에 필요한 여러가지 문제에 대한 연구도 필요하다.

참고문헌

[1] R.R.Bravoco, and S.B.Yadav,"Requirement Definition Architecture-An Overview", Computers in Industry, Vol.6, pp.237-251, 1985

[2] G. Doumeingt,"Design Methodology for Advanced Manufacturing Systems" Computers in Industry, Vol.9, pp. 271-296, 1987

[3] J.J.V.R.Wintraecken, The NIAM Information Analysis Method Theory and Practice, Control Data, 1989

[4] A.Di Leva et al., "Information System Analysis and Conceptual Database Design in Production Environments with M*", Computers in Industry, Vol. 9, pp. 183-217, 1987

[5] C.Hsu et al., "Information Modeling for Computerized Manufacturing", IEEE Trans. on Systems, Man ,and Cybernetics, Vol.20 No.4, July 1990

[6] M.Baudin, Manufacturing System Analysis with Application to Production Schedule, Yourdon Press, 1990

[7] M.Moiaffari, "ODM: An Object-Oriented Data Model", New generation Computing, Vol.7, pp. 3-35, 1989

[8] G.Menga, "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

[9] D.M.Weber et al., "Distributed, Intelligent Information System for Automated, Integrated Manufacturing Systems", in Advanced information Technologies for Industrial Material Flow Systems edited by S.Y.Nof and C.L.Moodies, pp.57-80, Springer-Velag, 1989

[10] E.Barkmeyer et al., An Architecture for Distributed Data Management in CIM, NBSIR86-3312, 1986

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

[12] S.Navathe et al., "Integrating User Views in Database Design", IEEE Computer, pp.50-62, 1986

[13] C.R.Mclean, "An AMRF Architecture Systems Overview", Proc. on Factory Standards Model, 1985

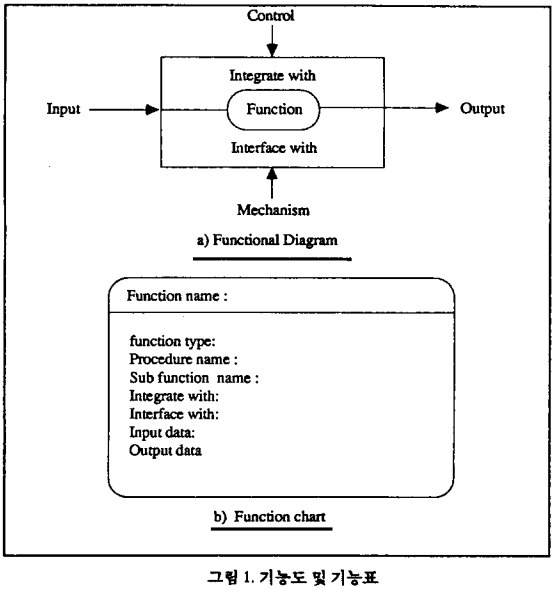

그림 1. 기능도 및 기능표

Figure 1. Functional diagram and Functional cahrt

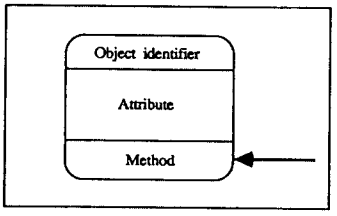

그림2. 객체의 표현

Figure 2. Object representation

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

LEVEL

L1 : Plant Area
L2 : Area Level
L3 : Cell Level
L4: Machime Level
L5:Sensor Level

FUNCTION

F1:Part Master
F2: BOM
F3:Forecasting
F4:Master Schedule
F5:Cost Planning
F6: Facility Management
F7: Process Planning

그림 3. 기능 일람표

Figure 3. Functional cross table

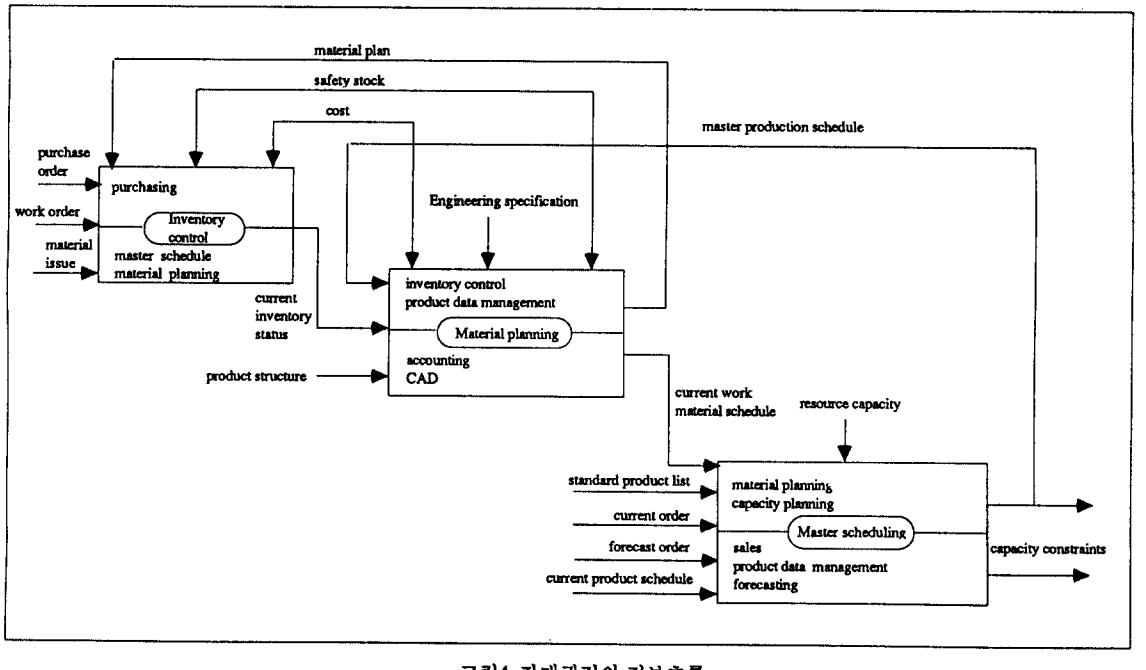

그림4. 자재관리의 정보흐름

Figure 4. Information Flow of Material Management

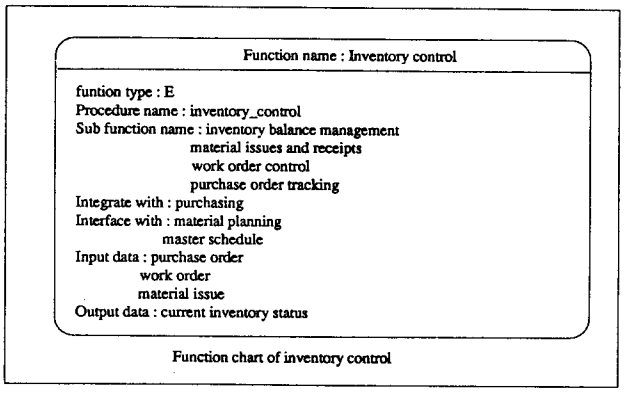

그림5. 기능도의 예

Figure 5. Example of Function chart

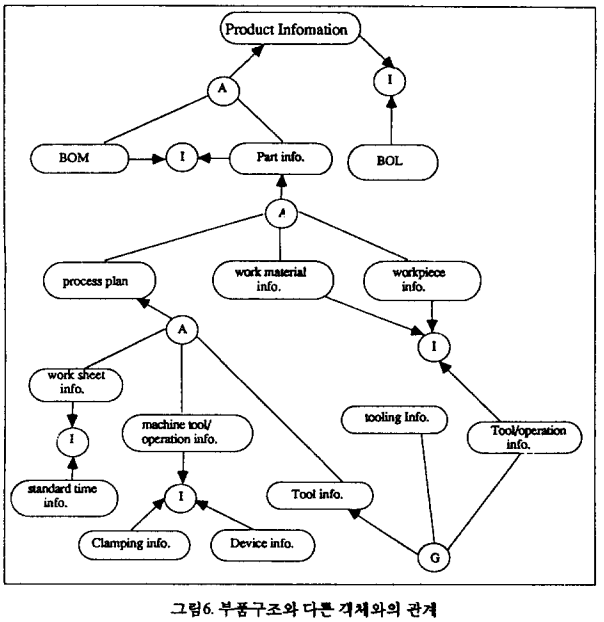

그림6. 부품구조와 다른 객체와의 관계

Figure6. Relationship between part structure and other related objects

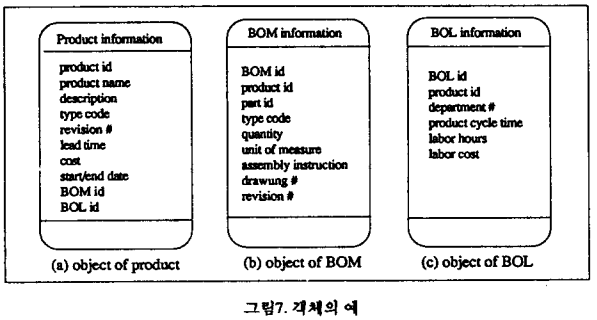

그림7. 객체의 예

Figure7. Representations of Object