

하도록하고 설계불량인 경우는 다시 금형설계를 한다. 또한 실적데이터의 수집 및 이력관리도 병행된다.

이상의 금형 생산과정을 충실히 반영할 수 있는 플랜트 통제시스템을 개발함에 있어, 처리되는 데이터의 Integrity와 기능의 Integrity에 따라 6개의 모듈로 구성하였으며 그 각각의 기능은 Table 1과 같다.

Table 1. 금형플랜트 통제시스템의 모듈별 기능

모듈	기능
작업 관리	생산능력계획을 통한 수주검토 금형의 Type, 재질의 결정 및 표준품 결정 제조오더의 수령, 변경, 삭제 및 출력 구매, 외주, 설계오더의 발주
작업계획 및 제어	주정공정데이터 입력을 통한 Rough Scheduling 제조오더의 일정계획 및 부하조정(Fine Scheduling) 작업준비 확인 및 작업지시서 발급 긴급오더, 가공불량품등의 일정조정 고장, 지연등 비상사태의 일정예외의 반영
Shop Monitoring	오더 및 공정의 진행상태 입력 비상사태정보 수집 및 보고 계획과 실제수행간의 일정확인 및 보고 현장배치도의 화면출력 및 기계상태의 도시
NC 프로그램 관리 및 분배	NC 프로그램의 수령, 등록 및 소거 NC 프로그램 목록관리 NC 프로그램의 장치(Blocking) 및 준비(Release) Shuttle을 통한 NC 기계로의 Down-load
공구 관리	공구의 등록, 변경, 삭제 및 출력 일정에 따른 공구의 준비 및 투입 공구의 소재파악 및 일정확인을 통한 재고관리 공구수명 감시
시스템 실적 관리	계획과 실제수행간의 편차집계 및 통계처리 가공비용, Leadtime의 계산 및 분석 고장원인 집계 및 분석 집계처리된 결과의 표준데이터로의 갱신

3. 업무분석을 통한 Relation 구성

(1) 업무-수행과정 모델화(Task-Step Modeling)

데이터베이스를 구성하기 전에 시스템의 기능분석을 통한 업무-수행과정 모델을 만들게 되는데 여기서는 그 모듈의 주요기능을 나열하고 거기에 따른 수행순서를 나열한다. 다음으로 관리하고자 하는 대상(Entity)은 무엇이고 그에 속한 데이터항목(Attribute)은 무엇인가를 도표화 할 수 있는데 이것이 다음 Relation 구성의 기초가 된다. Table 2와 Table 3은 작업관리 모듈의 기능 중 일부를 도표화한 것이다.

(2) Relation 구성

Relational 데이터베이스에서 데이터는 Relation이라고 부르는 2차원 테이블(Table) 속에 거주한다. Relation은 행(Row)과 열(Column)로 구성되어 있는데, 행은 데이터

Table 2. 작업관리 모듈의 업무-수행과정 모델(발체)

업무	수행과정
금형 등록	금형번호, 수주일, 납기등의 입력 금형 Type, 재질등의 결정 표준품 검색 및 구성품목 결정 금형의 생산예정일 부여
품목 등록	모든 품목에 대해 제조, 구매, 또는 외주 Code 부여 표준품인 경우 공정데이터 검색 및 추출 금형설계과에 공정설계 지시
오더 등록	설계도면 요구일, 자재 입고일 부여 작업의 긴급도에 따른 우선순위 결정 금형의 생산예정일을 토대로 오더의 제조예정일 부여
외주 등록	외주 발주일, 납기(Due Date) 결정 외주업체 결정 및 비용등의 기록

Table 3. 작업관리 모듈의 데이터항목표(발체)

Entity	Attribute	Entity	Attribute
Mold	Mold Number	Order	Order Number
	Mold Description		Order Description
	Mold Type Number		Planned Start Date
	Customer Company		Planned End Date
	Contract Date		Material Receipt Date
Component	Delivery Date	External Contract	Drawing Receipt Date
	Planned Date		Priority
	Quantity		Component Number
	Component Number		Operation Number
	Component Description		Company Name
Mold Number	Component Drawing No	Contract	Release Date
	Manufacturing Code		Due Date
			Cost

레코드(Record)를 표시하고 열은 레코드내의 각 필드(Field)들을 나타낸다. 레코드내의 하나의 필드에는 단지 하나의 데이터만이 저장된다.

앞에서 작성한 데이터항목표를 토대로 Relation을 만들면 대개의 경우 데이터 중복이나 Relation간의 연결오류(Linkage Error)가 발생한다. 이를 바로잡기 위해 다음과 같은 세 단계의 정규화(Normalization)를 수행한다. 첫째 단계는 한 Relation 내에서 배열 형태로 저장되어야 하는 필드들을 또 하나의 Relation으로 독립시키는 과정이다. 둘째 단계는 복합(Compound) Key를 갖는 Relation에서 Key가 아닌 Attribute가 Key의 일부분만으로 결정되는 경우를 배제한다. 셋째 단계는 어떤 Key가 아닌 Attribute가 또 다른 Key가 아닌 Attribute에 의해서 결정되는 경우를 없애는 작업이다.

두 Relation간의 연결에는 1:1(One to One), 1:M(One to Many), M:1(Many to One), 그리고 M:M(Many to Many)의 4가지를 생각할 수 있는데 정규화를 수행하면 1:M 구성만 남게된다. Fig. 2에서 금형과 품목은 1:M 구조를 갖고 있음을 보여준다. 반면 품목과 제조오더의 경우는 1:1

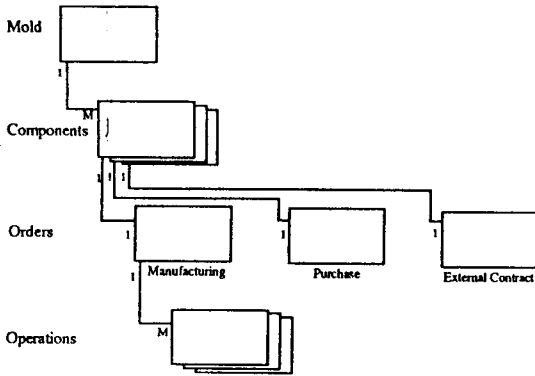


Fig. 2. 오더 구성을 위한 계층적 표현

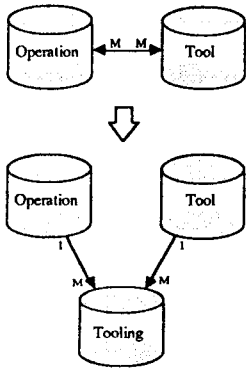


Fig. 3. M:M 관계를 갖는 Relation간의 정규화

구성을 갖는데 이는 정규화 개념에 의하면 불필요한 Relation을 하나 더 만든 것이다. 하지만 제조, 구매, 외주를 하나의 오더로 보면 역시 품목과 1:M 관계를 유지함을 보여준다. Fig. 3에서 각 공정은 여러개의 공구를 사용할 수 있고 각 공구는 여러 공정에서 사용할 수 있음을 보여 준다. 이 M:M 관계를 해소하기 위해서는 새로운 Relation을 도입하여 해결한다.

4. 데이터 구조

4.1. 데이터베이스 구성

생산통제를 위하여 중요한 정보는 크게 기술정보와 관리정보로 나눌 수 있다. 기술정보는 공정정보를 중심으로 하는 공작물 관련 정보들이고, 관리정보는 작업의 우선순위나 일정과 같이 작업오더와 관련되는 정보들이다. 그리고 작업의 현황을 나타내는 현황정보와 NC 프로그램, 공구, 치구, 설비 및 재고상황을 나타내는 자원(Resource) 정보들이 중요하다. 또한 금형의 표준화가 진척됨과 아울러 주어진 금형 Type에 대해 구성품목과 그들의 공정정보를 별도로 저장하여 표준품이 선택된 경우 바로 이용하고자 하는 필요성이 증대되었다.

앞서 언급한 금형공장의 전반적인 흐름을 수용하고 표준품과 그 표준공정을 관리하기 위해 Multi-file 데이터베이스를 구성하여 운영상 필요한 정보는 PIS 데이터베이스에서 관리하고 주로 조회만 하는 표준품에 대한 정보는 STD_PIS 데이터베이스에 관리하도록 하였다. 이 데이터베이스내의 각 Relation들 간의 관계를 Fig. 4에 나타내었다. 여기서 원통형 동들은 각각의 Relation을 의미하며 이들을 연결하는 화살표 위의 1 또는 M은 그들의 관계를 나타낸다.

4.2. 운영 데이터베이스와 표준 데이터베이스

(1) 운영 데이터베이스(PIS Database)

금형 한 벌을 수주받게되는 경우 그것은 어떤 품목(Component)들로 이루어져 있으며 외주를 할 것인지, 자체 제작을 할 것인지를 결정하여야 한다. 또 그 품목이 표준품이면 STD_PIS 데이터베이스로 부터 공정데이터를 복사해 오고 그렇지않은 경우 공정계획의 수행을 통해 공정순서와 공정데이터를 구하고 이를 저장해 두어야 한다. 제작품의 경우 제조오더로 등록하고 앞서 결정된 공정데이터를 토대로 일정계획을 수행하여 이를 각각의 공정으로 저장한다.

기계, 공구, NC 프로그램, 그리고 작업자에 관한 정보도 별도로 관리하여 일정의 효율적 수행을 지원하도록 하며 고장등 일정지연을 유발하는 사항들도 따로 관리한다. 현장의 휴무일, 잔업등을 반영하기 위해서는 공장력(Factory Calendar)의 관리도 필요하다.

이상의 데이터들은 끊임없이 입력, 변경, 삭제가 이루어지므로 이들의 관리에는 많은 주의를 요하며 특히 두 사람 이상이 이 데이터베이스에 접근(Access)하는 경우에도 대해서도 염두에 두어야 한다.

(2) 표준 데이터베이스(STD_PIS Database)

금형의 표준화로 Type번호만 주면 해당품목들의 목록을 자료집을 통해 쉽게 얻을 수 있다. 이러한 표준품의 경우 전문회사로 부터 직접 구매할 수도 있고 제작해야 하는 경우도 있는데 그 경우 공정데이터는 자료를 통해서나 공정설계를 통해 한 번 등록하면 차후 두고두고 쓸 수 있다는 장점이 있다.

금형 Type은 표준을 정한 회사마다 차이가 있으나 보통 순수 Type과 크기를 하나의 코드(Code)화 하여 사용하는 데 그 종류는 1400가지 정도로 알려져 있다. 이 각각은 2,30가지의 품목으로 구성되어 있어 이를 데이터베이스에 저장할 경우 큰 Memory를 필요로 할 것이다. 그러나 다행스럽게도 서로 다른 금형일지라도 같은 품목을 갖는 것이 많아 실제로 관리해야 할 품목의 수는 2000개 안팎이다.

금형 Type으로 부터 표준품을 찾고 다시 공정계획(Process Plan)을 구한다. 다음으로 각 공정계획에 대한 공정순서와 공정데이터를 구해 일정계획시 도움을 주고자 하는 것이 이 데이터베이스의 구성 목적이다.

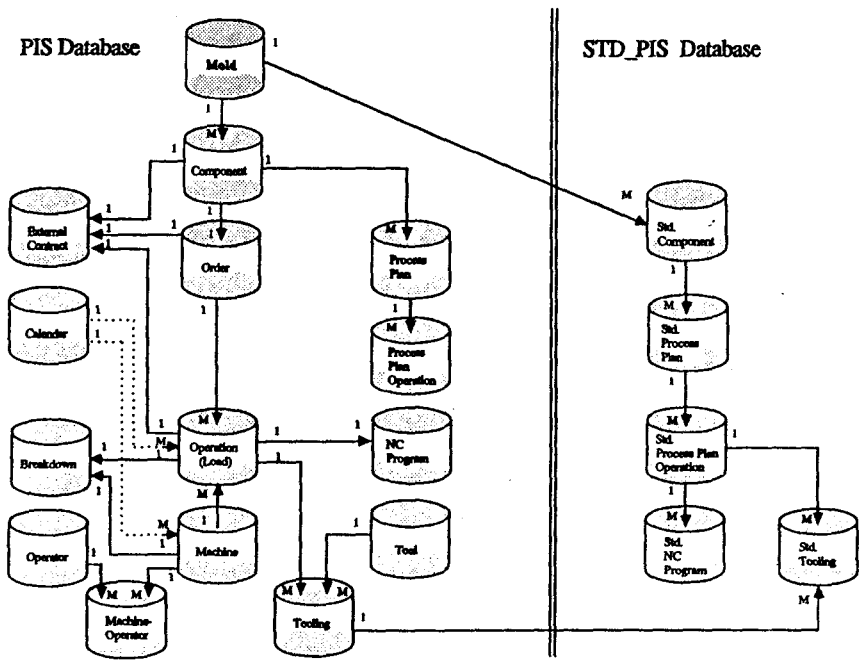


Fig. 4. 금형플랜트 통계시스템을 위한 데이터 구조

4.3. 데이터 구조의 이용 사례

금형플랜트 통계시스템의 기능은 앞서 설명한 자기 모듈화된 포괄적인 기능 외에도 Gantt Chart, 부하상태도 (Load Profile), 공정타임블과 같이 화면출력을 통한 지원기능과 공장력과 같이 일정을 계산하기 위해선 반드시 거쳐야 하는 내부적인 지원기능이 있다. 이 밖에도 표준품 선정, 공정계획의 이용, 외주품 관리등이 모듈화 이전의 기본 기능이라 할 수 있다. 이들을 수행하기 위한 데이터베이스내의 흐름을 정리한다.

(1) Gantt Chart

Gantt Chart는 오더단위로 그려진다. 따라서 이의 작도를 위해서는 먼저 오더번호를 입력받은 후 오더 Relation으로 가서 제조예정일, 우선순위의 데이터를 읽어온다. 이 제조예정일이 Gantt Chart가 출력할 시간대역 (Time Horizon)이 된다. 다음으로 공정 Relation으로 가서 주어진 오더번호를 갖는 공정들을 가공시작예정일을 기준으로 정렬(Sorting)한 후 그 순서대로 화면에 그리면 된다. 오더와 공정의 Relation으로 부터 Gantt Chart를 그리는 예를 Fig. 5에서 보여 주고있다.

(2) 부하상태도(Load Profile)

이것은 기계단위로 그려진다. 기계번호가 입력되면 기계 Relation으로 가서 해당 Record를 읽어오고 다음으로

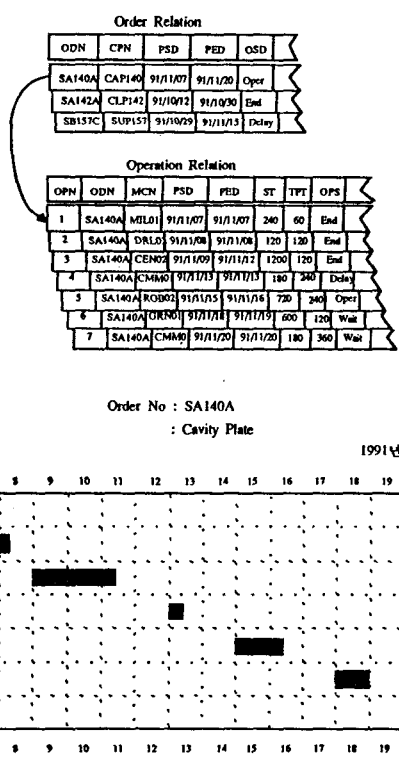


Fig. 5. Gantt Chart의 작도 예

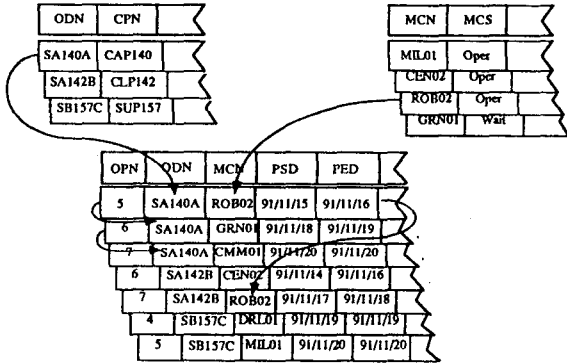


Fig. 6. Relation에서의 공정과 부하 관계 도식

부하 Relation으로 가서 주어진 기계번호를 갖는 부하들을 가공시작예정일을 기준으로 정렬한다. 그리고 이들을 주어진 시간역에 대해 날짜 순으로 도시한다. 기계에 대한 부하정보는 오더에 대한 공정정보로부터 얻어짐으로 부하 Relation과 공정 Relation은 같은 Relation이다. 이들의 관계를 Fig. 6에 나타내었다.

(3) 공정테이블(Process Table)

작업중인 오더의 진행상태나 기계의 Queue에 들어있는 부하를 테이블의 형태로 보고자 하는 경우에 유용하다. 현장에 있는 오더들의 목록을 보고싶은 경우에는 오더 Relation에서 모든 Record들을 오더번호로 정렬시켜 테이블에 출력한다. 이 과정에서 여러가지 조건을 줄 수 있는데 어떤 Code를 갖는 오더만 출력시키거나 제조시작예정일이 어떤 범위에 드는 경우만 보여주는 방법등이 그 예이다.

(4) 공장력(Factory Calendar)

표준시간, 운송시간등으로 관리되는 공정데이터를 날짜로 환산하거나 휴무일, 잔업등을 관리하여 일정계획에 반영하기 위해서는 공장력을 작성해야 할 필요가 있다. 그 날그날의 유효작업시간을 데이터베이스에 저장한 후 이를 기준으로 날짜와 시간을 구할 수 있다. 기계별로 잔업을 하는 경우도 많아 같은 기법의 기계력(Machine Calendar)의 관리도 필요로 한다.

(5) 표준 공정데이터의 이용

한 표준품의 공정계획도 여러개가 존재할 수 있다. 예를들면 같은 형상이라도 재질에 따라 공정데이터가 크게 달라질 수 있어 이들을 관리하는 Relation이 필요하다. 한 공정계획에 대한 각각의 공정정보들은 별도의 Relation으로 따로 관리한다. 여기에는 가공순서, 가공시간, 적용 가능한 기계등의 정보가 입력되는데 기계의 경

우 특정 기계를 직접 입력해 두고 실제 이용할 때는 Parallel Machine의 개념을 도입하여 사용 가능한 기계를 찾도록 한다.

주어진 급형의 Type번호로 부터 표준품의 목록을 구하고 각 표준품의 공정데이터를 얻는다. 다음으로 운영 데이터베이스의 공정계획 Relation에 이들을 복사해 두고 공정 Relation의 구성시여기의 데이터를 참조한다. 데이터의 복사라는 측면에선 비효율적인 구성이라 할 수 있으나 표준데이터의 성격, 일정계획의 잦은 변경을 고려하면 운영상 매우 편리한 구성이다.

5. 결론

다품종 단일수량 생산방식, 공정의 복잡성, 일정변경 및 조정의 빈번한 발생, 단납기화 요구의 증가등 급형공장이 안고 있는 근본적인 복잡성을 수용하여 그 통제를 원활하게 수행할 수 있는 정보시스템은 잘 알려져 있지 않은 상태이다.

급형공장의 통제시스템은 이와 같은 동적인 변화에 신속, 정확하게 대응할 수 있도록 의사결정을 지원하는 기능을 가져야 한다. 이를 위하여는 방대한 양의 정보를 신속히 처리할 수 있는 효율적인 데이터 구조가 요구된다. 본 연구에서는 급형플랜트 통제시스템을 위한 데이터 구조 설계시 접근방법 및 그 구현 예를 제시 하였다.

공장마다 실정이 다르고 관심사항이 다를 수 있으나 Relational 데이터베이스의 확장성을 이용하여 필요한 Relation을 더 추가하고 이에 맞춰 모듈을 개발하면 보다 유용한 소프트웨어가 될 것이다. 또한 전문 공구관리 시스템, CAPP(Computer Aided Process Planning) 시스템과 데이터베이스 공유가 이루어질 경우 CIM의 실현에 한 걸음 더 다가갈 수 있을 것이다.

참고문헌

- (1) George U Hubbard, 1979, "Computer-assisted logical database design", CAD, Vol. 11, No. 3, pp.169~179.
- (2) Mustapha Koriba, 1983, "Database Systems: their applications to CAD software design", CAD, Vol. 15, No. 5, pp.271~276.
- (3) Katsundo Hitomi, 1990, "Manufacturing systems engineering: the concepts, its context and the state of the art", Int. J. Computer Integrated Manufacturing, Vol. 3, No.5, pp.275~288.
- (4) F. Roy Piciacchia, 1989, "Plant Operations Control: A Vital Necessity for Executing the JIT Schedule in a CIM Plant", AUTOFACT, Oct 30, pp.22-49~22-67.
- (5) 강무진, 윤용일, 김영기, 도성희, 이경희, 1991, "컴퓨터를 이용한 통합생산 자동화 기술: 플랜트 통제시스템 개발", 한국과학기술연구원.
- (6) "표준 몰드베이스, 급형부품", 起信精密機械株式會社, 1990.