

능동형 객체지향적 데이터베이스 모델을 이용한 설계변경 개체 및 제약조건의 표현

도 남 철*

A Representation of Engineering Change Objects and Their Integrity Constraints Using an Active Object-Oriented Database Model

Nam Chul Do*

Abstract

This paper proposes a product data model that can express and enforce integrity constraints on product structure during engineering changes (ECs). The model adopts and extends an active object-oriented database model in order to integrate EC data and their integrity constraints. Tightly integrated with product structure, it will enable designers to maintain and exchange consistent EC data throughout the product life cycle. In order to properly support operations for ECs, the model provides the data, operations, and Event-Condition-Action rules for nested ECs and simultaneous EC applications to multiple options. In addition, the EC objects proposed in the model integrate the data and integrity constraints into a unified repository. This repository enables designers to access all EC data and integrity constraints through the product structure and relationships between EC objects. This paper also describes a *prototype product data management system based on the proposed model in order to demonstrate its effectiveness.*

Keywords : Engineering Changes, Product Data Model, Product Data Management, Active Object-Oriented Database, Event-Condition-Action Rules

1. 서론

제품개발에서 설계변경(Engineering changes : EC)은 많은 시간 및 자원을 필요로 하는 광범위한 정보처리를 야기시킨다. 또한 세계화 된 현대 시장에서는 설계변경 정보가 기업외부의 공급자와 고객과도 전자적으로 교환되어야 할 필요가 있다. 이러한 요구에 필요한 재작업과 낭비를 줄이기 위하여 설계변경 정보는 전산화 되고 교환시 항상 무결성(Integrity)이 유지되어야 한다.

제품자료 모델(Product data model)은 제품일 생주기 동안에 사용되는 응용 시스템(Application system)을 위한 통합된 자료구조, 관련연산, 그리고 무결성 제약조건(이하 제약조건)을 정의한다. 그러므로 효과적인 제품자료 모델은 설계변경 과정동안에 정보시스템에 일관되고 통합된 정보를 제공할 수 있도록 한다. 일반적으로 제품자료 모델은 데이터베이스 관리 시스템에 의하여 구현되며 사용자 정의 제약조건을 통합적으로 표현할 수 있는 능동형 데이터베이스 관리시스템(Active database management system)이 무결성을 포함한 자료모델 표현에 적합하다.

본 연구에서는 능동형 객체지향적 데이터베이스(Active object-oriented database) 모델을 이용하여 제품자료 모델과 이를 기반으로 한 설계변경 제약조건을 표현하는 제품 구조중심의 무결성 유지 방법을 제안한다. 이 연구는 설계변경에 적용되는 무결성을 위한 정형화된 제품자료 모델을 제안하고 이를 제품자료 관리시스템의 시제품을 이용하여 구현한다. 이를 위하여 설계변경에 관련된 제품구조(Product structure), 관련연산, 그리고 제약조건을 단일 자료 저장소에 통합시킬 수 있는 설계변경 개체(EC objects)를 제공한다.

본 논문은 다음과 같이 구성되어 있다. 2장에서는 능동형 데이터베이스가 지원하는 Event-Condition-Action (ECA) 규칙과 설계변경 관리를 위한 서로 다른 접근 방법을 알아본다. 3장에서는 일반적인 기계 조립 및 가공 산업에서 받아들여지는 제품구조에 관하여 서술한다. 또한 이 제품구조에 근거하여 설계변경 관리를 위한 기능적 요구사항을 확인한다. 4장에서는 이전 장에서 언급된 설계변경 요구를 만족시키기 위하여 능동형 객체지향 데이터베이스 모델에 근거한 제품자료 모델을 제안한다. 이 제품자료 모델은 요구되는 기능을 지원하기 위한 자료구조, 관련 연산 그리고 제약조건으로 구성된다. 5장에서는 예제와 이를 이용한 시제품에 관하여 언급한다. 6장에서는 추후 연구과제를 언급하며 결론을 도출한다.

2. 관련 연구

본 논문에서 제안한 제품정보 모델은 설계변경에 관련된 제약조건을 표현하고 강제하기 위하여 능동형 데이터베이스의 ECA 규칙의 문법을 사용한다[Dayal, 1988 ; McCarthy, 1989]. ECA 규칙은 제약조건이 검사되고 강제되어야 할 사건(Event)을 정의하고, 사건이 일어났을 때 평가할 논리적 표현인 조건(Condition)을 정의하고, 조건이 만족했을시 행해져야 할 행동(Action)을 정의한다. 객체지향적 데이터베이스에서 ECA 규칙을 객체모델과 통합함으로써 관계된 모든 자료와 연산 그리고 제약조건을 일관되게 표현하고 관리할 수 있다. 본 논문에서 제안된 제품자료 모델은 설계변경에 관계된 자료, 연산 그리고 제약조건을 통합하기 위하여 ECA 규칙이 지원되는 능동형 객체지향적 데이터베이스 모델을 사용한다.

ECA 규칙을 이용하여 공학제약조건을 표현

하려는 노력으로 Do와 Choi는 ISO STEP의 표현언어인 EXPRESS로 표현된 공학제약조건을 교환하기 위하여 ECA 규칙을 사용하였다[Do & Choi, 1996]. Choi와 Bae는 제품형상관리(Product configuration management)를 위하여 ECA 규칙을 사용하였다[Choi & Bae, 2001]. 하지만 위의 연구들은 제품구조와 통합된 설계변경에 관련된 연구결과를 제공하지는 못하였다.

설계변경관리에 관한 연구는 문서 중심, 프로세스 중심, 그리고 제품구조 중심의 접근으로 나눌 수 있다. 문서 중심의 접근방법은 설계변경을 전자 문서 관리 시스템을 통하여 관리한다[Baya, 1996; Baya & Leifer, 1994; Baya & Leifer, 1996; Brown, 1998; Gates *et al.*, 1996; Hardwick *et al.*, 1994; Hardwick *et al.*, 1995]. 이 방법은 일반적으로 설계변경과 제품구조를 분리해서 관리하게 된다. 이 분리는 설계변경 과정에서 가장 중요하고 많이 발생하는 제품구조의 분석과 검토를 통합된 환경에서 하지 못하도록 하며 또한 설계변경에 필요한 정보를 이질적이고 분산된 전산 환경에서 각각 검색하고 처리하도록 만든다. 그러므로 이 방법은 설계변경 활동에 긴 시간이 필요하고 많은 오류가 존재하도록 만든다.

프로세스 중심의 접근방법은 설계변경 관리를 위하여 업무 프로세스 개념을 도입한다[Brown, 1998; Casotto *et al.*, 1990; Shah *et al.*, 1996]. 이 방법도 역시 설계변경 자료를 제품구조와 분리 시킨다. 이는 문서 중심 접근방법과 유사한 문제를 야기시킨다. 또한 설계변경이 협동적인 활동이며 내용에 따라 프로세스가 변경되는 특성이 있으므로 이를 자세한 수준으로 정의하거나 표준화 하는것이 거의 불가능하다[Agilesoft, 1999]. 만일 자세한 수준의 정의나 표준화가 가능하다 하더라도 정의를 위한 많은 자료로 인하여 설계변경 자료의 무결성 유지가 어렵게 된다.

제품구조 중심의 접근방법은 설계변경 정보를 저장하고 관리하기 위하여 제품의 구조를 사용한다[Agilesoft, 1999; Enovia, 1999; Peng *et al.*, 1998]. 이 접근방법은 설계변경을 부품간의 조립관계의 변화로 관리한다. 이 접근방법에서 설계변경은 제품의 구조와 밀접하게 연동되어 있으므로 설계변경 기간동안 빠르게 변경되는 제품구조의 무결성을 유지할 수 있다. 이러한 효율성 때문에 본 연구에서는 제품구조중심의 접근방법을 채택하고 있다.

설계변경에 관련된 제품자료 모델에 관한 연구로는 Katz와 Chang[Katz & Chang, 1987], Katz[Katz, 1990], Hamer와 Lepoeter[Hamer & Lepoeter, 1996], Peng과 Trappey[Peng & Trappey, 1998] 그리고 ISO STEP을 이용한 PDM Schema[PDM Implementer Forum, 2002] 등을 들 수 있다. Katz의 Version model(1990)에서는 본 논문에서 제안하고 있는 동시 설계변경 제약조건을 해결하기 위하여 Group에 속한 개인의 변경을 최종적으로 하나의 변경으로 통합하는 과정인 Group check out/check in 방법을 제안하였다. 이 방법은 Group에 속한 모든 사용자의 설계변경을 하나로 조정하고 통합하는 과정이 필요하나 본 논문에서 제안한 방법은 설계자 개인이 설계변경을 Check out/Check in 할 수 있다. 또한 Version model[Katz & Chang, 1987; Katz, 1990]은 제품의 변경을 단순히 Version의 순서로 관리하므로 설계변경 단위에 대한 기준을 제시할 수 없다. 이에 비하여 본 연구에서는 설계변경 개체를 통하여 제품구조, 변경이력, 그리고 설계변경 개체 상호관계를 정의하여 효율적 변경정보 공유와 통합된 저장소로써 기능이 가능하다. 특히 본 연구에서는 설계변경 이력을 제품구조와 적용 연산의 순서로 정의하므로 보다 효과적으로 제품구조의 변화를

표현할 수 있다.

Hamer(1996)는 설계변경을 포함한 5가지 Dimension의 개념적 모델을 제안하고 설계상에서 다수의 Dimension이 복합적으로 상호 작용함을 예로 보여 주었다. 하지만 3가지 이상 Dimension이 복합 작용하는 구체적 모델을 보여 주지 못하고 있다. 그러나 본 논문에서 제안된 모델은 제품구조, 제품형상, 제품관점, 그리고 설계변경의 4가지 Dimension을 통합적으로 표현하고 있다. Peng *et al.*(1998)의 연구와 PDM Schema[PDM Implementer Forum, 2002]는 ISO STEP 표준에 기반으로 하고 있으며, 설계변경을 표현하기 위하여 ISO STEP의 Action Entity를 사용하고 있다. 그러나, Action Entity는 구체적인 설계변경 이력과 제약조건은 제공하지 못하고 설계변경 관리에 관한 표준만을 제공하고 있다.

본 논문에서 제안한 표현방법은 제품구조 중심의 접근 방법에 의한 이득을 제공할 뿐만 아니라 설계변경 자료의 통합된 저장소로서의 재사용 가능한 설계변경 개체를 제공한다. 이 설계변경개체는 관련된 자료, 연산, 그리고 제약조건을 통합할 수 있을 뿐만 아니라 설계변경의 이력도 효과적으로 기록한다. 또한 설계변경에 관한 제약조건이 ECA 규칙에 의하여 정의되었기 때문에 이들이 쉽게 상용 데이터베이스 관리 시스템에 구현할 수 있다.

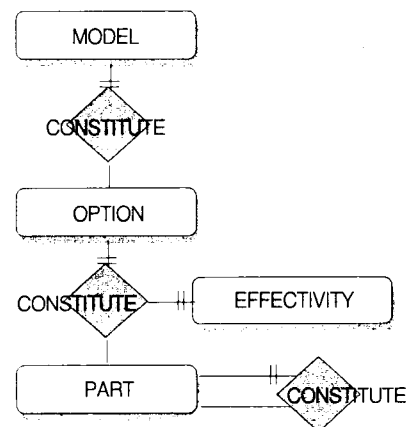
3. 설계변경을 위한 제품 구조

3.1 제품구조

이 장에서는 제품자료 모델의 대상이 될 제품 모델(Product model), Option, 그리고 조립부품(Assembly part)으로 구성된 제품구조[도남철, 2000]를 설명한다. 제품모델 개체는 제품형상

(Product configurations)을 포함하고 있는 최종 제품을 나타낸다. 제품모델 개체의 하위 요소로서 Option 개체는 제품형상을 위한 모듈을 제공하는 조립부품의 집합을 나타낸다. 조립부품 개체와 이들간의 구성(조립)관계(Constituent relationship)는 일반적인 기계 조립품의 제품구조를 표현한다. 이 개체들과 개체 사이의 관계는 다음 장에서 제안되는 제품자료 모델에서 표현된다.

제품에 대한 다양한 선택이 가능하게 하기 위하여 Option 개체는 유효성(Effectivity)이라는 주어진 조건에 의하여 서로 다른 제품구조를 정의할 수 있는 변형자(Variant) 형태로 정의되어 있다[ISO 10303-203, 1994]. 이 Option 개체는 일련번호 유효성이 적용되는 일련의 조립부품 개체와 구성관계로 연결되어 있다. 이 구조는 하나의 Option 개체가 하위 제품구조를 특정 제품 일련번호에 의하여 변경할 수 있다는 의미가 된다. 다음 그림은 지금까지 언급한 제품구조를 구성하는 3개의 주요 개체와 유효성 개체를 보여 준다.



(그림 1) 제품구조를 구성하는 개체와 유효성 개체의 개체관계도

추가(Add), 삭제(Cut), 그리고 치환(Replace)은

제품의 구조를 바꿀 수 있는 연산들이다. 추가 연산은 부품사이의 새로운 구성관계를 생성한다. 이 연산은 한 부품을 다른 부품의 한 하위 부품으로 추가하는 것을 의미한다. 삭제 연산은 부품사이의 구성관계를 삭제하며 이 의미는 한 부품과 다른 부품의 하위부품관계를 삭제하는 것을 뜻한다. 치환 연산은 한 조립품에서 하위부품을 삭제하고 대신 다른 부품을 하위부품으로 추가하는 것을 뜻한다. 이 연산과 삭제와 추가를 조합한 연산과의 차이점은 연산에 참여하는 두 부품사이에 호환성관계가 존재하는데 있다.

'Version up' 연산은 설계변경에서 사용되는 또 다른 중요한 연산 중에 하나이다. 이 연산은 기존의 부품의 구조를 복사하고 이의 Version 항목을 변경한다(예로 항목의 값이 '---'에서 '---A'로 변경). 새로운 Version을 생성하기 위하여서 설계자는 'Version up' 연산의 결과를 제품구조 변경에 적용되는 3가지 연산을 사용하여 변경시킨다.

3.2 설계변경에 대한 무결성 제약조건

설계변경이 진행되는 동안 설계자는 기술자료에 대한 무결성 제약조건을 만족시켜야 할 뿐만 아니라 제품구조에 대한 제약조건도 만족시켜야 한다. 본 논문은 여러 무결성 제약조건 중 중첩 설계변경과 Option에 대한 동시설계 변경 적용 제약조건을 고려한다. 첫 번째 제약조건은 제품구조가 설계변경 중일 때는 다른 설계변경에 의한 해당 제품구조 변경을 금지하는 제약조건이다. 만일 설계자가 한 조립부품의 하위부품의 변경을 시작했다면 제품자료 모델은 같은 조립품의 하위부품을 변경하려는 다른 설계변경 시도를 방지하여야 한다. 이러한 제약조건은 조립부품의 설계변경에 관련이 있기 때문에 조립부품의 제품구조에 정의되어야 한다.

두 번째 제약조건은 설계변경 되는 조립부품을 공유하고 있는 다수의 Option으로의 설계변경 전달을 제어한다. 이 전달에서 설계변경은 다수의 Option에 선택적으로 적용되어야 한다. 만일 하나의 Option이 속한 제품모델이 생산중지로 제품생산에 적용 불가능하면 설계변경은 해당 Option에 전달되어서는 안 된다. 제품자료 모델은 설계자가 설계변경을 특정한 Option에 적용할지 여부결정을 도와 주어야 한다. 이러한 제약조건은 Option의 설계변경에 관계 있으므로 제품구조 중에 Option 상에 정의된다.

3.3 설계변경의 분류

설계변경은 제안한 제품구조와 설계변경의 추상화 수준을 고려하면 3가지 종류로 나눌 수 있다. 조립부품 설계변경(Assembly type EC)은 제품설계와 관련된 기술적인 관점의 자료와 함께 조립부품 간의 변화에 관계된 정보를 포함한다. 반면에 Option과 제품모델 설계변경(Option and Model type EC)은 제품형상의 변화에 관계된 정보를 포함하며 이들은 판매나 생산에 관련된 정보를 포함한다. 그러므로 이들의 추상화 수준은 조립부품 설계변경과 차이가 있다.

서로 다른 추상화 수준의 자료 접근을 지원하는 설계변경 분류를 구현하기 위하여 제안된 제품자료 모델은 설계변경 개체를 제공한다. 설계변경 개체는 제약조건을 강제 시킬 대상을 제공하고 설계변경 이력을 저장한다. 또한 모든 자료, 연산, 그리고 제약조건은 제안된 설계변경 개체에 통합되어 개체 자체가 설계변경 자료의 통합 저장소가 될 수 있다.

관련 연구[Agilesoft, 1999 ; D.H. Brown, 1997]을 검토하여, 본 논문에서는 설계변경 관리에서 요구되어지는 기능요구를 정의하였다. 다음 장에서는 다음의 기능요구 사항들을 만족시키는

제품자료 모델을 제공한다.

- 설계변경을 위한 기본 제품구조와 관련 연산을 제공
- 설계변경 동안의 제품구조 변경 이력을 제공
- 설계변경 동안의 제품구조에 적용되는 무결성 유지
- 설계변경 자료와 제약조건을 통합할 수 있는 설계변경 단위 제공

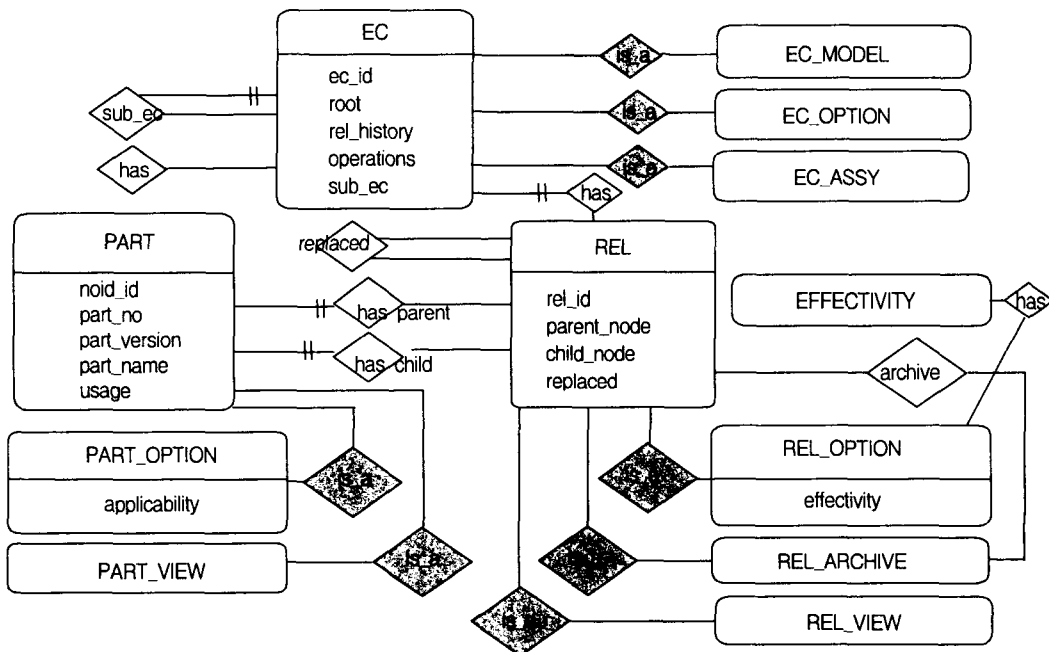
4. 설계변경을 위한 제품자료 모델

이 장에서는 이전 장에서 설명한 모든 요구사항을 만족시킬 수 있는 자료, 연산, 그리고 제약조건에 대한 정형화된 제품자료 모델을 제안한다. 이 모델은 관련사항을 통합하기 위하여 능동형 객체지향적 데이터베이스 모델을 기반으로 한다. 주어진 요구사항을 효과적으로 지원하기 위하여 제안된 모델은 상속과 Member function

을 제공하는 객체지향적 자료정의 언어(Object-oriented data definition language)를 이용하여 정의한다[Do, 1996 ; Kim, 1990]. 또한 제품구조에 정의된 제약조건을 정의하기 위하여 ECA 규칙이 사용된다. (그림 2)는 해당 모델을 개체관계도표(Entity relationship diagram)로 나타낸 것이다.

4.1 기본제품구조와 관련 연산들

제안된 제품자료 모델에서 'REL' 개체의 'has_parent'와 'has_child' 항목과 'PART' 개체는 부품사이의 구성관계와 부품을 나타낸다. 해당 개체들은 제품의 조립구조를 표현하는 Directed acyclic graph[Aho *et al.*, 1983] 상의 Node와 Edge에 대응된다. 'PART_OPTION'와 'REL_OPTION' 개체는 'PART'와 'REL' 개체를 각각 상속 받으며 Option과 Option에 속한 조립부품과의 구성관계를 표현한다. 나아가 'REL_OP-



(그림 2) 제안된 제품자료 모델의 개체관계도

TION' 개체는 'REL' 개체를 확장 시켜 유효성 자료를 저장할 수 있도록 하고 있다. 제품의 일련번호 유효성을 표시하는 'EFFECTIVITY' 개체는 'REL_OPTION' 개체와 관계를 가지고 있으며 일련번호 유효성을 이용하여 제품의 구조를 정의한다.

'REL' 개체에 대한 'add', 'cut', 그리고 'replace' Method가 제품구조를 변경시키는 추가, 삭제, 그리고 치환 연산을 표현하고 있다. 'REL_OPTION' 개체는 'REL' 개체의 모든 Method를 상속 받는다. 객체지향적 개념의 중요 개념인 상속성을 사용하여 'REL'의 정의를 재사용하여 'REL_OPTION' 개체를 정의하게 된다. 'EC' 개체와 'REL' 개체의 'replaced' 항목을 사용하여 제품구조의 변화인 설계변경의 이력을 기록할 수 있다. 'EC' 개체는 'REL' 개체의 Instance와 적용된 연산의 쌍을 순서(Sequence)로 기록하여 제품구조의 변화를 기록한다.

제안된 모델은 'REL' 개체의 하위 개체인 'REL_ARCHIVE' 개체를 지원한다. 이 개체는 치환되거나 삭제된 'REL' 개체를 설계변경 이력으로 남기기 위하여 저장한다. 치환이나 삭제 연산이 일어나면 연산이전의 'REL' 개체의 Instances들은 'REL' 개체로부터 'REL_ARCHIVE' 개체로 이동된다. 저장된 Instance 개체는 제품구조가 생성될 때는 사용되지 않지만 설계변경 이력을 제공할 때는 사용된다. 제안된 제품자료 모델은 구성관계를 설계변경 이력의 저장용도로 사용하므로써 설계변경을 제품구조와 밀접하게 통합시킬 수 있다.

'PART_VIEW' 개체는 설계에서 작성된 제품구조를 공유하기 위한 관련 분야(예로 공정설계 및 지침서 출판 등)의 응용 관점(Application view)을 표현한다. 'PART_VIEW'는 하위부품으로 'PART_ASSY'를 가지며 이 관계를 표현하기 위하여 'REL_VIEW' 개체를 사용한다. 또

한 해당 응용 관점에서 공유되는 부품을 표시하기 위하여 'PART' 개체의 'usage'항목을 사용한다. 'PART_VIEW'와 'REL_VIEW' 개체는 설계의 분야의 통합된 제품구조 공유 및 설계변경 전달에 사용될 수 있으며 이에 관한 연구는[도남철, 2003]에서 다루고 있다.

4.2 설계변경을 위한 제품구조의 제약조건

제안된 모델은 제품구조의 요소를 설계변경 단위로 사용하므로 하위 조립부품을 변경하는 모든 제품구조 변경을 금지할 수 있다. 설계변경에 관련된 제품구조에 참가하고 있는 'PART' 개체의 'working_flag' 항목은 제품구조를 잠가 중복된 설계변경을 방지한다. 다음의 정의는 해당 잠금 연산을 ECA 규칙의 생성자인 CONSTRAINT[Do, 1996]를 이용하여 정의한 것이다.

```
CONSTRAINT ON REL.replace( )
IF obj.parent_node.working_flag = TRUE
EXECUTE INVALIDATE TRANSACTION ;
```

이 제약조건은 'REL' 개체의 치환 연산(Replace())을 감시하게 된다. 만일 이 연산이 발생하면 제약조건은 연산이 일어난 'REL' 개체의 상위 부품의 'working_flag'의 값을 검사하고 연산을 실행하는 사용자가 현재 'PART' 개체의 변경 Session 사용자가 아니면 치환연산을 중지시킨다. 같은 종류의 제약조건이 추가와 삭제 연산에도 적용된다.

다음의 Method는 해당 부품과 하위부품 모두의 'working_flag' 항목의 값을 참으로 바꾸는 기능을 한다. 이 Method는 'EC' 개체의 'status' 항목 값이 'ON'으로 변경될 때 해당 개체의 'root' 항목에 적용된다. 그러면 설계변경 개체를 소유한 설계자가 독점적으로 제품구조를 변경할 수 있으며 제품구조의 변화는 'EC' 개체에 저장되

기 시작한다. 'EC' 개체와의 관계는 4.3에서 자세히 다룬다.

```

PART.turn_on()
{self.working_flag = TRUE ;
WHILE (obj is a REL AND obj.get_parent_
node() = self)
{obj.get_child_node().turn_on() }

```

이 Method는 'EC' 개체의 'root' 항목에 해당하는 'PART' 개체의 하위부품에 재귀적으로 적용되어 해당 'PART'의 'working_flag' 항목 값을 바꾸어 부품들을 잠그게 된다.

설계변경의 다수 Option에 동시적용에 관한 제약조건을 지원하기 위하여 제안된 모델은 'applicability' 항목과 다음의 제약조건 정의를 가지고 있다.

```

CONSTRAINT ON REL_OPTION.replace()
EXECUTE obj.multi_app() ;

```

이 제약조건은 'REL_OPTION' Instance 개체에 치환 연산이 일어날 경우에 다음의 'multi_app' Method를 실행시키게 된다.

```

REL_OPTION.multi_app()
{WHILE( rel is REL_OPTION AND
rel.get_child_node() = self.child_node)
{IF rel.get_parent_node().applicability =
TRUE THEN
NEW rel_option(NEW ID,
rel.get_parent_node(),self.child_node,
rel.get_id()); }
}

```

이 Method는 변경된 PART Instance('self.child_node')를 'child_node'로 항목의 값으로 가지고 있는 'REL_OPTION' Instance를 찾고 Parent node의 Option 개체의 Applicability가 참일

경우에만 변경된 부품과 해당 Option을 연결하는 새로운 'REL_OPTION' instance를 생성한다.

4.3 제약조건 관리를 위한 설계변경 개체

제안된 제품자료 모델은 앞 장에서 설명한 3가지 종류의 설계변경을 'EC' 개체를 상속 받은 3개의 개체로 구현하고 있다. 또한, 'EC' 개체는 제약조건 대상제품(Target products)을 'root' 항목으로 설계변경 이력 기록을 'REL' 개체의 순서로 제공한다. 'root' 항목은 해당 제품구조의 Root 부품을 지칭하여 표현하고 있다. 이 'EC' 개체의 Instances를 설계변경 개체라고 칭하며 간단한 설명을 위하여 제품모델을 제외한 Option과 조립부품 설계변경에 대하여 설명한다.

'EC' 개체를 상속 받은 'EC_ASSY' 개체는 일반 조립부품사이의 구조 변경을 표현한다. 즉 조립부품을 변경하기 위한 설계변경에 관한 기술적 정보를 담고 있다. 이 개체는 중복된 설계변경에 관한 제약조건에 관계가 있다. 'EC' 개체의 'status' 값이 활성화(Active) 상태로 변하게 되면, 이 개체는 'root' 부품의 제품구조를 잠그고 제품구조 변경을 기록하게 된다. 이 잠금과 기록은 해당 'EC' 개체가 활성 상태인 동안 계속되어 중복된 설계변경을 방지한다.

'EC' 개체를 상속 받는 또 다른 개체인 'EC_OPTION' 개체는 Option과 조립부품사이의 제품구조변경을 표현한다. 이 개체는 제품변경 비용 등의 설계변경의 관리적인 면을 설명하는 제품형상과 설계변경 유효성(EC effectivity)에 관한 자료와 관련된다. 다수 Option에 대한 동시 설계변경 적용에 관한 제약조건은 Option 설계변경과 관련이 있다. 하나의 Option 설계변경은 Applicability가 없는 Option을 'root'로 생성해서는 안 된다. 다음은 제약조건은 그러한 종류의 설계변경 개체의 생성을 방지한다.

CONSTRAINT ON EC.create()

```
IF new.type = 'OPTION' AND new.root.
  applicability = 'OFF' THEN
EXECUTE INVALIDATE TRANSACTION
```

이 제약조건은 설계변경 개체의 Root 부품의 Applicability를 개체가 생성될때 검사한다. 만일 Applicability 값이 'OFF'이면 해당 생성 Transaction을 무효화 시키게 된다.

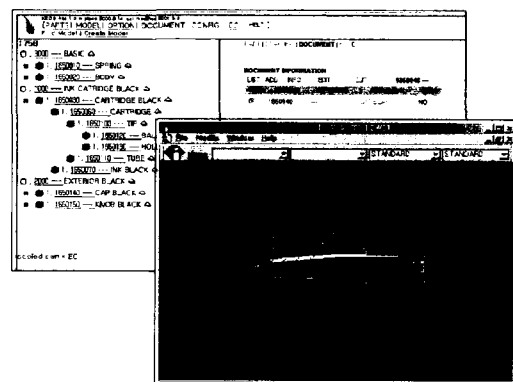
한 조립부품 설계변경이 Option 설계변경의 하위 설계변경(Sub EC)임은 설계변경 개체의 'sub_ec' 항목에 의하여 나타내어 진다. 이 모델은 설계자가 설계변경 자료를 일관되게 재사용하고 공유할 수 있도록 한다. 만일 여러 Option에 공유되어 사용중인 한 조립부품이 변경될 경우, Option 설계변경 개체는 하위 설계변경으로 연결될 조립부품 설계변경을 재 사용할 수 있다. 이러한 이용방법은 설계변경 자료의 중복을 방지하고 일관성을 유지할 수 있도록 한다. 또한 이 모델은 현재 참가하는 설계변경의 root 부품과 하위 설계변경과의 Applicability 제약조건을 유지할 수 있도록 한다.

하위 설계변경 관계를 이용하여, 설계자들은 서로 다른 종류의 설계변경을 하나의 통합된 설계변경자료 저장소에 구조화 시킬 수 있다. 또한 각각의 설계변경은 'root' 항목을 이용하여 관련된 제품구조에 밀접하게 연관되어 있다. 제약조건도 또한 설계변경 개체에 연관되어 있다. 중복된 설계변경에 관한 제약조건은 조립부품 설계변경의 시작과 끝에서 정의될 수 있으며, 다수 Option에 대한 설계변경 동시적용 제약조건은 Option 설계변경과 'REL_OPTION' 개체에 적용되어 있다. 이 관련된 제품구조와 통합된 저장소는 설계자들이 설계변경에 관련된 모든 정보를 접근하고 분석할 수 있는 통일된 환경을 제공한다. 다음 장에서 통합된 환경의 한 예가

소개된다.

5. 구현 예

본 논문에서 제안된 요구사항을 만족시키기 위하여 Web 기반 제품정보관리 시스템을 개발하였다. 구현을 위하여 능동형 객체지향 데이터베이스를 기반으로 제안된 제품자료 모델을 관계형 데이터베이스의 Table과 데이터베이스 응용 Program으로 전환하였다. 그리고 CONSTRAINT 생성자를 관계형 데이터베이스의 Trigger 생성자로 전환하였다. 개발된 시스템은 기본 제품구조관리 시스템의 기능인 부품목록 관리, 문서관리, 제품형상관리, 그리고 제품구조와 설계변경 관리 기능을 제공한다. (그림 3)은 제품형상과 조립구조를 중심으로 하는 제품구조의 확장과 관련된 기술도면을 보여 주는 개발된 시스템의 사용자 환경을 보여 주고 있다. 이 장에서는 중복된 설계변경에 대한 제약조건과 다수 Option에 대한 설계변경 동시 적용 제약조건의 구현을 살펴본다. 구현된 내용을 보기 위하여 간단한 가상 제품의 설계 예가 준비되었다.



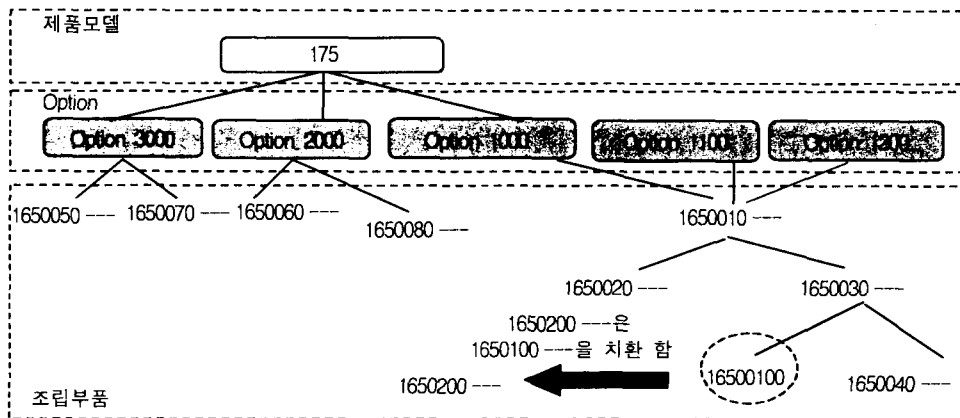
(그림 3) 개발된 제품정보관리 시스템 시제품 화면

(그림 4)는 예제 제품의 Option과 조립부품의 제품구조를 보여 준다. 이 175 제품모델 개체는

3개의 Option으로 구성되어 있다. 이 Option은 각기 모델명과 일련번호로 구성된 유효성 값을 지닌 체 조립부품의 제품구조와 연결되어 있다. 설계변경은 이 연결과 유효성을 바꾸게 된다. 조립부품의 제품구조는 제품의 물리적 제품구성관계를 표현한다. 부품도면이나 정의 등의 주요 기술문서가 이 제품구조에 통합되어 관리된다. 예에서는 주어진 조립부품중에 1650100 --- 부품을 새로운 부품으로 변경할 경우의 관련된 제약조건과 제품구조의 변경에 대하여 알아본다. 필요에 의하여 설계자들은 1650100 --- 부품

을 부품번호 1650200 ---인 새로운 부품으로 변경하기로 결정하였다. 이 변경으로 기존의 1650100 ---을 포함하고 있는 1650010 --- 조립부품의 제품구조를 (그림 5)의 설계변경과 같이 바꾸어야 했다. 이 변경 사항은 인식번호가 'EC10'으로 정해진 조립부품 설계변경에 저장된다. (그림 6)의 설계변경의 'A history of structure changes' 항목이 제품구조 변경의 이력을 보여 준다. 이후 설계변경에 대한 기술적인 정보가 설계변경 개체에 저장된다.

예의 조립부품 설계변경은 3개의 서로 다른



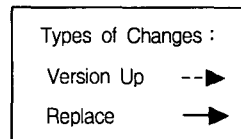
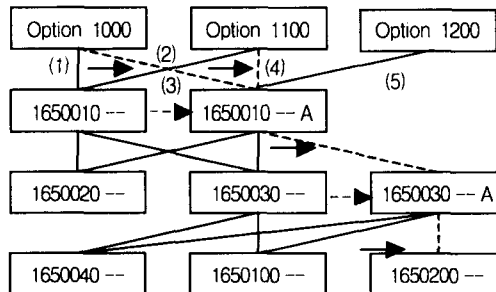
(그림 4) 예제 가상제품의 Option과 조립부품 구조

Effectivities on Edges :

- (1) 1~99
- (2) 100~current
- (3) 1~149
- (4) 150~current
- (5) 1~current

Operations :

- 1 : version up 1650010 --- to 1650010 --- A
- 2 : version up 1650030 --- to 1650030 --- A
- 3 : replace 1650030 --- to 1650030 --- A (under 1650010 --- A)
- 4 : replace 1650100 --- to 1650200 --- (under 1650030 --- A)
- 5 : replace 1650010 --- to 1650010 --- A (under option 1000)
- 6 : replace 1650010 --- to 1650010 --- A (under option 1100)



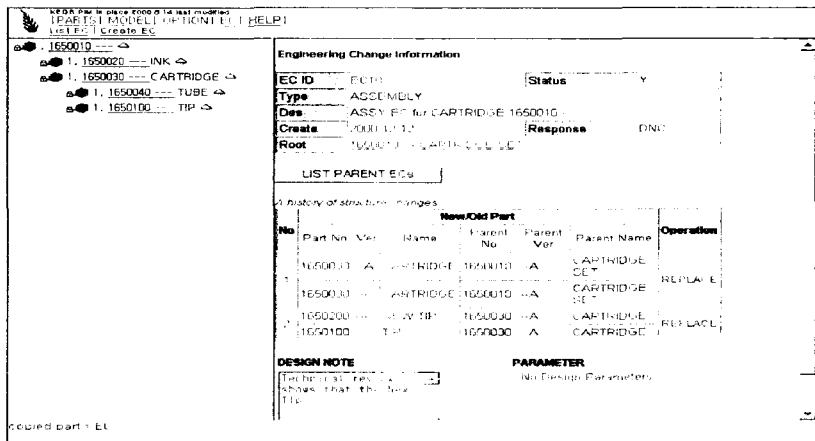
(그림 5) 1650100 --- 조립부품의 설계변경 예

Option, 1000, 1100 그리고 1200에 영향을 주게 된다. 그러므로 설계자는 서로 다른 3개의 설계변경을 생성할 수 있다. 하지만 1200 Option이 Applicable 하지 않기 때문에 단지 2개의 Option 설계변경, 'EC11' 그리고 'EC12'가 생성되어 Option과 조립부품사이의 설계변경 이력을 기록하게 된다. 이 두 개의 설계변경 개체는 설계변경에 대한 새로운 입력이나 중복이 없이 'EC10'을 자신들의 하위 설계변경으로 공유하게 된다.

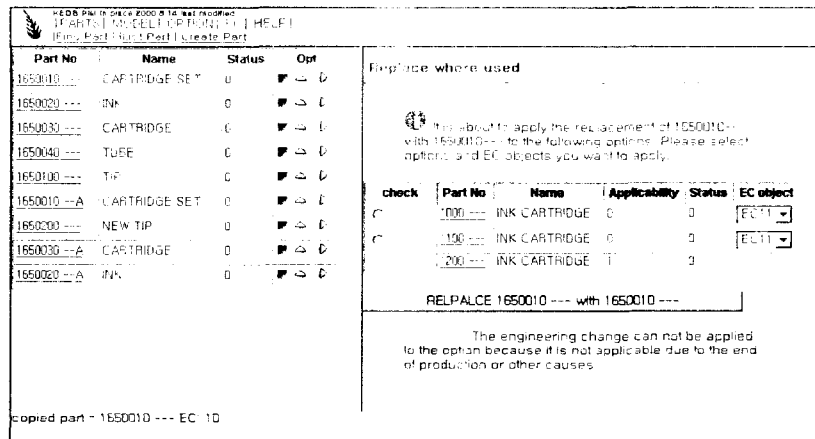
제품구조에 관계된 제약조건의 예로 'EC10'

설계변경을 고려할 수 있다. 만일 한 설계자가 해당 설계변경 개체의 'status' 값을 'ON' 시켰다면, 설계변경의 'root' 부품이 포함하는 하위 부품구조는 (그림 6)과 같이 모두 잠기게 된다. 각 부품의 자물쇠 그림은 해당 부품이 잠겨 있음을 뜻한다. 'Turn on' 연산은 설계자가 제품구조를 변화시키기 시작한다는 것을 뜻한다. 그러므로 해당 'status' 항목 값이 'Turn off'될 때까지 해당 제품구조는 잠겨진다.

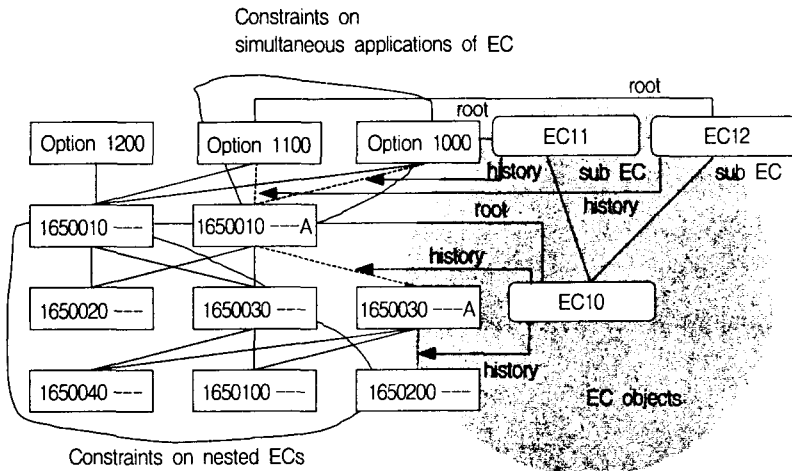
Option에 관한 제약조건의 예로 'Replace where used' 연산을 예로 들 수 있다. 이 연산



(그림 6) 'EC'의 설계변경 개체



(그림 7) 'Replace Where Used' 연산이 적용될 경우의 제약조건



(그림 8) 예제에 참여한 설계변경 개체와 제품구조사이의 통합된 관계

은 한 조립부품을 공유하고 있는 다수의 Option 을 검색하여 설계자가 변경한 조립부품을 해당 Option에 적용할 것인지를 결정하는데 도움을 준다. 예로 만일 Option 1200이 Applicable하지 않다면 1650010 ---에 대한 'Replace where used' 연산은 (그림 7)의 오른쪽 창과 같은 내용을 보여 주어야 한다. 그림에서 현재 활성화된 설계변경 개체 'EC11'은 Option 1200이 Applicable 하지 않으므로 Option 1200을 제외한 두 Option에 적용될 수 있다((그림 7)의 'INTEG-01-메시지 참조). 그러므로 다수 Option에 대한 설계변경에 대한 제약조건은 설계자가 제품구조를 일관되게 변경할 수 있도록 도와 준다.

설계자는 또한 설계변경 'EC10'의 'root' 항목을 통하여 관련된 제품구조에 접근할 수 있다. (그림 6)의 'EC10'의 왼쪽 창의 제품구조는 제품 정보관리 시스템에서 관리하는 설계변경 개체의 'root' 개체에서 생성된 것이다. 또한 'EC11'과 'EC12'가 'EC10'과 관련 있다는 사실을 'list parent ECs' 연산을 통하여 알 수 있으며 그들의 Option 제품구조도 검색할 수 있다. 그러므로 설계자는 설계변경 개체와 관련된 Option과 조립제품구조를 통하여 설계변경 관련 자료와

제약조건을 통합적으로 접근할 수 있다. (그림 8)은 설계변경 개체와 예제 제품구조 사이의 관계를 보여 주고 있다.

6. 결 론

최근의 전산화된 제품개발 환경에서는 무결성이 유지된 설계변경 정보가 회사내 뿐만 아니라 회사 밖의 고객이나 공급자에게도 빠르게 전달되어야 한다. 아울러 현대의 제품정보관리 시스템은 제품구조 중심의 자료 관리를 통하여 제품개발 전 기간에 걸쳐 제품정보의 일관성을 유지시키려 하고 있다. 본 연구에서는 능동형 객체지향 데이터베이스 모델을 도입하여 설계변경에 관한 제약조건을 효과적으로 표현하는 제품자료 모델을 제안하였다.

제안된 제품자료 모델은 설계변경 관리의 기본적인 기능을 제공하고 제품구조의 무결성 제약조건을 효과적으로 표현한다. 이 모델의 설계변경 개체는 제품구조에 관계된 제약조건을 관리하고 제품구조의 변화를 기록한다. 이 개체들은 또한 설계변경 자료를 제품구조에 따른 서로 다른 추상화 차원에서 관리한다. 또한 설계변경

개체간의 포함 관계는 설계자들이 설계변경자료의 일관성을 중복이나 재 입력 없이 관리할 수 있도록 한다. 그러므로 설계자는 설계변경에 대한 자료, 연산, 그리고 제약조건을 관련된 포함 관계와 제품구조를 통하여 상호 접근할 수 있다.

본 논문에서 제안한 기능적 요구사항을 만족시키는 Web 기반의 제품자료관리 시스템이 제안된 제품자료 구조 모델에 근거하여 구현되었다. 이 시스템은 두 가지 제약조건이 설계변경 개체의 활성화와 공유하는 다수 상위부품에서의 부품치환에 적용될 수 있음을 보여 준다. 또한 적용 예는 제안된 제품자료 모델의 설계변경 개체가 통합된 설계변경 자료 저장소를 제공할 수 있음을 보여 준다.

추후 연구에서 해결해야 할 문제 중 하나는 복잡한 제품설계에 데이터베이스 Trigger 기반의 ECA 규칙을 적용할 경우의 효율성 문제이다. Trigger는 데이터베이스 관리시스템에 많은 부담을 주는 생성자인 동시에 복잡한 제품설계 정보에 적용될 경우 많은 제약사항이 따를 것으로 예측된다. 특히 본 논문에서 제안한 제품구조 잠금 기능은 많은 데이터베이스 자원을 요구할 것으로 예측된다. 그러므로 본 논문에서 제안한 모델을 실용화 하기 위하여는 복잡한 설계 정보에 대한 ECA 규칙의 효율성을 높일 수 있는 방법에 관한 연구가 필요하다.

또 다른 추후 연구방향은 제품구조 무결성 유지를 위한 제약조건 생성자를 개발하는 것이다. 우리는 데이터베이스 관리시스템에서 제공하는 일반적 용도의 생성자(CONSTRAINT)를 사용하여 제품구조에 관련된 제약조건을 정의하였다. 이 생성자는 제품사이에 구성관계를 이루는 제품구조를 통해 전달되는 변경 전달을 효과적으로 지원할 수 없다. 그러므로 제품구조상에 제약조건을 표현하고자 하는 설계자에게는 설

계변경을 고려한 일반적인 제약조건 정의 생성자가 필요하다.

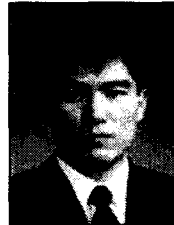
참고 문헌

- [1] 도남철. (2000). 설계변경관리를 위한 제품자료 표현. 한국경영과학회-대한산업공학회 춘계공동 학술대회 논문집.
- [2] 도남철. (2003). 일관된 제품자료관점을 지원하는 설계변경 전달에 관한 연구. 한국 CAD/CAM학회 학술발표회. 429-436.
- [3] Agilesoft. (1999). Why Document Management Systems Don't Work for the ECO Process. Agilesoft White paper.
- [4] Aho, A.V., Hopcroft, J.E., & Ullman, J.D. (1983). Data Structures and Algorithms. Mass. : Addison-Wesley Publishing Company Inc.
- [5] Baya, V. (1996, September). Information Handling Behavior of Designers During Conceptual Design : Three Experiments. Doctoral dissertation, Department of Mechanical Engineering, Stanford University.
- [6] Baya, V., & Leifer, L. (1994, September). A Study of the Information Handling Behavior of Designers During Conceptual Design. Proceedings of The 6th International Conference on Design Theory and Methodology.
- [7] Baya, V., & Leifer, L. (1996). Understanding Information Management in Conceptual Design. in Cross, N., Christiaans, H. and Dorst, K. (Eds.), Analyzing Design Activity. 151-168. John Wiley & Sons.
- [8] Brown. D.C. (1998, November). Using Design History Systems for Technology Transfer. Proceedings of MIT-JSME Workshop on Cooperative Product Development. 545-559.

- [9] Casotto, A., Newton, A.R., & Sangiovanni-Vincentelli, A. (1990). Design Management based on Design Traces. Proceedings of The 27th ACM/IEEE Design Automation Conf. 136-141.
- [10] Choi, I.J., & Bae, S.M. (2001, Vol. 18, No. 2). An Architecture for Active Product Configuration Management in Industrial Virtual Enterprise. Journal of Advanced Manufacturing Technology. 105-121.
- [11] Dayal, U., Blaustein, B., Buchmann, A., Chakravarthy, U., Hsu, M., Ledin, R., McCarthy, D., Rosenthal, A., & Sarin, S. (1988, March, Vol. 17, No. 1). The HiPAC project : combining active databases and timing constraints. SIGMOD Record. 51-70.
- [12] D.H.Brown Associates, Inc. (1997). Product vs. Process in Design-to-Manufacturing Integration. D.H.Brown Technical Report.
- [13] Do, N.C. (1996). Backward Propagation of User-defined Integrity Constraints in Active Object-oriented Databases. Doctoral dissertation, Pohang University of Science & Technology.
- [14] Do, N.C., & Choi, I.J. (1996, Vol. 9, No. 4). Implementing integrity constraints of EXPRESS in an active object-oriented database. Journal of Computer Integrated Manufacturing. 311-326.
- [15] Enovia. (1999). Enovia VPM 1.0 Users Manual. Enovia.
- [16] Gates, R. (1996, November). Production Implementation of STEP for the C-17 program. Proceedings of The international conference on Concurrent Enterprise.
- [17] Hamer, P. van den., & Lepoeter, K. (1996, January, Vol. 84, No. 1). Managing Design Data : The Five Dimensions of CAD Frameworks, Configuration Management, and Product Data Management. Proceedings of The IEEE. 42-56.
- [18] Hardwick, M., Downie, B., Kutcher, M., & Spooner, D. (1995, Vol. 15, No. 1). Concurrent Engineering with Delta Files. IEEE Computer Graphics and Applications. 62-68.
- [19] Hardwick, M., Spooner, D., Downie, B., Ferris, M., Jiang, Z., & DeWeese, T. (1994, August). A STEP Entity Control System for Concurrent Engineering. Proceedings of The first International Conference on Concurrent Engineering : Research and Applications.
- [20] ISO 10303-203. (1994). STEP PART 203 : Configuration Controlled Design. ISO.
- [21] Katz, R.H., & Chang, E. (1987). Managing Changes in a Computer-Aided Design Data. Proceedings of the 13th VLDB Conference. 455-462.
- [22] Katz, R.H. (1990, Vol. 22, No. 4). Toward a Unified Framework for Version Modeling in Engineering Databases. ACM Computing Surveys. 375-408.
- [23] Kim, W. (1990). Introduction to Object-Oriented Databases. Mass. : MIT Press.
- [24] Lee, K.S., & Lee, K. (2001, Vol. 44, No. 3). Framework of an evolutionary design system incorporating design information and history. Computers in Industry. 205-227.

- [25] McCarthy, D.R., & Dayal, U. (1989, March). The architecture of an active data base management system. Proceedings of ACM SIGMOD Conference on Management of Data. 215-223.
- [26] PDM Implementer Forum. (2002, October). Usage Guide for the STEP PDM Schema Release 4.3. [Web Site]. http://www.pdm-if.org/pdm_schema/.
- [27] Peng, T.K., & Trappey, Amy J.C. (1998, Vol. 14, No. 2). A step toward STEP-compatible engineering data management : the data models of product structure and engineering changes. Robotics and Computer-Integrated Manufacturing. 89-109.
- [28] Shah, J., Jeon, D., Urban, S., Bliznakov, P., & Rogers, M. (1996, Vol. 28, No. 5). Database Infrastructure for Supporting Engineering Design Histories. Computer-Aided Design. 347-360.

■ 저자소개



도 남 철

Nam Chul Do is an assistant professor in the department of industrial engineering in GyungSang National university, Jinju, Korea. His research interests include product data representation, product data management (PDM), and PDM extensions within enterprises. He obtained his master degree in industrial engineering from Pohang University of Science and Technology in Pohang, Korea, in 1993 and completed his candidate of PhD at the same department in 1996, working on active object-oriented databases for engineering constraint management. In addition to 2 years in academia, he has spent 5 years in the engineering industry.