

STEP 을 이용한 PDM/CE 환경의 구축과 데이터 무결성 확인

유상봉*, 서효원**, 고굉욱*

Building A PDM/CE Environment and Validating Integrity Using STEP

S. B. Yoo, H. W. Suh, K. W. Ko

ABSTRACT

In order to adapt today's short product life cycle and rapid technology changes, integrated systems should be extended to support PDM (Product Data Management) or CE(Concurrent Engineering). A PDM/CE environment has been developed and a prototype is presented in this paper. Features of the PDM/CE environment are 1) integrated product information model (IPIM) includes both data model and integrity constraints, 2) database systems are organized hierarchically so that working data cannot be referenced by other application systems until they are released into the global database, and 3) integrity constraints written in EXPRESS are validated both in the local databases and the global database. By keeping the integrity of the product data, undesirable propagation of illegal data to other application system can be prevented. For efficient validation, the constraints are distributed into the local and the global schemata. Separate triggering mechanisms are devised using the dependency of constraints to three different data operations, i.e., insertion, deletion, and update.

Keywords: PDM, CE, STEP, system integration, multidatabase system, integrity constraints, hierarchy of constraints.

* 인하대학교 자동화공학과

** 한국과학기술원 산업공학과

1. 서론

최근 CAD/CAM, 네트워크, 정보 시스템, 소프트웨어 인터페이스, 그리고 그래픽스와 같은 컴퓨터 기술들이 급속하게 발달하여, 생산 시스템들의 통합을 어느 정도까지는 구현할 수 있게 되었다. 그러나, 통합 시스템 그 자체로는 최근 짧아지고 있는 제품의 수명주기와 급속한 기술 변화에 대한 요구들을 만족시킬 수 없고, 제품의 생산성 향상 및 시장 출하 기간을 단축하기 위하여 설계, 공학 해석, 그리고 제품의 생산이 동시에 수행될 수 있도록 하는 CE (Concurrent Engineering) [Carter and Baker, 1992] [Trapp, 1991]와 제품 데이터를 지속적으로 통제하고, 그 통제된 데이터를 요구하는 사용자에게 자동으로 전달함으로써 CE의 이점을 극대화할 수 있는 PDM (Product Data Management) [CIMdata, 1994]의 원리 및 방법론을 지원할 필요가 있다. 본 논문에서는 제품 정보모델의 국제표준인 STEP을 이용하여 PDM/CE 환경을 설계하였으며, 이 환경의 주요한 특징은 다음과 같다.

- 1) 통합 제품 정보모델 (IPIM): IPIM은 모든 데이터 모델들과 통합 시스템에 포함된 응용 시스템들을 위한 제품 데이터를 표현하는데 필요한 관계들을 포함한다. 이러한 IPIM은 STEP [ISO, 1992]에 정의되어 있는 공통 자원들과 응용 프로토콜들을 사용하면서 특정한 응용 영역의 정보모델들을 추가하여 구성하였다.
- 2) 계층구조 데이터베이스: 각각의 응용 시스템은 작업 데이터를 관리하는 그 자체의 지역 데이터베이스를 가지며, 일단 설계 과정이 완료되면, 작업 데이터는 다른 응용 시스템들이 참조할 수

있도록 하기 위하여 전역 데이터베이스로 전달된다. 이러한 데이터베이스들의 계층구조는 대다수의 공학 환경을 위한 구성 관리의 진행을 제공한다. 또한, 다양한 저장소 시스템에 저장되어 있는 STEP 데이터들을 액세스할 수 있도록 하기 위하여 SDAI (Standard Data Access Interface) [ISO, 1993]도 지원한다.

- 3) 무결성 제약조건 확인: EXPRESS [ISO, 1994]로 정의된 무결성 제약조건들을 확인한다. EXPRESS는 STEP의 공식적인 정보 모델링 언어로서 객체 지향적이며 다양한 무결성 제약조건들을 표현할 수 있다. IPIM에 정의된 무결성 제약조건들은 지역 및 전역 데이터베이스로 분산되며, 각 데이터베이스에 저장된 데이터들의 무결성은 해당되는 제약조건들에 의하여 확인된다.

STEP (Standard for the Exchange of Product model data)은 이종 공학 환경들 간의 정보 공유를 촉진하고자 하는 국제적인 노력의 일환이다. STEP에서는 특히 공학 데이터의 풍부한 의미를 잘 표현할 수 있도록 하는 정보 모델링 언어인 EXPRESS를 정의하고 있다. EXPRESS는 전형적인 객체 지향 언어이기 때문에, 엔티티의 정의 내에 데이터와 제약조건들을 함께 요약하여 표현할 수 있으며, 또한 엔티티들의 상속 계층구조를 지원한다. STEP의 정보모델은 공통 자원, 응용 자원, 그리고 응용 프로토콜로 분류된다. 연관된 공통 자원과 응용 자원은 응용 프로토콜에 의해 참조되고, 다른 응용 프로그램들이 응용 프로토콜을 통하여 제품 데이터를 교환한다. 공통 자원과 응용 자원에 정의되어 있는 정보모델은 전역 규칙들과

엔티티의 하위 타입들을 기술함으로써 응용 프로토타입에서 수정된다 [Owen, 1993]. 본 연구에서는 CAD (Computer Aided Design) 시스템, CAPP (Computer Aided Process Planning) 시스템, 그리고 MRP (Material Requirement Planning) 시스템을 대상으로 통합 시스템 프로토타입을 개발하였으며, 이러한 프로토타입을 위해 STEP의 정보 자원들을 이용하여 통합 제품 정보모델을 개발하였다. 통합 제품 정보모델은 기계 가공 제품의 형상 및 위상 정보, 형상 특징 정보, 공차 정보, 그리고 BOM (Bill Of Materials) 정보를 포함한다.

통합 시스템에서는 공유되는 데이터의 무결성이 적절히 유지되지 않는 한 전체 시스템의 행위를 예측할 수가 없게 됨으로 데이터의 무결성 확인 및 유지는 필수적이다. 특히 PDM/CE 환경에서는 한 시스템에 의해 전역 데이터베이스로 전달된 설계 데이터가 이 환경에 포함된 모든 응용 시스템들에 의해 이용되므로, 적절한 무결성 확인이 필요하다. 그러나, 데이터 무결성의 확인은 데이터베이스로부터 검색될 동일 타입 또는 다른 타입의 엔티티 인스턴스들에 대한 많은 교차 참조를 요구하기 때문에 종종 매우 많은 비용이 든다. 본 논문에서는 이러한 PDM/CE 환경에서의 효과적인 데이터 무결성 확인에 대한 방법을 다룬다. 전역 데이터베이스 시스템으로부터의 무결성 확인 오버헤드를 절감하기 위하여, 통합 제품 정보모델에 포함된 무결성 제약조건들의 일부분을 지역 스키마들로 분산시켜 지역 데이터베이스 시스템에 의해 확인될 수 있도록 하였다. 지역 및 전역 데이터베이스 시스템들을 위한 무결성 규칙들의 분산과 확인 과정에 필요한 알고리즘을 본 논문에서 제시하였다.

2. 관련 연구

STEP에 의해 개발된 자원 모델들과 방법론들이 시스템 통합, CE, PDM 등에 적용되고 있다 [Hardwick 외 3, 1996] [Wu 외 2, 1992] [Carver, Bloom, 1991] [Hardwick 외 3, 1995]. EXPRESS는 STEP 프로젝트에서 개발되었고, STEP에서 개발된 모든 정보모델들을 정의하는데 사용된다. 이러한 표준화 작업 외에도 EXPRESS는 다양한 연구 분야와 산업계의 제품 모델링 및 시스템 통합을 위하여 사용되고 있다 [Vuoskoski, 1994] [Divoux, 1994] [Schoenefeld, Bohm, 1993] [Eberl, 1994] [서효원, 유상봉, 1995]. 파일 시스템 또는 데이터베이스 시스템에 저장된 STEP 데이터를 위한 EXPRESS의 무결성 제약조건 확인은 다중 처리 [Mueller 외 2, 1993], 데이터 의존성 분석 [Yoo, Cha, 1993], 트리거링 방법 [도남철 외 2, 1993], 그리고 뷰 구체화 [Alt, 1994]에 의해 최적화 되고 있다. 그러나, 하나의 통합 시스템을 구성하는 다중 저장 시스템들에 저장되어 있는 제품 데이터의 무결성 확인을 위한 연구는 아직 구현되지 않고 있다.

상업용 DBMS들의 무결성 유지는 매우 단순하여, 단지 도메인 규칙과 트리거 액커니즘에 대해서만 제한적으로 지원되고 있다. DBMS의 응용 영역이 공학, 멀티미디어, 그리고 지리 정보 시스템 등으로 확장되고 있으므로 데이터베이스의 일관성을 강화하기 위한 더 일반적인 방법들이 필요하다. 여러 연구 단체에서 다양한 접근 방법들을 제시하고 프로토타입을 구현하고 있지만, 두 가지의 문제점이 있는데, 하나는 제약조건 위반을 발견하는 방법이고, 다른 하나는 발견된 위반을 정정하는 방법이다. 제약조건 확인을 위한 비용을 절감하기 위하여, 다양한 컴파일

기법들이 사용되고 있다 [Henschen 외 2, 1984] [Jagdish, Qian, 1992]. 제약조건 규칙들은 트랜잭션 이후에 데이터베이스의 일관성이 유지되는지를 확인하기 위한 등가의 조건들의 집합으로 변환된다. 활성 제약조건들에 의해 위반들이 정정되고 나면 [Cacace 외 4, 1990] [Urban 외 2, 1992], 데이터베이스의 일관성을 유지하기 위하여 데이터를 생성하거나 삭제한다. 트리거 메커니즘은 일반적으로 컴파일되고 활성화된 제약조건들을 가동하기 위하여 사용된다.

다중 데이터베이스 시스템들에 있어 제약조건 유지는 다수의 물리적 또는 논리적 인터페이스 층들을 탐색할 필요가 있기 때문에 비용이 더 많이 든다. 인터페이스 층들은 데이터베이스 시스템들이 서로 다른 경우 스키마 변환을 포함한다 [Sheth, Larson, 1990]. 이러한 오버헤드를 피하기 위하여 전역 제약조건 규칙들은 지역 규칙들의 집합으로 변환될 수 있고, 각각의 데이터베이스 시스템 내에서 확인될 수 있다 [Blakeley 외 2, 1989] [Gupta, Widom, 1993]. 전역 제약조건들과 약간의 수정을 포함한 데이터베이스 트랜잭션이 주어지면, 이에 필요한 지역 검사들이 결정된다. 이러한 방법들은 원격 데이터베이스들과의 불필요한 상호작용을 제거함으로써 평가 비용을 절감한다. 검사 집합들은 다수의 프로세서들에 걸쳐 분산될 수 있고, 또한 검사 집합들이 논리적으로 독립되어 있다면 동시에 확인될 수 있다 [Blelloch, 1986].

많은 기존의 연구에서 무결성 제약조건들은 대수 비교 연산자들을 지닌 1차 논리 함수로 표현되고 있으나, EXPRESS에서는 제약조건들이 일반적인 알고리즘을 포함하는 절차적인 프로그램으로 표현되고 있으며, 이러한 절차적인 프로그램을 이용하여 다양한 제약조건을 표현할 수 있다. 본 논문에서는 우

선 EXPRESS로 표현될 수 있는 제약조건들을 분석하고, 분석된 제약조건들을 지역 데이터베이스 시스템들로 분산한다. 응용 시스템들로부터 전달되는 모든 설계 데이터들은 미리 해당되는 지역 데이터베이스들에서 제약조건으로 무결성이 확인되며, 또한 전체 PDM/CE 환경에서 데이터의 일관성이 유지될 수 있도록 전역 데이터베이스 시스템에 저장된 규칙들이 전달된 데이터에 적용된다.

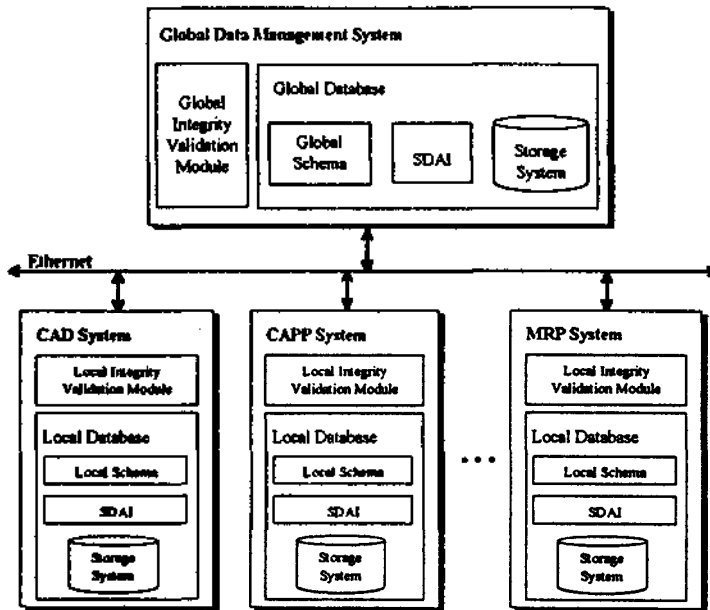
3. PDM/CE 환경의 구조

본 연구에서 개발한 PDM/CE 환경은 STEP의 방법론을 이용하였으며, 이 환경에서 전역 데이터베이스는 중추적인 저장소 시스템 역할을 한다. 다른 컴퓨터 원용 응용 시스템들은 전역 데이터베이스에 의해 설계 및 생산 정보를 공유한다. 전역 데이터베이스는 제품 모델과 무결성 제약조건들을 포함하는 전역 스키마, 즉 통합 제품 정보모델을 가진다. 각각의 응용 시스템은 자신만의 지역 데이터베이스를 가지고 그 응용 시스템의 작업 데이터를 관리한다. 또한, 각각의 데이터베이스는 전역 데이터베이스의 부분 집합인 EXPRESS로 기술된 지역 스키마를 갖는다. 지역 스키마는 자신이 지원하는 응용 영역에 관련된 엔티티들과 제약조건들만을 포함한다. 본 논문의 주요 주제인 제약조건의 배치와 확인은 5장과 6장에서 자세히 다룬다.

PDM/CE 환경의 기본 구조는 그림 1과 같으며, 다양한 컴퓨터 원용 시스템들(예를 들어, CAD, CAM, CAE, CAPP, 그리고 MRP 시스템 등)이 포함될 수 있다. 여기서는 이러한 응용 시스템들 중 CAD

(Computer Aided Design), CAPP (Computer Aided Process Planning) 및 MRP (Material Requirements Planning) 시스템을 통합하는 프로토타입을 제시한다. 본 프로토타입에서 CAD 시스템에 의해 생성된 제품 데이터는 다른 응용 프로그램들이 검색하고 사용할 수 있도록 하기 위하여 전역 데이터베이스에 전달된다. 제품 데이터가 다수의 응용 프로그램들 간에 공

유되는 동안 기존의 데이터가 갱신될 수도 있고 새로운 데이터가 추가될 수도 있다. 전역 데이터베이스는 여러 응용 시스템들을 위한 공통 저장소의 역할을 하며, 또한 각 응용 프로그램에 의해 갱신되거나 생성된 데이터의 무결성을 확인하여 부당한 데이터의 삽입을 미연에 방지한다.



<그림 1> PDM/CE 환경의 구조

STEP 은 다양한 산업계 (예를 들어, 자동차, 조선, 기계 설계, 공정 계획, 그리고 FEM (Finite Element Method) 등) 간에 데이터를 공유하기 위한 정보모형을 정의할 뿐만 아니라, 물리적 파일, 작업 데이터 형태 및 데이터베이스 등과 같이 다른 양식으로 저장된 STEP 데이터를 액세스하기 위한 응용 프로그램 인터페이스인 SDAI (Standard Data Access Interface)를 정

의하고 있다. SDAI의 목적은 STEP 데이터를 액세스하기 위한 개방형 인터페이스를 구축하는 것이다. 예를 들어, SDAI를 이용함으로써 CAD 시스템은 DBMS의 특정한 명령을 전혀 모르더라도 이기종의 DBMS로부터 STEP 데이터를 검색하거나, DBMS로 STEP 데이터를 저장할 수 있다. 결과적으로 이러한 PDM 환경에서는 파일 시스템 또는 데이터베이스 시

시스템을 저장 시스템으로 사용할 수 있다. 개방형 인터페이스를 사용함으로써 급속한 하드웨어 및 소프트웨어 기술의 변화에 대해 생산업체의 CAD/CAM 시스템에 대한 투자를 보호할 수 있다.

4. STEP을 이용한 통합 제품 정보모델 (IPIM)

본 연구에서 개발한 통합 제품 정보모델은 EXPRESS로 정의되어 있는 STEP의 자원들을 이용하고 있다. 관련된 STEP 파트들은 파트 42 (Geometric and topological representation), 파트 47 (Shape variational tolerances), 그리고 파트 224 (Mechanical Product Definition for Process Planning using machining features) 등이며, 또한 이러한 파트들 외에 파트 41 (Fundamentals of product description and support)과 파트 43 (Representation structures) 등도 참조하고 있다.

파트 41은 제품의 묘사에 대한 기본적인 사항들, 즉 제품구성, 인가, 계약, 기간, 인사 조직 및 치수 단위 등의 기본 타입들을 제공한다. 파트 43은 제품 표현을 위한 전체적인 구조를 기술하고 있으며, 사용되는 엔티티들이 모임에 참여하도록 하는 표현을 할 수 있게 해준다. 예를 들어, 파트 42에서 제공되는 정보모델들을 사용하여 정의된 형상 요소는 특정한 좌표계에 놓여지지 않는 한 아무런 의미를 갖지 못한다.

파트 42는 다음과 같은 세 가지 스키마에서 형상 및 위상 표현을 제공한다.

- 1) *Geometry_schema* : point, vector, parametric curve, surface 등을 정의.

- 2) *Topology_schema* : vertex, edge, face, path, loop, shell 등을 정의.

- 3) *Geometric_model_schema* : solid model (CSG 및 B-rep), surface model, wireframe model 등을 정의.

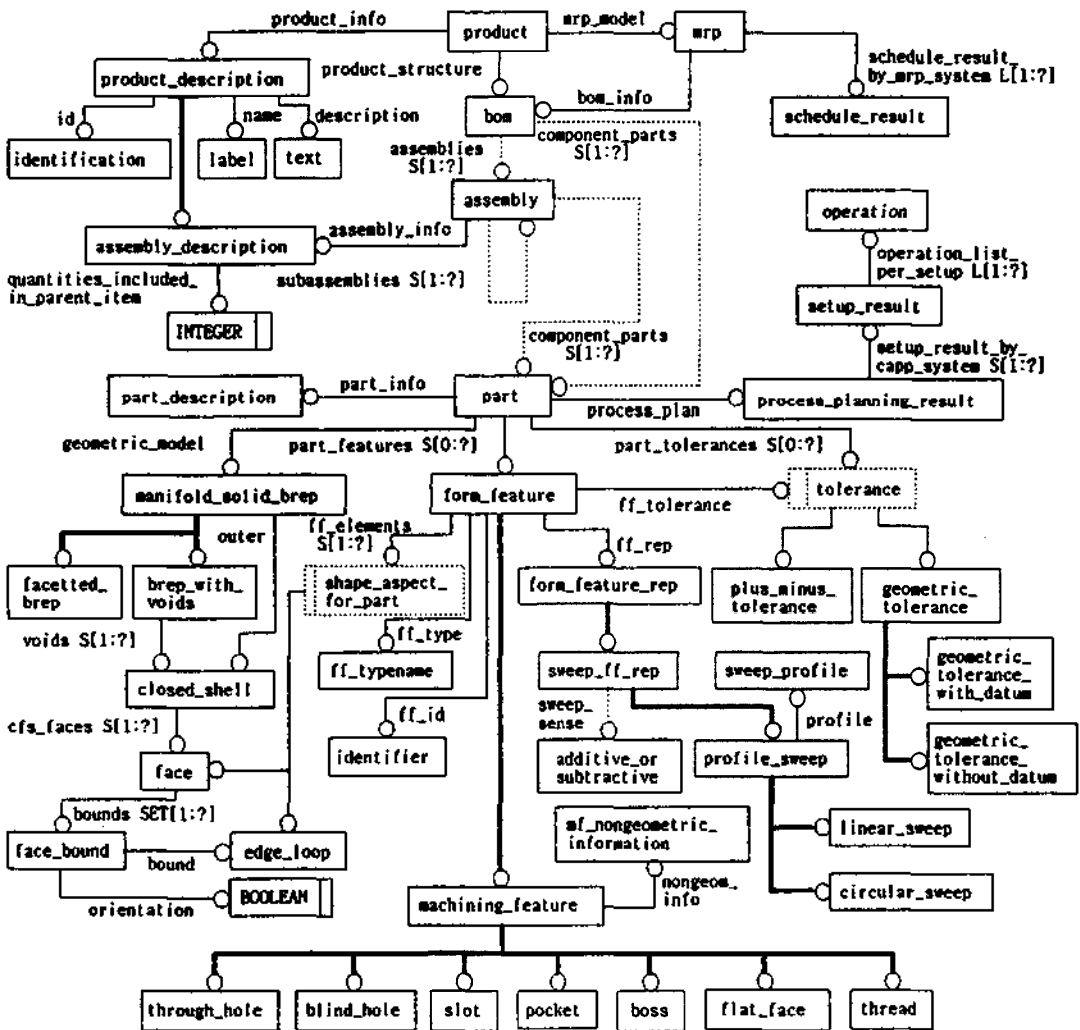
파트 47에 정의되어 있는 공차들은 크게 *geometric tolerance*와 *plus-minus tolerance*로 분류될 수 있다. *Geometric tolerance*는 제품의 기하학적 특성에 대한 허용오차 범위를 정의하며, 하위 타입으로 *geometric_tolerance_with_datum_reference*와 *geometric_tolerance_with_defined_unit*를 갖는다. 또한 속성으로서 *name*, *magnitude*, *toleranced_shape_aspect*, *zone_elements* 등을 가지며, *toleranced_shape_aspect*는 공차가 적용되는 대상을 표현한다. *Plus-minus tolerance*는 주어진 치수에 대한 허용오차의 범위를 정의하며, 속성으로는 *toleranced_dimension*와 *range*를 갖는다.

파트 224는 기계 가공 부품의 제조를 위해 필요한 제품 데이터를 정의하는데 요구되는 정보의 표현과 교환을 위한 정보 요구조건 및 그 범위를 정의하고 있으며, 이러한 요구조건들을 만족시키기 위해 필요한 통합 자원들을 기술하고 있다. 즉, 파트 224에서의 제품데이터 표현은 제품 형상을 기존의 부품 설계에 바탕을 둔 가공 형상특징 (*machining feature*)에 의하여 표현하고 있으며, 정보 교환을 위한 요구조건들은 공정 계획을 위한 부품의 정의에 필요한 부품 식별 데이터, 형상 정의 및 형상 표현 데이터, 그리고 재료 데이터 등을 기술하고 있다.

통합 제품 정보모델에서는 처리 대상물인 기계 제품을 *product*로 표현하며, *product*는 그 속성으로 *product_info*, *mrp_model*, 그리고

product_structure 를 가진다. 그림 2는 product 와 관련된 최상위 단계의 정보 구조만을 EXPRESS 의 그래픽 표현 방법인 EXPRESS-G [6]를 이용하여 표현한 것이다. EXPRESS-G에서, 박스는 엔티티를

나타내고, 가는 실선은 엔티티와 그 속성을 연결 하며, 이 경우 연결되는 속성에는 작은 원이 덧붙는다. 굵은 실선은 상위 및 하위 타입의 관계를 나타내며, 하위 타입에는 작은 원이 덧붙는다.



<그림 2> 통합 제품 정보모델의 상위 단계 표현

속성 `product_info`는 제품의 전반적인 특성 (예를 들어, 식별자, 제품명, 제품 설명 등)에 관한 정보를 표현하기 위하여 사용하며, 이 속성은 엔티티 `product_description`을 참조한다. 속성 `mrp_model`은 제품의 MRP 정보를 표현하기 위한 속성이며 엔티티 MRP를 참조한다. 속성 `product_structure`는 제품 구조에 관한 정보를 표현하기 위한 속성이며 엔티티 BOM을 참조한다.

엔티티 BOM은 그 속성으로 `assemblies`와 `component_parts`를 선택적으로 가진다. 속성 `assemblies`는 엔티티 `assembly`의 집합으로 표현되며, 속성 `component_parts`는 `part`의 집합으로 표현된다. 이 속성들을 선택형으로 정의한 이유는 제품의 복잡성 여부에 따라 제품 구조의 표현이 상당히 복잡해지기 때문이다. 엔티티 `assembly`는 속성으로 `subassemblies`와 `component_parts`를 선택형으로 가지며, `assembly`의 전반적인 특성을 표현하기 위하여 속성 `assembly_info`를 가진다. 속성 `subassemblies`는 계귀적으로 엔티티 `assembly`의 집합으로 표현되며, 속성 `component_parts`는 엔티티 `part`의 집합으로 표현된다. 속성 `assembly_info`는 엔티티 `product_description`의 서브 타입인 엔티티 `assembly_description`을 참조하며, 따라서 이 엔티티는 식별자를 표현하기 위한 속성인 `id`, `assembly` 이름을 표현하기 위한 속성인 `name`, `assembly`의 설명을 나타내기 위한 속성인 `description`을 엔티티 `product_description`으로부터 상속 받고 `assembly`의 상위 단계에 있는 품목, 즉 `product`에 포함되는 `assembly`의 개수를 표현하기 위하여 자신만의 속성인

`quantities_included_in_superitem`을 그 속성으로 가진다.

`Product` 구조에서 최하위 단계인 엔티티 `part`는 속성으로 `part_info`, `geometric_model`, `part_features`, `part_tolerances`, 그리고 `process_plan`을 가진다. 속성 `part_info`는 `part`의 전반적인 특성에 관한 정보를 표현하는 엔티티인 `part_description`을 참조하며, 이 엔티티는 속성으로 도면에 나타나는 표제란 정보를 나타내기 위한 `title_info`와 부품의 관리 정보를 나타내기 위한 `management_info`를 가진다.

`Part`의 형상 (Geometry) 및 위상 (Topology) 정보는 STEP의 파트 42에 정의되어 있는 `solid model` 중에서 `manifold_solid_brep`을 이용하였다. 이는 여러 가지 `solid model` 중에서 B-rep (Boundary representation)을 사용하는 것이 `part`의 형상과 그것에 연관된 형상 특징 및 공차 사이의 관계를 표현하는데 있어서 효율적이기 때문이다.

엔티티 `part`는 다수의 형상 특징을 가질 수 있으며, 이러한 형상 특징들은 엔티티 `form_feature`로 표현되고, 속성 `part_features`에 의해 참조된다. 엔티티 `form_feature`는 특별히 가공 형상 특징을 나타내기 위하여 `machining_feature`를 서브 타입으로 가지며, 형상 특징의 타입을 나타내기 위하여 속성 `ff_type`을 가진다. 형상 특징은 다수의 `face` 또는 `edge_loop`들로 구성되는데 이러한 `face`나 `edge_loop` 정보는 속성 `ff_elements`에 의해 참조되며, 엔티티 `shape_aspect_for_part`를 통하여 형상 정보모델과 연결되도록 하였다. 여기서 정의

한 엔티티 `shape_aspect_for_part` 는 파트 43 이나 파트 224 의 `shape_aspect` 에 해당되지만 거기에서는 매우 포괄적으로 정의하고 있어 다소 이해하기 어렵게 할 여지가 있다. 따라서 본 정보모델에서는 응용의 문맥상 명백한 이해와 적용을 위하여 `face` 와 `edge_loop` 로 그 범위를 제한하였다. 이러한 엔티티 범위의 정의 문제는 필요에 따라서 확장 또는 축소함으로써 모호성을 제거할 수 있을 것이다. 형상 특징 사이의 공차 관계는 속성 `ff_tolerance` 를 사용하여 표현하는데 이것은 선택형 타입인 `tolerance` 를 통해 파트 47 에 정의되어 있는 `plus_minus_tolerance` 또는 `geometric_tolerance` 를 참조하도록 하였다. 그리고 형상 특징의 구체적인 형상을 표현하기 위하여 속성 `ff_rep` 를 가지며, 이 속성은 엔티티 `form_feature_rep` 를 참조한다. 그리고 가공 형상 특징 즉, `machining_feature` 는 비기하학적 정보(예를 들어, 면취, 가공 여유, 표면 조도 등)를 표현하기 위하여 속성 `nongeom_info` 를 가진다.

`Part` 는 다수의 공차들을 가질 수 있으며, 각 공차는 STEP 의 파트 47 에 정의되어 있는 `geometric_tolerance` 와 `plus_minus_tolerance` 중 하나가 된다. 파트 47 에서는 공차의 대상을 `shape_aspect` 로 규정하고 있으나, 여기서는 새로운 엔티티인 `shape_aspect_for_part` 를 구성하고 이 엔티티를 통하여 형상 정보모델인 `manifold_solid_brep` 의 `face` 나 `edge_loop` 엔티티를 참조함으로써 공차 정보와 형상 정보가 연결되도록 하였다.

`Part` 의 공정계획 정보모델인 엔티티 `process_plan` 은 `setup_result_by_capp_system` 을

속성으로 가진다. 이 속성은 접속하는 CAPP 시스템에 의한 셋업 결과를 표현하기 위한 속성이며 엔티티 `setup_result` 의 집합으로 표현된다.

`Product` 와 MRP 정보모델인 엔티티 MRP 는 접속하는 MRP 시스템의 입력으로 사용되는 BOM 정보를 표현하기 위한 속성인 `bom_info` 와 MRP 시스템에 의한 스케줄링 결과에 관한 정보를 표현하기 위한 속성인 `schedule_result_by_mrp_system` 을 그 속성으로 가진다.

5. EXPRESS 의 무결성 제약조건

EXPRESS 는 무결성 제약조건을 정의할 수 있는 강력한 표현력을 가지고 있어 일반적인 알고리즘을 포함할 수 있으나, 표현이 강해진 만큼 요구되는 계산도 많아진다. 무결성 제약조건에 포함되는 알고리즘들은 개별적으로 실행 가능한 함수들로 변환되며, 이러한 함수들은 무결성 제약조건들이 확인될 동안에 필요할 경우 호출될 수 있다. EXPRESS 에서는 유일성 제약조건, 지역 제약조건, 존재성 제약조건, 그리고 전역 제약조건 등과 같은 네 종류의 무결성 제약조건들이 있다.

5.1 유일성 제약조건

UNIQUE 절은 엔티티 타입의 단 하나의 속성 또는 여러 속성들의 집합에 대하여 유일성 제약조건을 선언한다. 만일 UNIQUE 가 단 하나의 속성에 대하여 정의된다면, 두개 이상의 엔티티 인스턴스들은 그

속성에 대하여 동일한 값을 가질 수 없으며, 여러 속성들의 집합에 대한 UNIQUE 정의는 명명된 속성들에 대한 동일한 값들의 조합을 제한한다. 유일성 제약조건들의 확인은 지역 데이터베이스에서 행해질 수 있으며, 만일 동일 엔티티에 대한 인스턴스들이 다른 응용 시스템들에 의해 생성된다면, 전역 데이터베이스에서도 확인되어야 한다.

예제 1. 엔티티 *part_management_information* 은 part의 식별자, 이름, 설명, 그리고 assembly에 포함된 수량과 같은 전반적인 관리 정보를 포함한다. 다음의 엔티티 정의에 포함되어 있는 UNIQUE 절의 *url*은 *id*가 유일성을 지닌 속성임을 기술하고 있으며, 엔티티 *part_management_information*에 대한 모든 인스턴스들은 동일한 식별자를 가질 수 없음을 선언하고 있다.

```
ENTITY part_management_information:
  id : identifier;
  name: label;
  description: text;
  quantities_included_in_superit
em: INTEGER;
UNIQUE
  url: id;
END_ENTITY;
```

5.2 지역 제약조건

WHERE 절은 엔티티 타입의 각 인스턴스에

적용되는 지역 제약조건들을 정의하며, 지역 제약조건들은 하나의 속성 또는 여러 속성들의 조합에 대한 유효한 값을 기술한다. 이러한 제약조건들이 단 하나의 인스턴스에만 적용되기 때문에, 모든 지역 제약조건들은 지역 데이터베이스에서 확인될 수 있다.

예제 2. 이 예제의 WHERE 절은 *through_hole*의 형상 특징의 구성 요소들의 개수가 세 개 (즉, 두개의 *edge_loop*와 하나의 *surface_of_revolution*)이고, 그 중 하나는 *face* 이어야 함을 기술하고 있다.

```
ENTITY through_hole:
SUBTYPE OF (machining_feature);
WHERE
  wr1 : SIZEOF (SELF.
ff_elements) = 3;
  wr2 : SIZEOF (QUERY(temp <=
ff_elements | TYPEOF (temp) =
['face'])) = 1;
END_ENTITY;
```

5.3 존재성 제약조건

존재성 제약조건은 한 엔티티의 인스턴스 존재 여부가 다른 관련된 엔티티의 인스턴스 존재에 의존함을 선언한다. 만일 다른 엔티티가 명시적인 속성에 의하여 현재 엔티티와의 관계를 설정하였다면, 역 (inverse) 속성은 현재 엔티티의 문맥에 따라 그 관계를 기술하기 위하여 사용될 수 있다. 존재성 제약조건들의 확인은 관련된 두 엔티티가 동일한 지역 스

키마에 정의되어 있는지 또는 그렇지 않은지에 따라 달라진다. 만일 두 엔티티가 동일한 지역 스키마에 정의되어 있다면, 이 제약조건은 그 스키마가 포함된 데이터베이스에서 확인될 수 있고, 그렇지 않을 경우에는 전역 데이터베이스에서 확인되어야 한다.

Example 3. 이 예제는 두 엔티티 *bolt* 와 *nut* 를 포함하고 있으며, 엔티티 *bolt* 는 속성으로 대응하는 *nut* 를 기술하기 위하여 *fastener* 를 갖고 엔티티 *nut* 는 대응하는 *bolt* 를 기술하기 위하여 *fasten* 을 속성으로 갖는다. 데이터베이스의 무결성이 적절히 유지되도록 하기 위하여 서로 대응하는 *bolt* 와 *nut* 의 인스턴스 쌍이 데이터베이스 내에 존재해야만 한다.

```
ENTITY bolt:
  SUBTYPE OF (part):
    size: bolt_size_type;
    type: bolt_type;
    fastener: nut;
END_ENTITY;

ENTITY nut:
  SUBTYPE OF (part):
    size: nut_size_type;
    type: nut_type;
  INVERSE
    fasten: bolt FOR fastener;
END_ENTITY;
```

5.4 전역 규칙

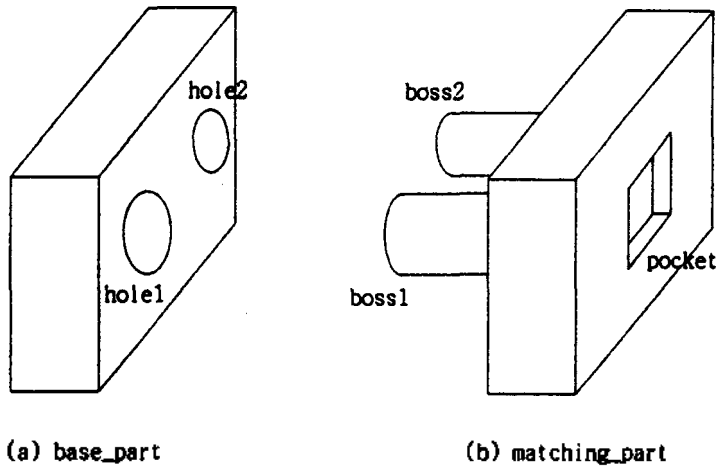
전역규칙은 EXPRESS 의 RULE 로 정의될 수 있으며, 하나의 엔티티 타입 또는 여러 엔티티 타입의 인스턴스 집합 간의 제약조건들을 기술한다. RULE 은 실행 가능한 문장들과 실행의 결과에 의한 데이터의 유효성을 결정하는 WHERE 절로 구성된다. RULE 의 선언에는 규칙에 의해 참조되는 엔티티들의 리스트가 포함되며, RULE 의 확인은 참조되는 엔티티들이 선언된 장소에 의존한다. 즉, 참조되는 엔티티들이 모두 하나의 지역 스키마에 선언되어 있다면, 상용하는 지역 데이터베이스에서 확인될 수 있고, 그렇지 않을 경우는 전역 데이터베이스에서 확인되어야 한다.

예제 4. 이 예제에서는 그림 3 과 같은 두 개의 part, 즉 *base_part* 와 *matching_part* 에 대하여 고려한다. 성공적으로 조립되도록 하기 위하여 *base_part* 에 있는 hole 들과 *matching_part* 에 있는 boss 들은 크기가 서로 맞아야 한다. 이 예제에서 *rule1* 은 *base_part* 에 있는 *hole1* (또는 *hole2*)의 최소 크기가 *matching_part* 에 있는 *boss1* (또는 *boss2*)의 최대 크기보다 적어도 더 커야 함을 기술하고 있다. 이러한 제약조건은 세 엔티티, 즉 *base_part*, *matching_part*, 그리고 *assembly* 에 대하여 기술되며, 엔티티 *assembly* 에 의하여 함께 조립되도록 정의된 *base_part* 와 *matching_part* 의 쌍에 대하여 확인된다.

```

RULE rule1 FOR (base_part, matching_part, assembly):
LOCAL
    part1: base_part;
    part2: matching_part;
    h1, h2: through_hole;
    b1, b2: boss;
END_LOCAL;
part1 = QUERY (temp <* assembly | TYPEOF (temp.component_parts)
               = ['base_part']);
part2 = QUERY (temp <* assembly | TYPEOF (temp.component_parts)
               = ['matching_part']);
IF (EXISTS(part1)) THEN
    h1 = QUERY (temp <* part1.part_features | temp.ff_id = 'hole1');
    h2 = QUERY (temp <* part1.part_features | temp.ff_id = 'hole2');
END_IF;
IF (EXISTS(part2)) THEN
    b1 = QUERY (temp <* part1.part_features | temp.ff_id = 'boss1');
    b2 = QUERY (temp <* part1.part_features | temp.ff_id = 'boss2');
END_IF;
WHERE
wr1: IF (EXISTS(part1) AND EXISTS(part2)) THEN
    ((h1.ff_rep.profile.diameter - h1.ff_tolerance.range.lower_bound) >
     (b1.ff_rep.profile.diameter + b1.ff_tolerance.range.upper_bound))
    AND
    ((h2.ff_rep.profile.diameter - h2.ff_tolerance.range.lower_bound) >
     (b2.ff_rep.profile.diameter + b2.ff_tolerance.range.upper_bound))
END_IF;
END_RULE;

```



<그림 3> 엔티티 *base_part* 와 *matching_part* 의 개념도

5.5 제약조건의 상속

엔티티 타입에 대하여 정의된 제약조건들은 그 엔티티의 하위 타입에 상속된다. 지역 및 전역 제약조건들은 하위 타입에서 다시 정의될 수 없지만, 새로 추가적인 제약조건들은 상속된 속성들에 추가될 수 있다. 이러한 상속 매커니즘으로 위에서 설명된 것과 동일한 확인 방법들이 상속된 모든 속성들에 대해서도 적용된다.

6. 무결성 제약조건의 확인

앞장에서 논의된 바와 같이 EXPRESS에는 네 가지 타입의 무결성 제약조건들이 있다. 각 타입의 제약조건은 다른 의미 및 사용법을 가지고 있기 때문에, 전체 시스템의 일관성과 효율성을 유지하기 위하여 전체적인 확인 시스템은 이를 고려하여야 한다.

6.1 제약조건의 계층구조

본 연구에서 설계된 PDM/CE 환경에서는 다른 응용 시스템들이 전역 데이터베이스를 통하여 제품 데이터를 공유한다. 각 응용 시스템들은 그 자신의 지역 데이터베이스 시스템을 소유하고 그 안에서 자신의 작업 데이터를 관리한다. 일단 응용 시스템이 작업 데이터를 마무리 저으면, 그 데이터는 전역 데이터베이스로 전달되며, 다른 응용 시스템들은 전역 데이터베이스를 통하여 그 전달된 데이터를 사용할 수 있다. 이러한 구성 관리 전형에서는 응용 시스템으로부터 전달되지 않는 어떤 사전 준비 데이터도 PDM/CE 환경에 있는 다른 응용 시스템들에 의해 참조될 수 없다. 이러한 환경에서 설계 데이터는 우선 지역 데이터베이스들로 삽입되고, 상용하는 지역 스키마에 포함된 무결성 제약조건들로 확인된다. 따라서 지역 스키마는 단지 지역 데이터베이스에 저장된 데이터들과 함께 확인될 무결성 제약조건들만 포함

하며, 예로 WHERE 절로 표현되는 지역 제약조건을 들 수 있다. 즉, 엔티티의 인스턴스가 생성되거나 갱신되는 순간에 WHERE 절은 그 인스턴스의 유효성을 보장하기 위하여 검사되어야 할 수 있다.

한편, 전역 데이터베이스에 있는 전역 스키마는 지역 스키마들과는 달리 모든 제약조건들을 포함하며, 전역 데이터베이스에 저장되어 있는 제약조건들은 응용 시스템들로부터 전달된 데이터의 유효성을 검사한다. 다른 응용 시스템들로부터의 몇몇 엔티티들에 대한 전역 규칙들은 전역 데이터베이스에서 확인된다. 예제 4의 제약조건은 CAD 시스템으로 설계한 part들에 포함된 형상 특징들의 최소 및 최대 크기를 지정하고 있다. 그러나, 이 규칙은 두 part의 모든 인스턴스들의 모든 가능한 조합에 대한 관계는 검사하지 않는다. 이러한 관계는 단지 part들의 어떤 두 인스턴스들이 함께 조립되도록 계획된 경우에만 의미를 가지며, 조립 정보는 MRP 시스템에 의해 생성된다.

일반적으로 무결성 제약조건은 하나 이상의 스키마에 포함될 수 있다. 우선 무결성 제약조건이 다수의 지역 스키마들에 포함될 수 있으며, 이 경우는 엔티티가 다수의 지역 스키마들에서 정의될 때 발생한다. 예를 들어, PDM/CE 환경에서는 여러 개의 CAD 시스템들이 있을 수 있고, 동일한 엔티티에 대하여 다른 인스턴스들이 다른 시스템들에 의해 설계될 수 있다. 이런 경우에 지역 규칙은 여러 개의 지역 스키마들에 포함될 수 있으며, 인스턴스들이 생성되거나 갱신되는 순간 확인될 수 있다. 다음으로 제약조건은 지역 스키마와 전역 스키마 양쪽에 포함될 수 있다. 예를 들어, 만일 엔티티의 인스턴스들이 두 응용 시스템으로부터 생성되고 유일성 제약조건이 그

엔티티에 대해 정의되어 있다면, 그 유일성 제약조건은 지역 데이터베이스뿐만 아니라 전역 데이터베이스에서도 검사되어야 한다.

위에서 논의된 일반적인 지침을 따라 무결성 제약조건들의 계층구조를 다음과 같이 요약할 수 있다.

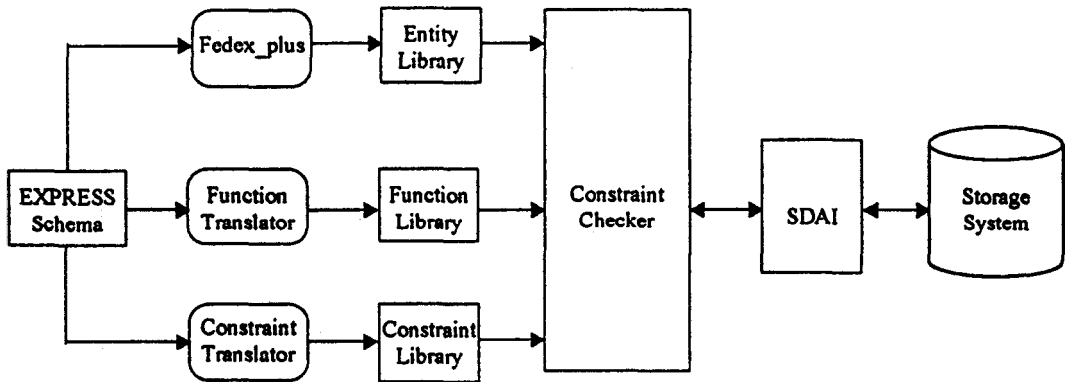
- 1) 유일성 제약조건: 유일성 제약조건은 상용하는 엔티티가 정의되어 있는 지역 스키마에 포함된다. 하나 이상의 지역 스키마가 동일한 엔티티를 정의할 경우, 유일성 제약조건은 전역 스키마에도 포함되어야 한다.
- 2) 지역 제약조건: 지역 제약조건은 상용하는 엔티티가 정의되어 있는 지역 스키마에 포함된다. 지역 제약조건은 단지 엔티티의 개별 인스턴스만을 고려하기 때문에, 전역 스키마에 포함될 필요는 없다.
- 3) 존재성 제약조건: 만일 INVERSE 절에 의하여 관련된 두 엔티티가 하나의 지역 스키마에 정의되어 있다면, 존재성 제약조건이 지역 스키마에 포함되는 것으로 충분하지만, 그렇지 않을 경우는 전역 스키마에 포함되어야 한다.
- 4) 전역 규칙: 만일 전역 규칙에 의해 관련된 모든 엔티티들이 하나의 지역 스키마에 정의되어 있다면, 전역 규칙은 상용하는 지역 데이터베이스에 포함되고 확인될 수 있다. 그렇지 않을 경우는 전역 스키마에 포함되어야 한다.

6.2 제약조건 확인을 위한 소프트웨어 구조

EXPRESS 스키마로 정의된 무결성 제약조건들을 확인하기 위하여 제약조건들은 적절한 형태로

변환될 필요가 있다. 그림 4는 무결성 제약조건 확인에 관련된 소프트웨어 모듈들 사이의 관계를 나타내는 것이다. 첫번째로, 제약조건들에 포함된 모든 함수들은 프로그래밍 언어 (예를 들어, 본 프로토타입에서 사용된 C와 같은 언어들)로 변환되어야 하고, 그 다음 변환된 함수들은 컴파일된 후 함수 라이브러리로 구축되어야 한다. 두 번째로, 네 가지 타입의 무결성 제약조건들 역시 제약조건 라이브러리를 구축

하기 위하여 프로그래밍 언어로 변환될 필요가 있다. 마지막으로, 스키마에 정의된 엔티티 구조와 제약조건들은 분석된 후 제약조건 트리거 모듈로 컴파일되어야 한다. 각 라이브러리들 및 제약조건 트리거 모듈과 함께 링크되는 제약조건 검사기 (Constraint Checker)는 저장 시스템에 저장되어 있는 제품 데이터들을 액세스하고 그 데이터들을 확인한다.



<그림 4> 무결성 제약조건 확인을 위한 소프트웨어 모듈들 간의 관계

EXPRESS에는 두 타입의 함수들이 있는데, 하나는 내장 함수이고 다른 하나는 사용자 정의 함수이다. 내장 함수들 (예를 들어, $\sin()$, $\cos()$, and $\text{sizeof}()$ 등)에 대해서는 C 라이브러리 함수들이나 사용자 정의 C 함수들이 사용되고, 사용자 정의 EXPRESS 함수들은 C 함수들로 변환된다. EXPRESS의 기본 구조와 C의 기본 구조가 유사하며, 또한 EXPRESS의 키워드들과 C의 키워드들은 상용하기 때문에 함수 변환이 상대적으로 간단하다. 두 언어 사이의 차이점은 중첩 함수의 지원이다. 즉 EXPRESS는 함수들이 다른 함수의 내부에 정의되는 것을 허용

하지만, C에서는 허용하지 않는다. 이러한 차이점을 다루기 위하여 EXPRESS의 중첩 함수들은 C에서는 별개의 함수들로 변환된다. 각각의 무결성 제약조건들은 C에서 실행 가능한 모듈로 변환된다. 현재는 이러한 실행 가능한 모듈들이 SDAI를 통하여 저장 시스템에 저장되어 있는 제품 데이터를 액세스한다.

6.3 확인 트리거 메커니즘

PDM/CE 환경에서 무결성 제약조건들은 두 가지의 다른 기본적인 접근 방법, 즉 온라인식 확인

방법과 일괄처리식 확인 방법을 사용하여 확인될 수 있다. 온라인식 확인 방법에서는 각각의 인스턴스가 삽입되거나 갱신되기 전에 무결성이 미리 확인되고, 반면에 일괄처리식 확인 방법에서는 인스턴스가 여러 번 삽입 또는 갱신된 후에 무결성 확인 절차가 실행된다. 한편, 온라인식 확인 방법은 데이터베이스 시스템들과의 상호작용을 느리게 하지만 제품 데이터가 잘못된 상태가 되는 것을 줄여주는 반면에, 일괄처리식 확인 방법은 한밤중이나 주말과 같은 시스템 유휴 시간을 이용하지만 다음 확인 때까지 데이터베이스 시스템 내에 부적절한 제품 데이터를 내버려둘 수도 있다.

온라인식 확인과 일괄처리식 확인의 두 경우 모두에 있어 모든 삽입들이나 갱신들이 확인될 필요는 없기 때문에, 어떤 트리거 메커니즘이 필요하다. 새로운 인스턴스가 삽입되거나 기존의 인스턴스가 갱신될 적마다 어떤 제약조건들의 확인이 필요한지 결정할 필요가 있다. 이러한 결정을 위하여 각 제약조건들의 확인에 포함된 엔티티들과 속성들을 알아야 한다. 인스턴스들과 제약조건들 간의 관계를 고려해 보면, 트리거 의존성들이 존재함을 알 수 있다. 만일 제약조건이 엔티티의 속성 값들을 평가하면 엔티티와 제약조건 사이에 트리거 의존성이 존재한다고 정의하고, 어떤 조건이 위반되고 있는지를 검사한다.

PDM/CE 환경에서는 삽입, 삭제, 갱신과 같은 세 가지의 주요한 데이터 연산이 있다. 엔티티의 인스턴스가 삽입되거나 삭제될 때, 그 인스턴스에 트리거 의존성을 가지고 있는 제약조건들은 확인될 필요가 있으며, 갱신에 대해서는 다른 접근 방법이 필요하다. 즉, 갱신은 인스턴스의 속성 값을 변경하기 때문에, 만일 그 속성이 제약조건에 의해 사용되지 않

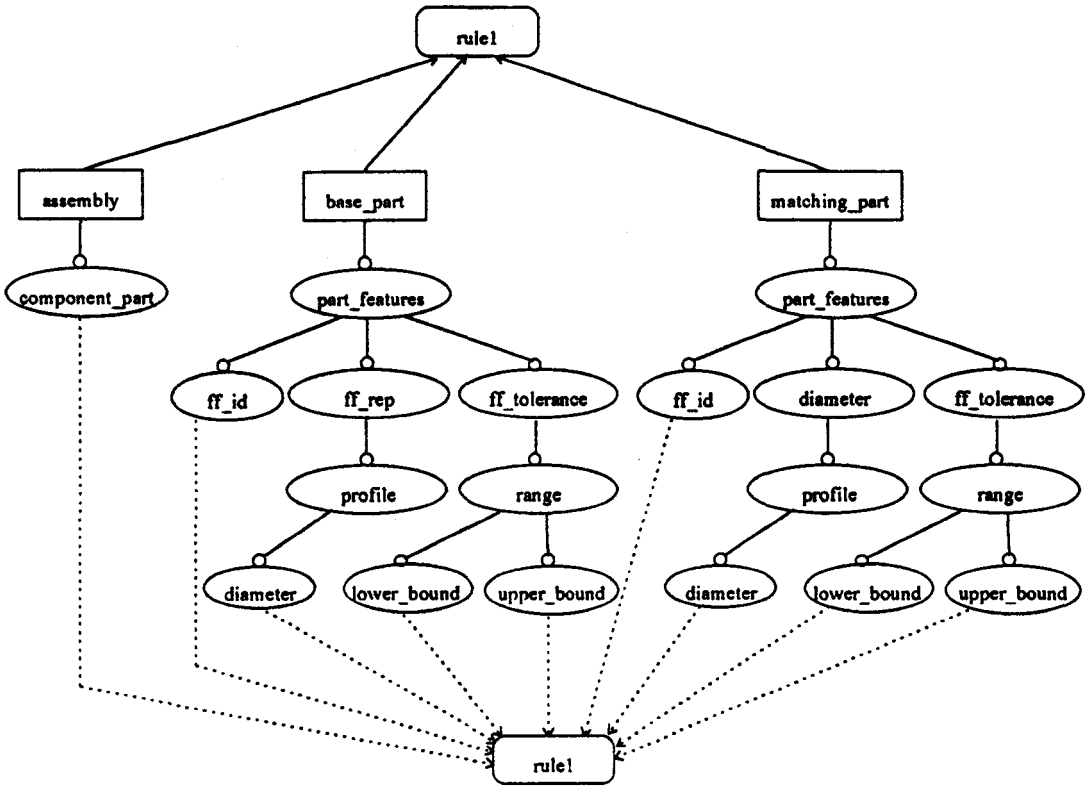
는다면 트리거에 의존하는 제약조건을 평가할 필요가 없다. 이러한 차이점을 반영하기 위하여, 삽입 의존성, 삭제 의존성, 그리고 갱신 의존성과 같은 세 종류의 트리거 의존성을 정의한다. 각 의존성은 상용하는 데이터 연산들에 대한 확인 과정을 활성화한다. 예를 들면, 삽입 의존성은 단지 삽입 연산에 대해서만 확인 과정을 활성화한다. 이러한 의존성 관계들은 주어진 EXPRESS 스키마로부터 분석되어지고, 그림 4와 같이 제약조건 라이브러리로 컴파일되어야 한다. 각 제약조건에 대한 트리거 의존성은 다음과 같이 요약된다.

- 1) 유일성 제약조건: 유일성 제약조건은 제약조건이 정의된 엔티티 상의 삽입에 의존하며, 유일하다고 정의된 속성 상의 갱신에 의존한다. 유일성 제약조건에 대한 삭제 의존성은 전혀 없다.
- 2) 지역 제약조건: 지역 제약조건은 제약조건이 정의된 엔티티 상의 삽입에 의존하며, 제약조건에 의해 평가되는 속성들 상의 갱신에 의존한다. 지역 제약조건에 대한 삭제 의존성은 갖지 않는다.
- 3) 존재성 제약조건: 존재성 제약조건은 제약조건이 정의된 엔티티 상의 삽입에 의존하며, 역(inverse) 속성에 의해 참조되는 엔티티 상의 삭제에 의존한다. 역 속성과 역 속성에 의해 참조되는 속성 양쪽 상의 갱신에 의존한다.
- 4) 전역 규칙: 전역 규칙은 규칙으로 평가되는 속성들을 가지고 있는 엔티티들 상의 삽입 및 삭제 양쪽 다에 의존한다. 또한, 규칙에 의하여 평가되는 속성들 상의 갱신에 의존적이다.

예제 5. 예제 4의 전역 규칙을 위한 트리거 의

존성들은 그림 5와 같이 나타낼 수 있다. 그림 5에서 사각형은 엔티티를, 타원형은 속성을, 그리고 동근 모서리의 사각형은 제약조건을 나타내며, 실선 화살

표는 삽입 및 삭제 의존성을 나타내고 점선 화살표는 갱신 의존성을 나타낸다.



<그림 5> 예제 4의 rule1에 대한 트리거 의존성

7. 결론

본 논문에서는 STEP을 이용하여 구현한 PDM/CE 환경을 제시하였다. 동시에 발생하는 공학 활동들을 지원하기 위하여, 이러한 환경에서는 데이터 저장 시스템들을 지역 및 전역 시스템으로 분류한다. 지역 데이터베이스에서 관리되는 작업 데이터는 그 데이터들이 전역 데이터베이스로 전달될 때까지 다른 응용 시스템들에 의하여 액세스될 수 없다. EXPRESS로 정의된 무결성 제약조건들은 지역 데이터베이스들 및 전역 데이터베이스 양쪽에서 확인된다. 제품 데이터의 무결성을 유지함으로써 부적절한 데이터가 다른 응용 시스템으로 바람직하지 않게 전파되는 것을 미연에 방지할 수 있다. 효율적인 확인을 위하여 제약조건들은 지역 및 전역 스키마들로 분산된다. 주요한 새 가지의 다른 연산들, 즉 삽입, 삭제, 그리고 갱신에 대하여 별개의 트리거 메커니즘들이 제약조건들의 의존성을 이용하여 고안되었다.

또한, 통합 시스템의 프로토타입도 제시하였으며, 통합 제품 정보모델은 STEP의 공통 자원들과 응용 프로토콜들을 이용하여 개발하였다. STEP을 이용한 시스템 통합과 EXPRESS 제약조건들의 확인에 대한 많은 연구 프로토타입이 발표되고 있지만, 본 논문에서는 계층구조로 된 저장 시스템들, 즉 지역 및 전역 데이터베이스들에 저장된 제품 데이터의 무결성 확인 방법을 제시하였다. 현재의 프로토타입에서는 응용 시스템들이 SDAI를 통하여 데이터베이스들을 액세스하기 때문에, 다중 사용자 액세스를 지원하지 않는다. 현재의 확인 절차는 인덱스 파일, 포인터 또는 활성 데이터베이스와 같은 저장 시스템들의 시스템 의존적인 특징들을 이용하여 더 최적화될 수 있으며, WWW (World Wide Web)와 CORBA (Common Object Request Broker Architecture) [OMG, 1996] 등의 산업 표준을 활용하여 분산 시스템에서의 온라인 처리기능을 추가할 수 있다.

참고 문헌

- [도남철 외 2, 1993] 도남철, 최인준, 김광수, "객체지향적 데이터베이스 시스템에서의 PDES/STEP의 구현," *산업공학회 93 추계학술발표회 논문집*, 1993년 10월, pp. 526 - 534.
- [서효원, 유상봉, 1995] 서효원, 유상봉, "STEP을 이용한 통합정보모델(IPIM) 개발," *대한산업공학회지*, 제 21권 제3호, 1995년 9월, pp. 441 - 461.
- [Alt, 1994] Jochen Alt, "Optimizing EXPRESS rule evaluation with file-grained dynamic materialization in CAD applications," *Proc. Fourth International EXPRESS User Group Conference, Meeting Associates, Clifton Park, N.Y.*, 1994.
- [Blakeley 외 2, 1989] J. Blakeley, N. Coburn, and P. Larson, "Updating Derived Relations: Detecting Irrelevant and Autonomously Computable Updates." *ACM Trans. on Database Systems*, Vol. 14, No. 3, 1989, 369 - 400.
- [Blleloch, 1986] G. Blleloch, "CIS: a massively concurrent rule-based system," *Proc. Fifth Nat'l Conf. Artificial Intelligence, AAAI*, 1986, 735 - 741.
- [Cacace 외 4, 1990] F. Cacace, S. Ccri, S. Crespi-Reghezzi, L. Tanca, and R. Zicari, "Integrating Object-Oriented Data Modeling with a Rule-Based Programming Paradigm," *Proc. ACM-SIGMOD 1990 Int'l Conf. on Management of Data*, 1990, 225 - 236.
- [Carter, Baker, 1992] D. Carter and B Baker, *Concurrent Engineering: The Product Development Environment for the 1990s*, Addison-Wesley publishing company, Reading, Massachusetts, 1992.
- [Carver, Bloom, 1991] G. Carver and H. Bloom, *Concurrent Engineering Through Product Data Standards*, National PDES Testbed Report Series #NISTIR 4573, NIST, Gaithersburg, MD, 1991.
- [CIMdata, 1994] CIMdata, *Product Data Management: A Technology Guide*, 1994.
- [Divoux, 1994] T. Divoux, et. al., "Using EXPRESS: to Define MMS Manufacturing Communication," in P. Wilson, editor, *Proc. Fourth International EXPRESS User Group Conference, Meeting Associates, Clifton Park, N.Y.*, 1994.

- [Eberl, 1994] M. Eberl, "Integrated system for furniture manufacturing," *Proc. of International Conference on Data and Knowledge Systems for Manufacturing and Engineering*, Hong Kong, May, 1994, pp. 383-388.
- [Gupta, Widom, 1993] A. Gupta and J. Widom, "Local Verification of Global Integrity Constraints in Distributed Databases," *Proc. ACM-SIGMOD 1993 Int'l Conf. on Management of Data*, 1993, 49 - 58.
- [Hardwick 외 3, 1995] M. Hardwick, B. Downie, M. Kutcher, and D. Spooner, "Concurrent Engineering with Delta Files," *IEEE Computer Graphics and Applications*, pp. 62 - 68, January 1995.
- [Hardwick 외 3, 1996] Martin Hardwick, David Spooner, Tom Rando and K. C. Morris, "Sharing Manufacturing Information in Virtual Enterprises," *Communications of the ACM*, vol 39, no. 2, pages 46-54, February 1996.
- [Henschen 외 2, 1984] Henschen, W. McCune, and S. Naqvi, "Compiling Constraint-Checking Programs from First-Order Formulas," *Advances in Database Theory*, Vol. 2, H. Gallaire, J. Minker, and J-M. Nicolas ed., Plenum Press, 1984, 145 - 170.
- [ISO, 1992] ISO TC184/SC4, *STEP Part 1: Overview and fundamental principles*, September 15, 1992.
- [ISO, 1993] ISO TC184/SC4/WG7, *STEP Part 22: Standard Data Access Interface*, 1993.
- [ISO, 1994] ISO TC184/SC4, *STEP Part 11: EXPRESS Language Reference Manual*, July 1, 1994.
- [Jagadish, Qian, 1992] H. Jagadish and X. Qian, "Integrity Maintenance in an Object-Oriented Database," *Proc. ACM-SIGMOD 1992 Int'l Conf. on Management of Data*, 1992, 469 - 480.
- [Mueller 외 2, 1993] W. Mueller, G. Lehrenfeld, and N. Wiechers, "Parallel Validation of STEP Files," *Proc. Third International EXPRESS User Group Conference*, Meeting Associates, Clifton Park, N.Y., 1993.
- [OMG, 1996] OMG, *CORBA 2.0 Specification*, March 1996.
- [Owen, 1993] Jon Owen, *STEP An Introduction*, Information Geometers Ltd, Winchester, UK, 1993.
- [Schoenefeld, Bohm, 1993] F. Schoenefeld and C. Bohm, "Using EXPRESS Database Technology for Accessing NCBI Genomic Data," in P. Wilson, editor, *Proc. Third*

- International EXPRESS User Group Conference*, Meeting Associates, Clifton Park, N.Y., 1993.
- [Sheth, Larson, 1990] A. Sheth and J. Larson, "Federated database systems for managing distributed, heterogeneous, and autonomous databases," *ACM Computing Surveys*, Vol. 22, No. 3, 1990, 183 - 236.
- [Trapp, 1991] G. Trapp, "Advancing Concurrent Engineering Using STEP," *Proc. CALS Expo*, J&J Assoc., Phoenix, Arizona, 1991.
- [Urban 외 2, 1992] S. Urban, A. Karadimce, and R. Nannapaneni, "The Implementation and Evaluation of Integrity Maintenance Rules in an Object-Oriented Database," *Proc. Eight Int'l Conf. on Data Engineering*, 1992, 565 - 572.
- [Vuoskoski, 1994] J. Vuoskoski, et. al., "Using EXPRESS in High Energy Physics Research Environment," in P. Wilson, editor, *Proc. Fourth International EXPRESS User Group Conference*, Meeting Associates, Clifton Park, N.Y., 1994.
- [Wu 외 2, 1992] J. Wu, T. Liu, and G. Fisher, *An Integration of CAE and CAM Applications Using PDES/STEP Information Model*, Technical Report, Center of Computer-Aided Design, The University of Iowa, Iowa City, IA, 1992.
- [Yoo, Cha, 1993] S. B. Yoo and S. K. Cha, "Checking EXPRESS Constraints in A Heterogeneous Database Environment," *Proc. Third International EXPRESS User Group Conference*, Meeting Associates, Clifton Park, N.Y., 1993.

저자소개

- 유상봉**: 1982년 서울대학교 제어계측공학과 학사, 1986년 Arizona 주립대학교 전기및컴퓨터 공학과 석사, 1990년 Purdue 대학교 전기및컴퓨터공학과 박사, 1989년 AT&T Bell 연구소 연구원, 1990년 삼성전자 컴퓨터부문 선임연구원, 1992년 ~ 현재 인하대학교 자동화공학과 부교수, 관심분야는 공학 데이터베이스, 시스템 통합, STEP, CALS, PDM, 동시공학 등임.
- 서효원**: 1991년 미국 웨스트버지니아대학교 산업공학 박사, 1983 ~ 1987년 대우중공업 중앙연구소 주임 연구원, 1992 ~ 1995년 생산기술연구원 수석연구원, 1996년 ~ 현재 한국과학기술원 산업공학과 조교수, 관심분야는 CE, CIM 등임.
- 고평욱**: 1993년 인하대학교 자동화공학과 학사, 1996년 인하대학교 자동화공학과 석사, 1996 ~ 현재 인하대학교 자동화공학과 박사과정, 관심분야는 공학 데이터베이스, STEP, EDM/PDM, 분산 시스템 등임.