

설계 DATA의 통합을 위한 EDB 시스템의 구조에 관한 고찰

김철호 · 임종균 · 정운영 · 김우승*

A Study on the EDB System Structure for Integration of Design Data

C.H.Kim · C.K.Lim · W.Y.Chung · W.S.Kim

(요 약)

현재까지의 EDB 시스템이 Data를 중심으로한 관리나 프로세스 관리 같은 것들을 프로그램으로 Hard Coding되어 있었던 관계로 인하여 단위 업무의 성력화 또는 순수 데이터 관리 측면에서는 효율성을 가졌으나, 기업의 무한경쟁시대에 생존 수단으로서 행해지고 있는 BPR(Business Process Restructure) 또는 CE(Concurrent Engineering)에 대응 할 수 있는 유연성이 결여되어 있었다. 따라서 프로세스나 데이터에 관련된 규칙 또는 데이터가 있는 위치등과 같은 프로세스 및 데이터의 속성에 대한 정보를 Meta Database에 저장하여 보다 유연성이 있고 일관성 있게 데이터를 취급해 줄 필요성이 대두 되었다. 이러한 Meta Database를 사용자 시각에서 바라보면 프로세스가 변화 된다하더라도 일관된 설계 작업을 할 수 있을 것이다. 즉, 사용자는 자신이 사용하는 EDB의 단위 Application을 사용하면서 새로 정의된 프로세스에 따라 자신의 데이터를 Release 하고 지방변경, 삭제, 의뢰등을 처리 할 수 있을 것이다. 따라서 현재의 EDB시스템들의 개발이 전체의 개발 프로세스 중에서 일부만을 부분부분으로 개발하고 통합 할 뿐 실제 일관된 데이터의 유지나 공유가 매우 힘들었던것을 부분적인 형태로 시스템이 구축된다하더라도 Meta Database를 통하여 개발 프로세스 전반에 걸친 다양한 PART 시스템간의 데이터 공유를 가능하게 할 수 있는 EDB 시스템구조에 대해 설명 하고자 한다.

1. 시스템 구조를 위한 요구조건

EDB 시스템의 기능이 제품개발의 전반에 걸쳐 생성되는 데이터(도면문서, 부품, BOM 등) 및 정보(과제진행, 제품사양, Know-How 등)를 유기적으로 결합하여 데이터의 가치를 재 창출하고 관련된 부서에 정보를 전달하여야 한다고 볼 때 필수적으로 갖고 있는 요소를 다음과 같이 규정 할 수 있다.

- ① 혼자 사용하지 않고 복수의 인원이 사용한다.
- ② 결재나 배포 또는 변경과 같은 업무절차를 포함하고 있다.

③ 사용목적(예를들면, 도면작성 등)에 맞는 기능을 작동시켜야 한다.

④ 실제의 데이터(예를 들면, Index나 File 등)를 갖고 있어야 한다.

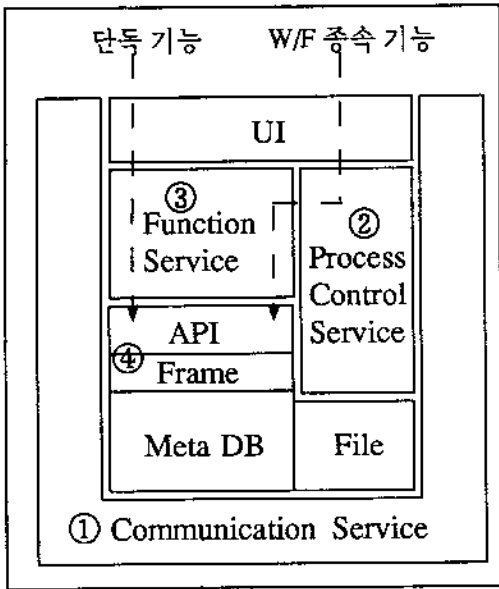
이상과 같은 조건을 도식화하면 <그림 1>과 같이 표현 할 수 있다.

2. Communication Service 기능

Network 상에서 흐르는 데이터의 원활한 흐름을 제어 할 뿐만아니라 하드웨어의 투명성을 가지고 데이

* 삼성전자(주) E-CIM 센터

1) Drag & Drop이라는 것은 마우스를 이용해서 원하는 객체를 복사 또는 원하는 위치로 이동시키기 위한 방법으로 객체(Icon)를 마우스로 선택해서 계속 마우스버튼을 누른상태에서 원하는 장소로 객체가 이동/복사될 수 있도록 하는 기능을 제공한다.



〈그림 1〉 통합 EDB시스템의 기본구조

타를 상호 교환 할 수 있도록 하여야 한다. 또한 가장 저수준에서의 데이터교환을 하기 때문에 높은 신뢰성과 안정성을 가져야 할 뿐 아니라 통합 EDB시스템이 아닌 환경에서의 요구사항을 처리 할 수 있는 Interpreter를 제공하여야 한다.

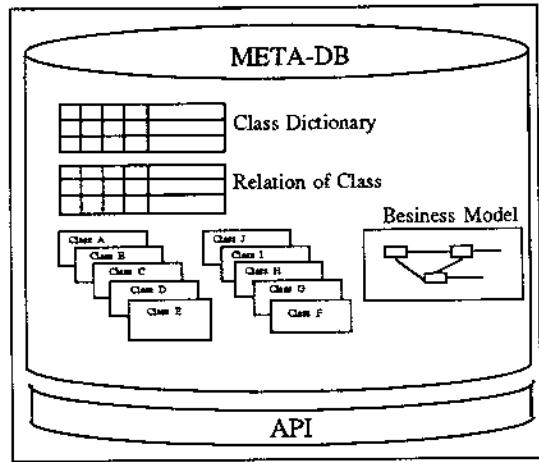
3. Data Distributed Service 기능

시스템에서 사용되는 모든 Object(이 경우 Object는 Data + Function이라고 볼 수 있음)와 그 Object에 대한 정보를 저장하고 관리하여야 한다. 따라서 EDB 시스템에서 공유하고자하는 데이터에 대한 정보와 그 처리 방법에 대한 정의가 존재하여야 하며, 기업의 기본적인 프로세스 정의가 되어 있어야 한다.

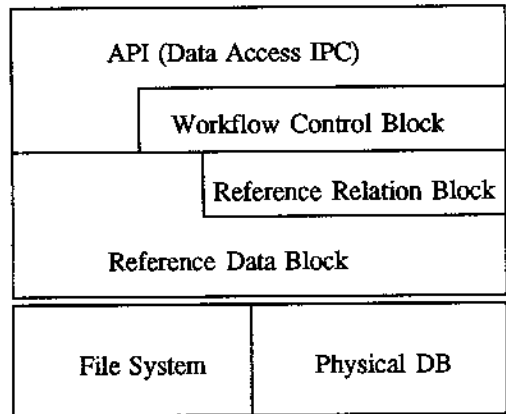
이 영역을 Meta Database라고 하면 구성 요소는 〈그림 2〉와 같이 볼 수 있으며 이에 대한 구조는 〈그림 3〉의 형태로 정리 할 수 있다.

여기에서 각각의 구조상의 역할은 아래와 같이 정의할 수 있다.

- ① Physical Data: File System 및 Relation DBMS(예: Oracle)
- ② Reference Data Block: 각 시스템에서 정의한



〈그림 2〉 Meta DB 구성요소

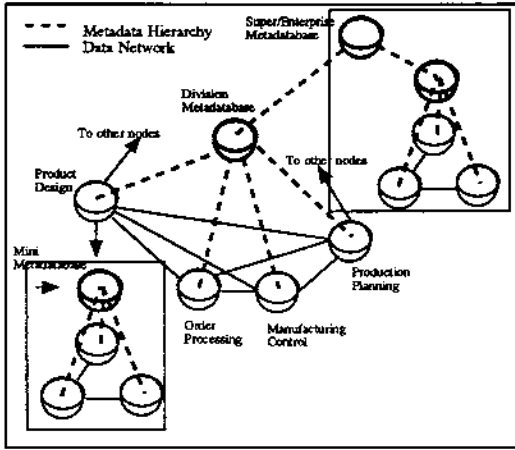


〈그림 3〉 Meta DB 구조

Physical Data(예: 도면 데이터)를 찾아 가기위한 기본 참조 정보로서 이를 통하여 각 시스템에서 찾고자 하는 실 데이터를 제공한다.(예: 문서 코드, 화일명 등)

③ Reference Relation Block: 여러 시스템간의 연계성을 주기위한 Link 정보를 갖는 부분으로 이것을 통하여 어떤 작업중에서 관련된 다른 시스템의 정보를 참조할 수 있다. 즉, 중심 데이터의 Primary Key의 집합으로서 이를 통하여 각 시스템이 연관 데이터를 직접 Access할 수 있다.

④ Workflow Control Block: 단위 TASK를 조합하여 하나의 업무 프로세스로 구성하는 Block으로 이



〈그림 4〉 사용상에서의 MDB 계층 구조

Block을 통해 프로세스의 변화에 능동적으로 대처하는 시스템 아키텍처를 구성한다.

⑤ API: Meta에 있는 참조 정보를 Get/Set등을 할 수 있는 표준화된 Function group을 말한다.

반면에, 구현상의 모습은 〈그림 4〉와 같은 형태로 볼 수 있으며, 필수적으로 갖추어야 할 기능을 요약하면 아래와 같다.

① Scalability: 설계정보의 통합 환경이 효율적으로 유지되어 지기 위해 크기와 구조를 변형 할 수 있으면서 타 Meta DB를 연결 할 수 있는 구조.

② Adaptability: 존재 하거나 또는 상용 Application이 보유하고 있는 DB내용을 Release시켜 Meta DB의 통제하에 있도록 하여야 하며, 이중 시스템과 현존 응용 프로그램들을 지원 할 수 있어야 함.

③ 시스템 전역적 동조화(Synchronization)나 연속화(Serialization) 없이도 병렬(Parallelism) 운영이 가능하여야 함.

④ Local 시스템들의 자치권을 부여할 수 있어야 함.

4. Process Control Service 기능

개발 업무 프로세스의 정형화 및 조직의 역할에 대한 기능 정의가 이루어져야 함. 이를 위한 단위구성 Block은 TASK로 설정하며, 이들 TASK는 RRB (Relation Reference Block)의 생성 여부에 따라 세가

지 형태로 구성 된다고 볼 수 있다.

① RRB를 사용한다: 이 경우 Work Flow는 RRB를 공유하는 TASK들 간의 관계성을 가진다.

② RRB를 사용하지 않지만 Work Flow를 정의한다: 이 경우 Work Flow는 독립된 TASK의 조합으로 구성된다.

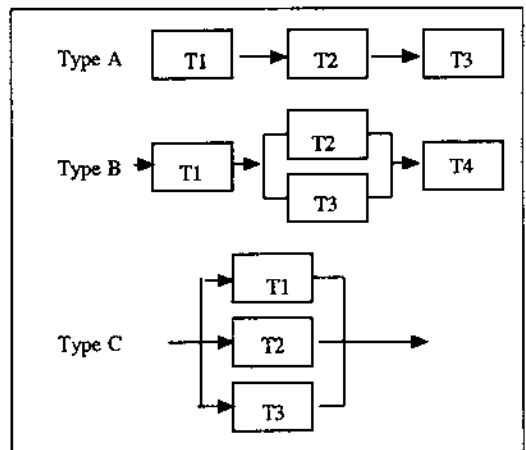
③ RRB를 사용하지 않지만 Work Flow도 사용하지 않는다: 이때 TASK는 Work Flow와 무관하게 작업을 수행하고자 하는 사용자에게 기능을 제공한다.

따라서 Work flow는 TASK들의 조합으로 구성되며, 구성되어지는 방법은 Work Flow를 구성하는 TASK들 간의 독립성에 의거하여 〈그림 5〉와 같은 여러가지 형태로 표현 될 수 있다.

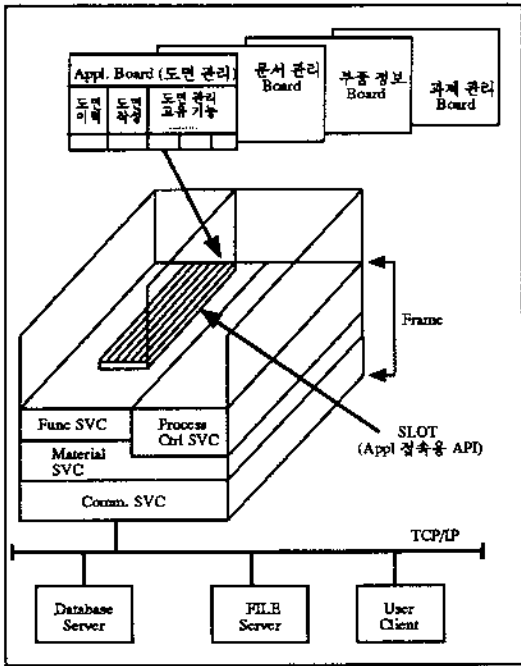
이러한 형태의 Work Flow 중 하나를 Icon의 형태로 GUI상에 제공 된다면 향후 프로세스가 변화된다 하더라도 그에 대응 할 수 있는 시스템의 유연성을 확보할 수 있을 것이라고 본다. 또한 프로세스의 노드에 응용프로그램을 설정 한다면 해당 프로세스에서 설정된 응용 프로그램이 수행 될 수도 있을 것으로 본다.

5. Function Service기능

실제의 데이터를 가공하는 응용 프로그램으로서 존재하며 2가지의 내부 제약 조건을 해결 하여야 한다.



〈그림 5〉 Work flow의 Type



<그림 6> SLOT의 개념도

① SLOT 개념의 구현 <그림 6>에서 보듯이 필요에 따라 Application을 선택하여 사용 할 수 있어야 데이터 변환에 있어 사용자는 이에 대한 것을 알지 못하여도 자신이 원하는 데이터를 다룰 수 있을 뿐만 아니라 업무 프로세스의 변경을 시스템에서 지원이 가능할 것이다.

② 상용 TOOL과의 Encapsulation의 구현: 설계 영역에서 사용되는 CAD와 같은 상용 TOOL을 통합 EDB내에서 구동시키기 위한 표준 API(Application Program Interface)를 설정하여야 한다.

6. 통합시스템의 사용 방법

6.1 시스템 사용의 기본 사상

전체적인 시스템의 일관성을 유지 할 수 있는 상태에서 사용되어 질 수 있도록 함으로써 나타나는 장점은

① 시스템에서는 도면관리, 문서관리, 부품정보 등을 하나의 작업 환경에서 할 수 있고 각각의 시스템

구분없이 작업할 수 있도록 하는 기본 사상을 충족시킬 수 있다.

② 각 사용자는 자신이 작업 할 수 있는 고유의 작업영역(Workspace)을 갖을 수 있다.

③ 모든 데이터는 반드시 자신이 작업 할 수 있는 작업영역으로 제공되어 진다. 즉, 단순히 데이터를 검색하는 경우에도 반드시 자신이 작업영역에 Reference를 가져다 준다.

④ 작성된 데이터는 다른 사람에게 보내기 위해서는 시스템에서 제공되는 Packet을 사용할 수 있다. 따라서 대부분의 결재, 배포, 합의 등과 같은 작업은 Packet을 조정함으로써 가능하다.

⑤ 시스템에서 제공되는 Folder를 이용하여 데이터를 묶어서 관리 할 수 있는 기능을 제공한다. 따라서 이 folder는 화일시스템의 디렉토리라 유사하다고 할 수 있다. 여기에서 가장 위에있는 folder는 Workspace라고 볼 수 있으며, 이것은 모든 작업의 기본 환경으로 사용된다.

6.2 시스템에서 사용하는 기본 객체(데이터)

사용자는 기본적으로 다음과 같은 5가지 기본 객체를 사용해서 모든 작업을 하게 된다. 뿐만아니라 향후 새로운 객체들을 손쉽게 시스템에 추가할 수 있는 환경을 시스템에서 제공한다.

☞ 문서(Document): 이 객체는 워드프로세스나 DTP로 작성한 문서 및 손으로 작성한 것을 광화일에 저장한 문서 등을 관리하기 위한 것이다. 또한 시스템내에 있지 않아도 그것에 대한 인덱스를 관리할 수 있도록 정의된 것이다.

☞ 도면(Draw): 이 객체는 CAD에서 발생하는 도면화일을 관리하기 위해서 정의된 것이다.

☞ 부품(Part): 이 객체는 부품 및 부품의 구성을 관리하기 위해서 정의된 것이다.

☞ Folder: 이 객체는 다른 객체를 담을 수 있는 container역활을 하는 것으로 많은 데이터를 보다 쉽게 관리할 수 있도록 정의한 것이다.

☞ Packet: 기본적인 기능은 Folder와 같으나 데이터를 전송하기 위한 기능을 가지고 있다. 따라서 사

용자는 반드시 Packet에 데이터를 담아서 다른 사람이나 부서에 전해야 한다.

6.3. 사용자 환경 및 사용방법

시스템에서 사용되는 각각의 화면과 그와 함께 사용되는 기능들을 살펴보면 다음과 같다.

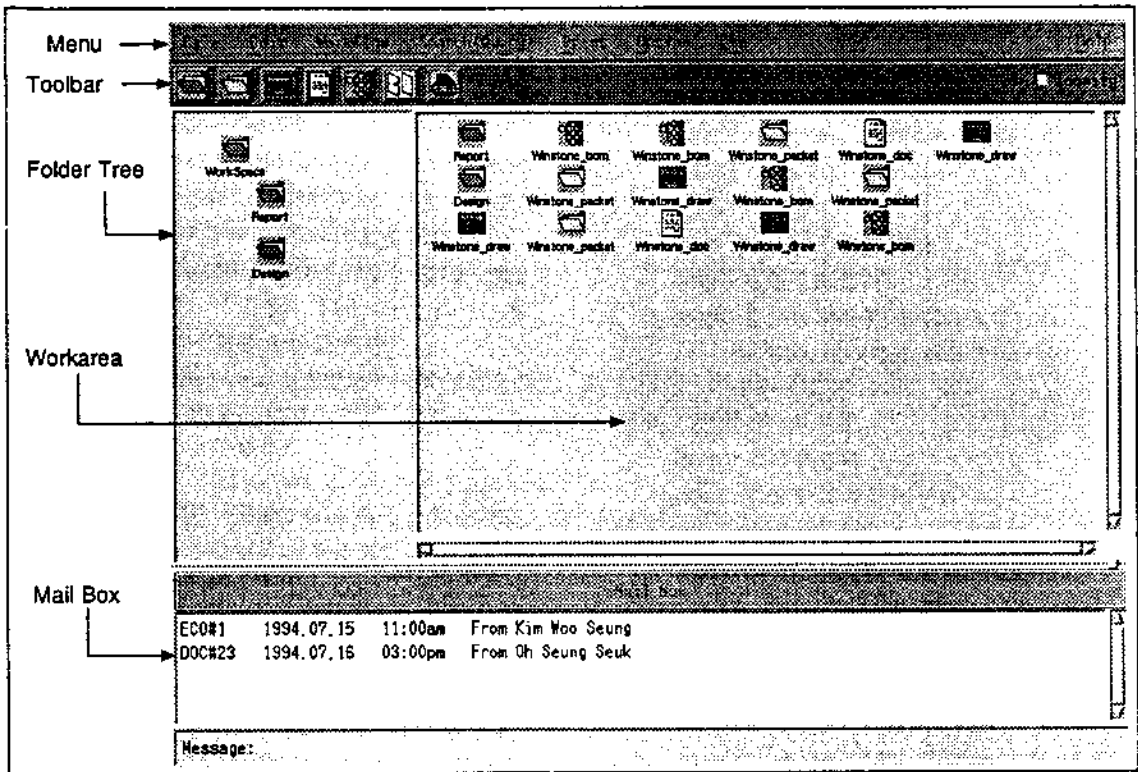
1) Workspace

Workspace는 사용자가 작업하는 기본적인 환경이다. 모든 작업의 시작은 workspace에서 이루어지게 된다.

즉, Login화면을 거친 후 사용자가 보게되는 첫번째 화면인 것이다. 이때 시스템은 현재 Login한 사용자가 전에 logout하였을 때의 상태를 그대로 유지하고 있다. 따라서 사용자는 자신이 작업하고 있던 상황을 그대로 유지하면서 연속성 있는 작업을 할 수 있

게 되는 것이다. Workspace의 화면은 다음의 <그림 7>과 같이 MS WINDOWS의 File Manager와 유사하게 작업을 할 수 있도록 되어 있다. 차이가 있다면 시스템에서 다루는 데이터가 단순한 화일이 아닌 Product Data라는 차이가 있을 뿐이다.

Workspace의 특징은 기존의 EDB시스템과는 달리 도면, 문서, 부품정보를 가능하면 일관된 방법으로 처리할 수 있는 환경을 제공한다는 것이다. 앞장에서도 언급한 5가지 기본데이터를 가지고 사용자가 보다 손쉽게 관련정보를 조회하고 처리할 수 있게 해준다. <그림 7>을 보면 알 수 있듯이 시스템에서 처리하는 데이터는 보다 직관적으로 사용할 수 있도록 Icon형태로 화면에 표시되며 이 Icon을 가지고 대부분의 작업을 할 수 있게 된다. 뿐만아니라 list의 형태로도 데이터를 볼 수도 있게 되어 있다. 사용자는 Toolbar의 Iconify버튼을 off하면 Icon형태의 데이터를 list를 볼



<그림 7> Workspace 화면

수 있게 되는 것이다. 물론 list 형태로 작업을 할지라도 icon을 가지고 작업하는 것과 같은 방법으로 작업이 가능하다.

화면의 구성

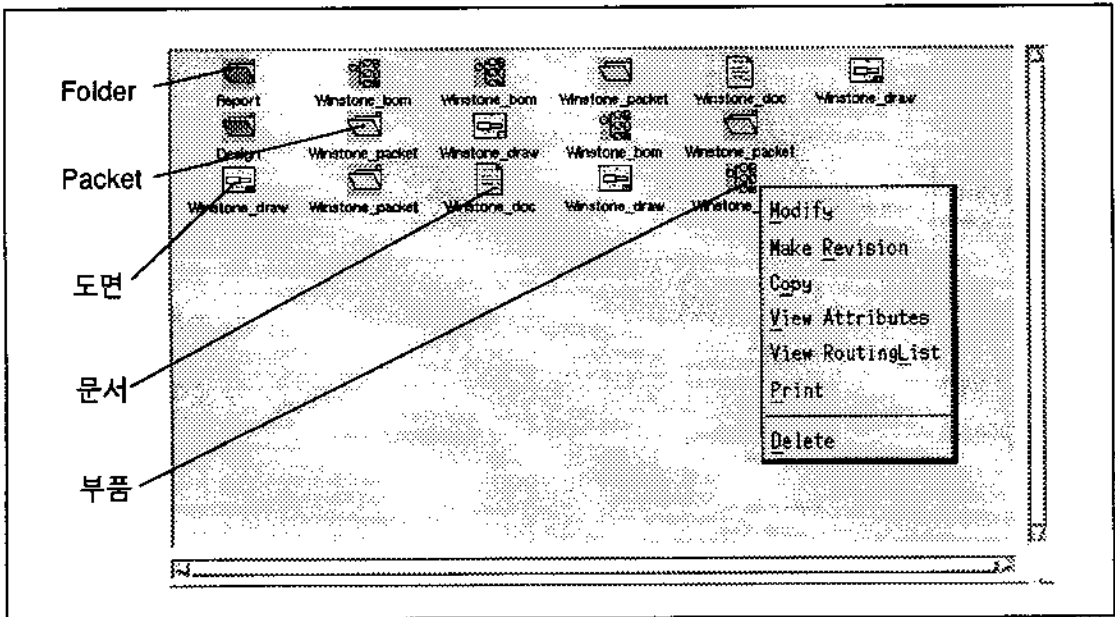
Workspace의 화면은 크게 6가지로 나뉘어져 있다.

- A) Menubar: 데이터를 처리하는 대부분의 명령들이 모여있다. 사용자는 Workspace에서 처리하고자 하는 객체를 선택한 후에 Menubar에 있는 적절한 명령을 선택해서 실행시킬 수가 있다.
- B) Toolbar: Toolbar는 자주 쓰이는 명령들을 Icon으로 표시해서 사용자가 쉽게 작업을 할 수 있도록 하였다. 주로 Menubar에 있는 기능등을 보다 빨리 사용할 수 있도록 해준다. 또한 사용자가 원하는 기능을 손쉽게 이 Toolbar에 수정/추가할 수 있다.
- C) Folder Tree: 현재 사용자가 Workspace에 만들어 놓은 Folder의 구성을 화면에 보여준다. 이 Folder은 앞에서도 설명을 했듯이 일종의 directory와 같은 것으로 계층적인 구조로 구성할 수 있게 되어 있다. 이 화면에는 Folder뿐만 아니라

Paket역시 일종의 Folder으로 간주해서 화면에 보여준다. 실제로 Paket이 Workspace에 있는 경우에는 Folder와 동일하게 인식되어 사용되고 다만 데이터를 전송을 할 때 이 Paket을 사용하게 되는 것이다.

- D) Workarea: <그림 8>은 사용자가 작업하는 데이터를 표시하는 화면이다. 각 Icon에는 데이터를 처리하는 Popup메뉴를 제공함으로써 보다 작업을 손쉽게 할 수 있도록 되어 있다.
- E) Mail Box: 이부분은 현재 사용자에게 전송되는 Mail들이 표시된다. 실제 시스템에서 Mail이라는 것은 Paket들이 전송된다고 생각하면 된다. 앞서 강조하였지만 사용자가 새로운 Mail(또는 Paket)을 받아서 작업을 할 경우에는 반드시 자신의 Workarea에 옮겨는 후에 작업을 해야만 한다.
- F) Message Line: 이부분에는 시스템이 동작을 하면서 현재의 작업 내용이나 에러를 표시해준다.

지금까지 Workspace의 화면구성과 기능에 대해서 간단히 설명을 하였다. 대부분의 데이터의 처리가



(그림 8) Workarea 화면

Workspace을 중심으로 이루어지기 때문에 사용자는 Workspace의 동작원리에 대해서 잘 이해할 필요가 있다.

도면이나 문서, 부품 등에 대한 구체적인 데이터의 입력이나 처리는 개별적인 화면을 사용하게 되지만 이러한 처리를 하기 위해서는 반드시 Workspace을 거쳐서 작업을 하도록 되어 있는 것이다.

2) Search Result 화면

사용자가 데이터를 검색한 결과를 표시해주기 위한 것으로 Workspace의 Mailbox와 마찬가지로 찾은 데이터는 반드시 Workarea안에 옮겨 놓은 후에 작업을 해야한다. 데이터를 Workarea에 옮길때 시스템은 데이터에 대한 사용자의 권한을 검사해서 비록 데이터(객체)가 Workarea에 있을지라도 단순히 검색작업만을 한 다든가 아니면 수정작업을 한 다든가 하는 것등을 사용자가 작업을 할 때 알려주게 된다.

6.4 Workspace의 사용방법

실제로 Workspace화면에서 사용자가 어떻게 작업을 하는가에 대해서 구체적으로 살펴보면 다음과 같다.

1) 기본작업방법

작업방법은 새로운 데이터를 생성하는 경우와 기존의 데이터를 가지고 작업하는 2가지 경우로 크게 나눌 수 있다.

A) 신규작업

- step 1: 새로운 객체를 생성
- step 2: 기초정보 입력
- step 3: 생성된 객체를 가지고 작업을 한다.(워드작업, 캐드작업, 부품구성 등)

B) 기존데이터의 이용

- step 1: 객체를 검색
- step 2: 필요한 객체를 자신의 Workspace에 이동
- step 3: 가져온 객체를 가지고 작업을 한다.
 - 데이터의 수정
 - Revision 작업(설계변경)
 - Copy를 해서 신규객체를 생성후 작업(신규작업과 유사)

이러한 작업을 하기 위한 명령들은 대부분 Workspace화면의 Menubar에 있는 Edit부분에 있다. 또한 검색된 객체를 자신의 Workspace로 가져오기 위해서는 Search Result에서 객체를 선택한 후에 Edit에 있는 [Show to Workspace]버튼을 실행시키든가 Drag & Drop이라는 방법을 쓰면 된다. 객체의 수정/출력/속성검색 등의 작업은 Menubar의 edit부분에서 명령을 수행할 수도 있고 Icon에서 마우스의 3번째 버튼을 누르면 같은 기능을 하는 Popup menu를 사용할 수가 있다.

2) Folder와 Packet의 이용방법

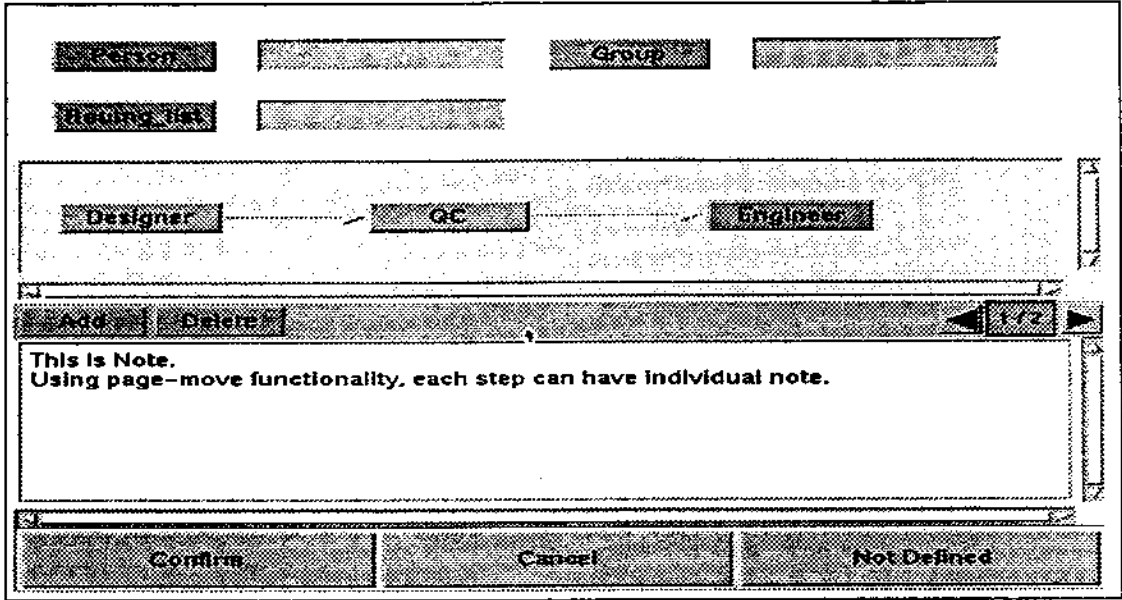
Folder, Packet은 기본적으로 다른 객체를 담을 수 있는 Container역할을 한다. 이 Folder을 잘 이용하면 Workspace에 있는 많은 객체들은 체계적으로 정리할 수 있다. 사용자는 Menu bar의 Edit메뉴의 Create을 이용해서 다른 객체와 마찬가지로 Folder나 Packet을 생성할 수 있다. 새로운 Folder나 Packet을 만들면 Workspace의 Folder tree부분에 생성된 Folder/Packet이 표시되고 동시에 Workarea부분에도 표시된다. 사용자는 표시된 Folder/Packet icon을 선택해서 자신이 작업하고자하는 Folder/Packet을 열 수 있다. 그안에서 사용자는 새로운 객체등을 생성하거나 원하는 객체를 이동/복사할 수 있게 된다. 이때 사용되는 방법은 주로 Drag & Drop방법이 사용된다.

또한 Packet은 모든 사용방법에 있어서 Folder와 동일하다. 하지만 Packet은 Folder가 가지고 있지 않은 속성이 있다. 그것은 다른 사람에게 데이터를 전송하기 위한 Route List나 보낼 대상에 대한 정보등이다. Packet의 이러한 기능을 이용해서 사용자는 원하는 사람에게 필요한 정보나 메세지등을 보낼 수 있다.

3) Workflow

본 시스템에서 Workflow라는 것은 단위업무에 있어서의 flow를 말한다. 즉, 문서나 도면의 결재, 배포, 합의 등과 같은 작업을 말한다. 따라서 업무프로세스를 총괄적으로 관리하고 처리하는 것은 Workspace가 아닌 Process Engine의 영역에서 제공해준다.

여기서는 주로 데이터를 다른 사람에게 또는 다른



〈그림 9〉 Workflow 화면

부서로 데이터를 보내거나, 정해진 결재선을 따라 데이터를 전송하는 방법에 대해서 설명하도록 하겠다. 이것은 데이터의 효율적인 관리와 더불어 SI시스템이 제공하는 가장 유용한 기능으로 각 부서나 개인의 데이터 흐름을 시스템이 관리하고 이를 바탕으로 데이터를 전송하고 결재하는 것을 보다 손쉽게 할 수 있도록 한다.〈그림 9 참조〉

4) 데이터의 검색방법

데이터의 검색을 위해서 사용자는 관련된 데이터키를 입력해서 검색을 수행한다. 특징적인 것은 기존의 EDB시스템이 도면이면 도면, 문서면 문서, 부품이면 부품으로 각각 검색한 후 작업을 해나갔지만 현재 시스템은 이러한 도면/문서/부품을 가능하면 일관된 방법으로 동시에 작업하도록 되어 있기 때문에 데이터의 검색 역시 도면/문서/부품을 동시에 검색할 수 있는 기능을 포함하고 있다. 또한 이르기 위해서 데이터를 분류하는 가장 중요한 Primary key로써 모델명을 사용한다. 이것은 어떠한 데이터든지 거의 대부분 어떠한 제품개발 과제와 개발모델에 종속된다고 보기 때문이다. 따라서 사용자는 데이터를 검색할 경우 문서/도면/부품의 각각에 대해서도 할 수 있고 동시에

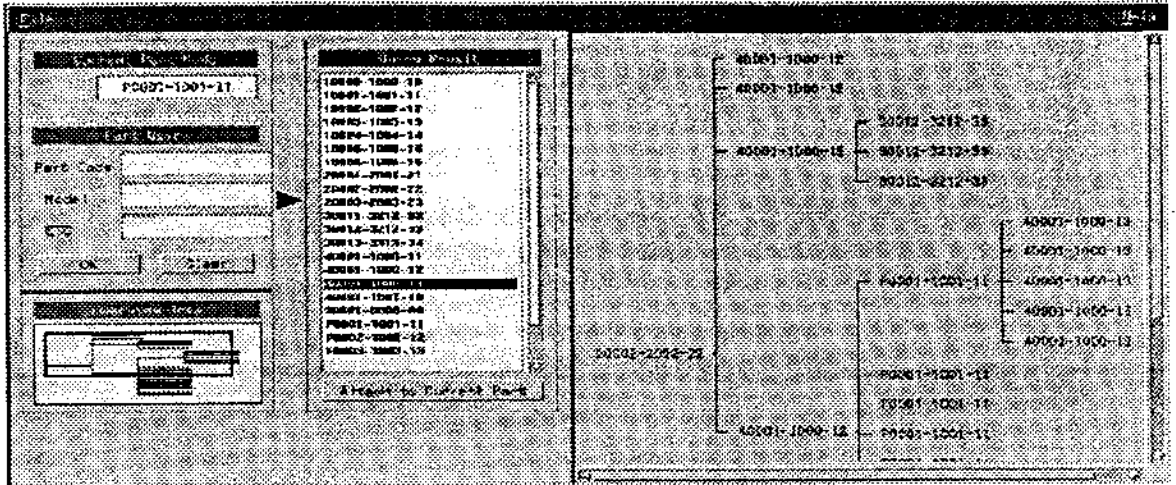
도 가능하도록 되어 있다.

5) PART의 구성(BOM)

〈그림 10〉은 하나의 Part를 생성하고 그 Part의 구조를 설정하기 위한 화면이다. 사용자는 간단한 Part에서부터 보다 복잡한 Assembly까지 이 Builder을 가지고 구성 할 수 있다. 사용자는 오른쪽의 Tree형태의 Part화면을 이용해서 편리하게 Subpart들을 삭제, 추가, 수정할 수 있다. 시스템에서는 이러한 Part들을 지우거나 수정할 수 없도록 한다. 아래의 Part Structure Builder는 CAD나 WP와 마찬가지로 별도의 Tool로써 사용될 수 있도록 되어 있다. 비록 Builder가 SI System과 연동되어서 사용되겠지만 BOM전담자는 이 Builder만을 가지고 작업할 수 있도록 되어있다.

7. 결론

이상에서 살펴본 바와 같이 기술 분야의 정보 시스템 형태(EDB)도 업무처리 기능과 특정 목적 기능이 Hard Coding된 상태에서 구성 요소의 분리와 동시 설계 방식에 맞춰시스템이 제공되는 상태로 변하고 있다고 볼 수 있다. 또한 특정 용도의 S/W Engine(예:



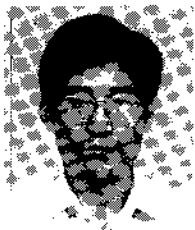
(그림 10) Part Structure Builder

Search Engine, 예측 Engine 등)

이 제공되어지고 상호 결합에 의한 업무의 효율을 더욱 향상시켜 나갈수 있을 것이라고 생각한다.

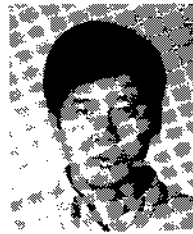
[참고문헌]

[1] Hsu,C and Rubenstein,A “Enterprize Information Management for Global Manufacturing”, RPI, 1992.
 [2] PDM BUYER’s GUIDE, Cimdata, 1992.
 [3] M.Schatter, R.Furegati, F.Jeger, H.Schneider, H. Streckesen, “The Business Object Management System”, IBM System Journal, 1994.
 [4] Thornton,A.May “Know Your Work-flow Tools”, BYTE, 1994. July to appear in CE94(Pittsburgh, PA, Aug 29-31, 94): “The Concurrent Engineering Toolkit”, GE Corporate Research & Development.



김철호

1981년 인하대 기계과 졸업
 1983년 삼성전자 입사 Engineering Database 연구 및 개발
 현 재 삼성전자 기술총괄 E-CIM 센터에서 SI관련 연구 및 개발업무총괄



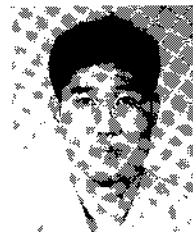
임종균

1986년 뉴욕대학 전산학과 졸업
 1988년 콜롬비아 대학원 전산과 석사과정 졸업
 1993년 삼성전자 입사 PDM 및 기술정보 문서처리 연구
 현 재 콜롬비아 대학원 전산과 박사과정 재학중



정운영

1984년 서울대 전산과 입학
 1988년 RPI 대학 전산과 졸업
 1991년 RPI 대학 대학원 전산과 석사과정 졸업
 1993년 삼성전자 입사 PDM의 Meta-DB 관련 연구
 현 재 RPI 대학원 전산과 박사과정 재학중



김우승

1990년 고려대학교 기계공학과 졸업
 1990년 삼성전자 입사 Engineering Database 관련 시스템 개발
 현 재 PDM 관련 연구 및 시스템 개발 담당