

효율적인 BOM 사용을 위한 부품간의 상관관계 구현.

권영훈*(호서대 대학원 기계공학과), 김성근**(호서대 기계공학과)

Correlation embodiment between parts for efficient BOM use

YoungHun Kwon*(Hoseo University), SeongKun Kim**(Hoseo University)

ABSTRACT

A study was conducted to improve validity in new BOM structure through using the similitude of existing BOM in the PDM. The system is developed in Web-Based PDM environments using servlets and database. The system provides the correlation information between parts to inflect the reusability of existing BOM. User-defined correlation is used to inform the relations between design precedence and co-relations of parameters. Realistic application in injection mold production is constructed using this structure.

Key Words: Web-Based PDM, BOM, Product Structure,

1. 서론

기업 경쟁력을 높이기 위한 방법 중 하나로 제품을 얼마나 신속히 설계하여 생산할 수 있는냐는 것이다. 이를 위해선 설계자들이 제품에 대한 정보를 원활하고 효율적으로 공유 하는 것이 필수적이다. 이를 위한 방법 중 하나로 대두된 것이 PDM(Product Data Management) 시스템이다.⁽⁴⁾

PDM 시스템은 제품과 관련된 모든 주변 정보를 관리하고 제공하는 시스템으로, 제조 업체의 내부와 외부에서 제품과 관련된 작업자가 병렬적으로 작업을 진행시킬 수 있도록 지원하는 정보 관리 시스템이다. PDM은 이미 상용화 되어 많은 기업에서 사용되고 있고 차세대 PDM을 위한 연구도 활발히 진행되고 있다.⁽⁵⁾

PDM의 대표적인 기능들은 제품구성관리, 전자금고 및 문서관리, 작업흐름과 프로세스관리, 분류 및 코딩 시스템, 제품 개발 일정 관리 등이 있다.

이중 중요한 기능 중 하나는 제품구성관리(BOM)이다. BOM은 부품간의 상하 조립관계를 나타내는 것으로서 상위부품, 하위부품, 소요 수량 등의 정보를 가지고 있다. BOM은 어떤 관점에서 제품을 분해하느냐에 따라 Engineering BOM,

Manufacturing BOM, Planning BOM 등 여러 관점이 있을 수 있다. 일반적으로 PDM 시스템에서는 주된 사용자가 설계자이기 때문에 주로 Engineering BOM의 개념이 사용되고있다. 설계자들은 새로운 제품을 설계할 때 BOM을 통하여 기존정보를 활용하게 된다.

이 과정에서 BOM내에서 설계변경이 자주 일어나게 되는데 이때 설계자들이 이와 관련된 모든 BOM 정보를 찾아 전부 수정해야 하는 일이 발생한다. 설계자들이 효율적으로 완벽하게 수정하지 못하면 또 다른 설계 변경 요인이 발생하여 비용과 시간의 손실이 따르게 된다.

본 논문에서는 시간이나 장소에 구애 받지 않고 유지보수, 운용, 사용이 편리하게 하기 위하여 서버-클라이언트 모두를 웹 기반으로 하는 PDM 시스템을 구현하였다. 기존 BOM의 효율적인 활용과 구성을 위한 웹 PDM과의 연결을 구현하였다.

또한, PDM내에 BOM의 한 부품이 변경되었을 때에 관련된 부품의 수정을 쉽게 하기 위해서 BOM 정보 내에 있는 도면들의 상관 관계를 미리 정하여 한 부품의 수정이 발생하였을 때 상관관계에 있는 다른 부품들의 설계 담당자에게 통보 되도록 설계하였다.

2. PDM 시스템 아키텍처

현재 PDM 시스템 설계에는 여러 가지 요구 사항이 있는 발생하는데, 본 논문에서는 다음과 같은 요구사항을 중심으로 구현하였다.

2.1 요구사항

1) 과거 정보 활용

도면이나 BOM 정보를 효율적으로 재사용 하기 위해서는 과거 정보를 일괄적으로 복사하는 정도의 기능뿐 아니라 새로 적용 하려는 프로젝트에 맞게 정보가 변경 되어야 한다.

또한 한 부품에 대해 수정이 발생했을 때에 다른 부품의 수정이 용이 하도록 설계하여야 한다.

2) 플랫폼 독립성

PDM 시스템은 여러 환경에서 운용되기 때문에 플랫폼에 독립적이어야 한다. 여러 하드웨어 및 운영 체제 하에 설치되어 실행될 수 있어야 한다.

3) 위치 투명성

지리적으로 분산되어 있거나 근무지역이 자주 바뀌는 경우에 위치 투명성을 지원하기 위해서 클라이언트 프로그램을 따로 설치 하지 않아도 사용할 수 있어야 하는 것이 가능해야 한다.

4) 효율적인 작업 통보

PDM 시스템에서의 효율적인 작업이 되기 위해선 적절한 방법을 통해 정보를 통보 받을 수 있어야 한다. 특히 작업의 변경사항 등이 발생 하였을 경우는 신속한 통보가 이루어 져야 한다. 설계에 자주 참여 하지않은 구성원들도 효과적으로 진행 상황이나 변동상황을 통보 받을 수 있어야 한다.

5) 공동작업 지원

일반적으로 설계 업무에는 많은 담당자들이 공동 참여 하므로 서로간의 정보공유와 유기적인 상호관계를 할 수 있는 환경이 필요하다.

2.2 시스템 아키텍처

본 연구에서 개발한 웹 기반 PDM 시스템의 전체 아키텍처는 Fig.1 과 같다. 그림에서 보는 바와 같이 시스템은 PDM 엔진, BOM 관리, 클라이언트, 데이터베이스로 구성 되어있다. PDM 시스템은 PDM의 기본기능인 도면관리, 사용자관리, 프로젝트관리, 등을 구현하였으며 각 정보는 웹 서버를 통하여 DB 에 저장되고 필요에 따라 클라이언트를 통하여 사용자에게 전달된다. 시스템은 자바 서블릿과 애플릿 이용하여 웹 기반으로 구현 하였는데 이는 각 사용자의 위치 투명성과 독립적인 플랫폼 하에서 운용될 수 있도록 하기 위해서다.

클라이언트는 사용자가 직접 시스템을 사용하는

인터페이스다. 시스템에 접속할 때 권한을 주어 관리자 와 일반 사용자로 분류했다. 모든 GUI 가 자바 애플릿으로 구현 되었기 때문에 사용자는 익스플로러나, 넷스케이프와 같은 웹 브라우저를 통하여 다운로드 받은 후에 사용할 수 있도록 하였다. 이렇게 구현함으로써 PDM 의 요구 사항인 위치 투명성을 충족할 수 있다.

클라이언트와 서버사이의 통신용으로 자바 서블릿을 사용했는데, 자바 서블릿은 클라이언트가 브라우저로 접속을 하고 SQL 명령을 서버에 전달하는데, 그 전달자 역할을 하는 것이 서블릿이다. 이것은 기존의 방법보다 빠르고, 또한 서버와의 전달 역할까지 해주므로 응용 프로그램을 디자인 할 때 도 효율적이다. 그리고 자바는 운영체제에 구애를 받지 않으므로 서블릿도 운영체제에 구애를 받지 않아서 사용자 입장에서 매우 편하다.

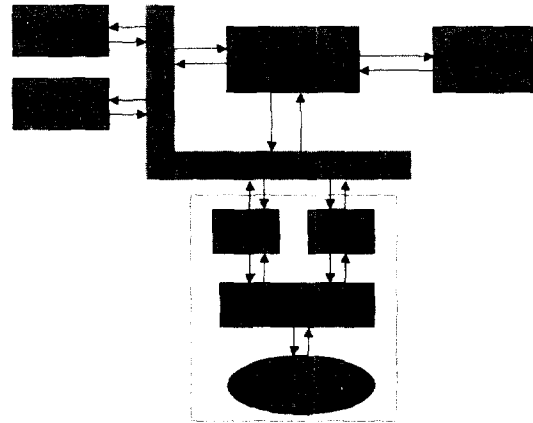


Fig. 1 Overall system architecture.

3. BOM 의 상관관계

3.1 상관관계 설정

상관관계 설정은 CAD 시스템에서의 설계 변수로서 구성해야 한다. 하지만 본 논문에서는 중소기업형 CAD 시스템에서 구성된 기존 BOM 을 효율적으로 활용하고 구성하기 위한 상관관계 설정을 하였다.

PDM 내에 중요한 기능 중 하나인 과거 BOM 정보의 활용은 대부분의 PDM 시스템에서 지원 하고있다. 하지만, 단지 과거 정보의 일괄 복사를 통한 사용에 그치고 있고 현재 프로젝트에 맞게 부품 코드나 담당자 등을 맞게 수정해야 하는 번거로운 일이 발생한다. 또한, 하나의 부품에 대해 수정을 가하였을 경우 다른 부품의 수정이 필요하게 되는

데 이 또한 작업자가 일일이 찾아서 수정해야 하는 방식으로 구성되어 있다.

일반적으로 BOM의 구성은 조립 체를 설계하는 방식인 하향 지향적 설계(Top-down approach)의 개념을 가지고 구성되어 있다. 설계자들이 각자 담당 한 부품들을 먼저 설계한 후에, 이들을 최상위의 조립 체로 올라 가면서 설계하는 방식 상향 지향적(Bottom-up approach) 설계방식은 많은 설계자가 동시에 유기적으로 설계를 진행 해야 할 경우에는 자신이 설계하거나 변경한 부품이 다른 부품과의 상관 관계에 있다는 것을 기대 하기는 어렵다. 반면에 하향 지향적인 개념은 먼저 BOM 으로부터 전체 부품의 조립구조를 정의해 두고, 이에 따라 진행해 나가는 것이다. 이는 과거 BOM 정보의 사용과도 유사한 개념이다.

제품구성을 표현하는 방법으로 Modular BOM, family BOM, matrix BOM, add/delete BOM, 등과 피쳐(feature)와 옵션을 고려한 BOM 등이 있다. Modular BOM 은 옵션과 공통 부품들로 구성된다. 옵션으로 관리하므로 방대한 양의 BOM 데이터가 감소한다. Family BOM 은 modular BOM 과 유사하나 제품들을 군으로 구성하는 것이 다르다. Matrix BOM 은 부품의 수량을 제품별로 테이블의 형태로 나열하는 것이고, add/delete BOM 은 기준 BOM 에 없는 부품은 더하고 불필요한 부품은 빼고 하는 방법으로 저장 효율을 높인 것이다.⁽²⁾ 제품의 일반적인 BOM 구조는 Fig. 2 와 같이 부품의 상하 관계와 하위부품의 수량으로 표현된다.

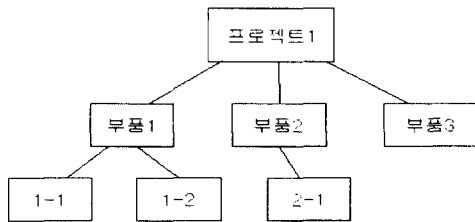


Fig. 2 A typical BOM structure

그림과 같은 BOM 구조는 전형적인 BOM 의 구조이다. 간단한 제품은 BOM 구조 또한 간략하게 표현 되어 하나의 수정이 발생하였을 때 다른 부품과의 간섭관계를 쉽게 확인하고 변경할 수 있다. 그러나 복잡한 제품은 최하위 부품과 최상위 부품의 단계가 많아지고 부품 수 또한 늘어 나기 때문에 서로의 간섭관계를 확인하고 변경하기란 쉬운

작업이 아니다.

그래서 Table1 과 같이 부품간의 상관관계를 미리 정의해서 BOM 에 적용시켜야 한다. 테이블 1 과 같이 BOM 정보에 상관관계가 설정되어 있으면 새로운 프로젝트에 기존 BOM 을 적용시킨 후에 부품 A100 의 설계 변경이 일어 나게 되면 상관관계의 설정에 따라 부품 A120 ,A140 설계 담당자에게 통보가 된다. 이는 BOM 을 생성 하였을 경우에도 기존 정보를 그대로 가져가며 기본 부품이 설계 변경되어 Rev. 이 변경되었을 경우와 상관부품의 Rev. 이 변경 되었을 경우도 그대로 적용되도록 구현하였다.

| 부품코드 | Rev. | 상관부품코드 | Rev. |
|------|------|------------|------|
| A100 | a | A120, A140 | a, b |
| A110 | a | A130, A150 | b, a |
| A120 | b | A100 | a |
| A130 | a | A110, A150 | a, a |
| A140 | a | A100 | b, b |
| A150 | a | A110, A130 | c, a |

Table 1 embodiment between parts

이런 상관 관계의 구축을 위해선 미리 조립체 전체에 대한 제품 구성도가 필요하게 된다. 그래서, 하향 지향적 설계 방식의 개념하에 BOM 을 구축하여야 한다. 먼저 상관관계에 따라서 구축되어 있는 BOM 을 사용 할 때 기존 상관관계를 그대로 적용 시킴 으로서 각 부품들의 수정 시에 향상된 기능을 보여 주게 된다. 효과적인 작업 통보 기능을 충족시키기 위해 SMTP(Simple Mail Transfer Protocol)을 사용하여 설계변경 되었을 경우 담당자의 E-mail 뿐만 아니라 PDA, 핸드폰, 또는 MSN 메시지로 연락 가능하도록 하였다.

3.2 BOM 정보의 활용

제품들 중에서 완성품으로 제조 되는 최상위 부품을 일반적으로 모델이라고 부르고, 각 모델에는 사용자의 요구에 따라 변형품이 존재하게 된다. 이러한 변형품이 많아 지게 되면 변형품에 비례해서 BOM 정보가 늘어 나게 된다. 또한 중복 되는 데이터도 많아지고, 한 부품을 수정할 경우 각 제품 구성별로 그 부품을 찾아 수정해야 되는 불편함도 있다. 이런 문제점을 해결 하기 위해서 하나의 모델에 여러 가지 변형품을 표현 할 수 있는 옵션 BOM 을 구성해야 한다. 옵션 BOM 을 구현 하기 위해서는 여러 옵션들을 표현하기 위한 가상 피쳐가 도입 되어야 하는데 예를 들면 자동차의 엔진은 1.5 리터, 2.0 리터등의 옵션을 가진 가상 피쳐가 되

는 것이다. 기어박스 와 같은 경우는 실제 부품이 고 그 밑에 트랜스미션의 경우는 4 단, 5 단의 옵션을 가진 가상 피쳐가 되는 것이다.

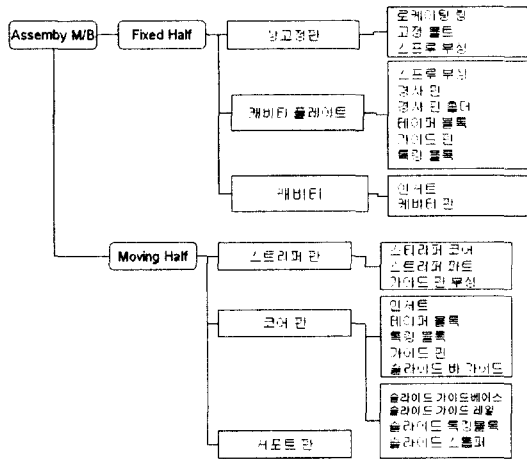


Fig. 3 Option BOM example

Fig.3 은 몰드 베이스의 한 모델의 제품 구조이다. 이 제품은 상고정판, 캐비티 플레이트 캐비티 등으로 구성된다.

Fixed Half의 경우는 가상 피쳐가 되는 것이고 그 밑에 상고정판, 캐비티 플레이트와 같은 경우는 옵션이 되는 것이다. 이러한 형태로 BOM 을 구축하게 되면 옵션을 이용하기 때문에 BOM 내의 중복되는 데이터도 줄 일수 있을 뿐만 아니라 선택적으로 제품을 구성할 수 있기 때문에 효율적인 방법이 된다.

3.3 시스템 구현

Fig.4 는 본 시스템의 주요 화면들이다. 그림(a)는 BOM 정보를 관리하는 화면이다. 오른쪽 옵션은 BOM 상관계와 정보를 입력하는 부분을 보여주고 있다. 하단 부분은 BOM 의 상관 관계와 정보 등을 표시해 주는 부분이다. (b)는 각 프로젝트의 도면 관리에 관한 모든 정보를 관리하는 부분이다.

프로젝트별로 도면이 정리되어 있으며 각 도면에 대한 리비전과 담당자 등의 정보를 확인할 수 있다 (a)와 (b)의 연결을 통해 모든 BOM 정보와 도면관리를 할 수 있다. 도면 관리 메뉴에서 기존 도면의 수정이 발생하여 리비전이 발생하면 BOM 정보의 도면 상관계에 대한 설정도 자동으로 바뀌게 된다.

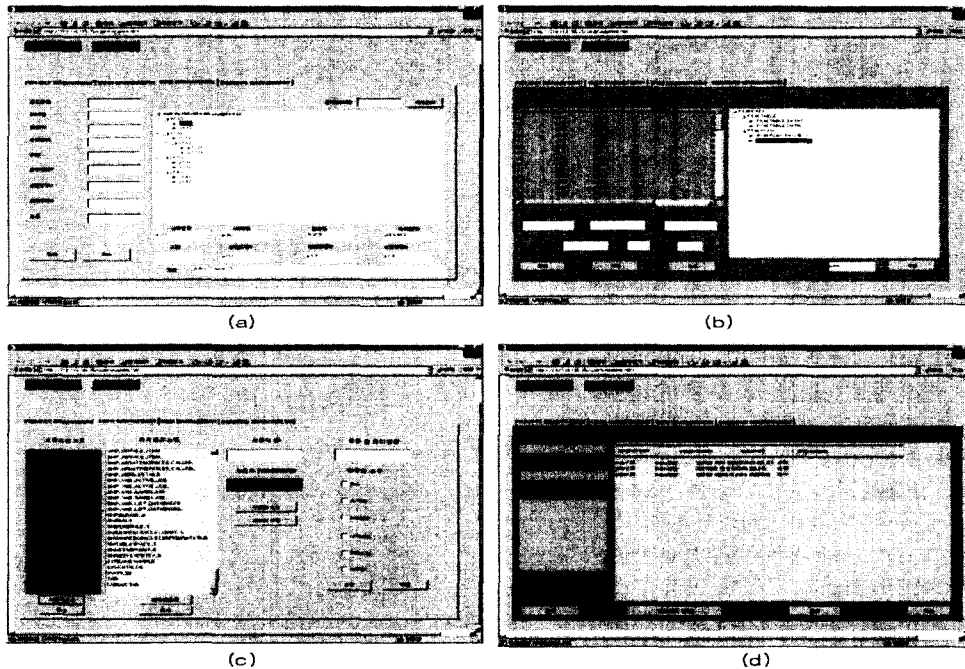


Fig. 4 System embodiment screen

(c)는 관리자용 메뉴로써 사용자 관리와 Data 테이블의 권한 등을 관리하는 메뉴이다. 각 사용자마다 정보에 대한 접근 제어를 할 수 있다. (d)는 프로젝트관리 하는 화면으로써 프로젝트 상태와 진행 상태 등이 나타나고 새로운 프로젝트를 구성할 때 BOM 정보와 연결 된다.

Fig.5는 BOM을 효율적으로 사용하기 위하여 가상 피쳐와 옵션을 사용한 BOM을 웹 PDM 시스템과 연동하여 구성한 결과와 BOM의 주요 정보를 나타내는 화면이다.

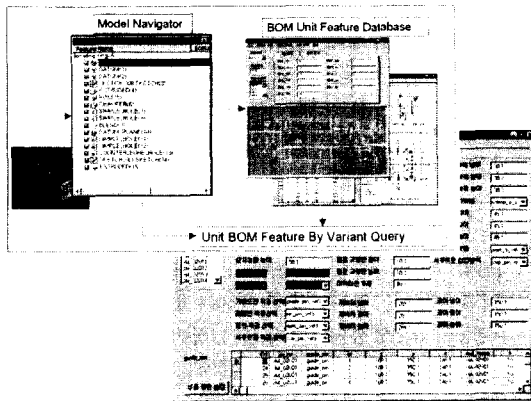


Fig. 5 Web PDM linked BOM information summary window

4. 결론

과거 정보의 효율적인 사용과 위치 투명성, 플랫폼 독립성 등을 고려한 웹 PDM 시스템의 설계와 개발에 관하여 설명하였고 PDM 시스템 내에 BOM의 상관관계를 고려하여 수정 부품의 수정이 발생하였을 때 효율적인 수정이 이루어 지도록 개발하였다. 가상 특징형상과 옵션을 사용해 구성된 BOM의 활용성과 효율성을 높였다. 이로 인해 시스템의 확장과 관리자의 시스템 관리가 용이해졌다. 또한 효과적인 과거정보 재활용과 부품 수정에 대한 신속한 대처로 효율적인 개발이 가능해졌다.

향후 연구되어야 할 과제는 PDM의 워크플로우 기능을 포함한 모든 PDM 시스템의 구축과 사용이 웹 기반으로 개발되어야 할 것이며, BOM내에 부품의 상관 관계 설정 또한 사용자의 수작업을 통한 입력 방식을 탈피하여 각 제품의 구성에 따라 자동으로 구축되는 지능화 시스템으로 개발되어야 할 것이다.

참고문헌

1. Zhu Yunlong, Li Hongxin, Xue Jinsong, Wang Hongtaot: "The Design of Cooperative Workflow Management Model Based on Agent," 1999.
2. 김선호, 정병용, 주경준, 정석찬 "제품구조 및 구성을 위한 옵션조합관리 기능 개발" 한국 CAD/CAM 학회 논문집, 제 5 권, 제 3 호, pp.224-231, 2000.9 월
3. 정용호 "부품수가 많은 조립 체를 설계를 위한 동시공학의 구현," 한국 CAD/CAM 학회 논문집, 제 2 권, 제 2 호, pp. 93-102, 1997.6 월
4. 김영호, 강석호, 이수홍, 유상봉 "분산, 개방, 능형 제품정보관리시스템" 한국 CAD/CAM 학회 논문집, 제 4 권, 제 3 호, pp. 210-223, 1999.
5. 이수홍, 서범석 "자동차산업 설계정보기술 PDM" 자동차공학회지 1999.6 월
6. E.A.I Matrix One "eMatrix Navigator Guide" 1999