

설계이력 정보를 이용한 CAD 모델의 오류 수정

양정삼*, 한순홍**

Healing of CAD Model Errors Using Design History

Yang, J. S.* and Han, S. H.**

ABSTRACT

For CAD data users, few things are as frustrating as receiving CAD data that is unusable due to poor data quality. Users waste time trying to get better data, fixing the data, or even rebuilding the data from scratch from paper drawings or other sources. Most related works and commercial tools handle the boundary representation (B-Rep) shape of CAD models. However, we propose a design history-based approach for healing CAD model errors. Because the design history, which covers the features, the history tree, the parameterization data and constraints, reflects the design intent, CAD model errors can be healed by an interdependency analysis of the feature commands or of the parametric data of each feature command, and by the reconstruction of these feature commands through the rule-based reasoning of an expert system. Unlike other B Rep correction methods, our method automatically heals parametric feature models without translating them to a B-Rep shape, and it also preserves engineering information.

Key words : Boundary representation, CAD model, Design intent, Feature, Healing, Interoperability

1. 서 론

최근 시장에서 업체들간의 지속적인 경쟁으로 인해 제품개발 단계에서 리드타임(Lead-time)이 줄어들고 제품의 복잡성이 증가되고 있다. 또한 기업의 세계화 추세에 따라 하나의 제품을 개발하는데 있어서 특정 업체의 경계를 넘어 분산환경에서 공동설계가 이루어지고, 공통의 부품(Part) 또는 컴포넌트(Component)에 대한 CAD 데이터가 빈번하게 교환이 이루어지고 있다. 그러나 설계자들을 끊임 없이 괴롭히고 있는 것은 저급한 품질로 인해 사용할 수 없는 CAD 모델을 다루는 일이다. 설계자는 전달받은 CAD 데이터를 활용하기 위해 품질 문제가 발생한 형상의 위치를 찾고 수정하기 위해 많은 시간을 투입하고 있다. 미국자동차산업의 공급망(Supply chain)에서 CAD 모델을 교환하는 과정에서 발생한 CAD 모델의 저급한 품질로 인

해 연간 약 10.5억 달러의 비용이 소요되고 있다^[1]. 미국의 3대 완성차 조립업체에서 연간 450,000건의 PDF(Product Data Exchange)가 일어나고 있으며 CAD 모델의 품질을 향상하기 위해 총 개발 기간의 50%를 투입하고 있고 한 건의 모델당 4.9시간이 소요되고 있다. 일본자동차산업에서는 250,000건의 PDF가 발생하고 연간 71억 엔의 비용과, 한 개의 모델당 1.5시간의 리드타임이 발생하고 있다^[2].

CAD 모델에서 오류를 파악하거나 수정하는 기존의 연구는 B-Rep(Boundary Representation) 형상을 대상으로 진행하였다^[3-5]. 즉, B-Rep 형상에서 위상요소와 기하요소에 대한 수학적 연산 과정을 통해 오류를 찾고 위상요소와 기하요소에 대한 수정을 통해 오류를 수정하였다. 상업용 S/W들도 B-Rep 정보를 통해 CAD 모델의 오류를 파악하고 수정하는 방법을 사용하고 있다. B-Rep 정보를 이용하는 방법은 정량적인 입력 값을 통해 오류의 정확한 위치를 파악하는데 효과적이지만, B-Rep 정보에 대한 수정은 특정 형상의 왜곡을 발생시킬 수 있고, 상황에 따라서는 형상의 붕괴(Core-dump)를 초래한다. 이로 인해 대부분의 상

*학생회원, (주)부품다비 기술연구소

**중신회원, KAIST 기계공학과

- 논문투고일: 2004. 12. 06

- 심사완료일: 2005. 04. 25

업용 S/W는 제한된 오류 항목들에 대해서만 수정 기능을 제공하고 있고, 이들 수정 기능들에 대한 자동화 수준이 낮다. 한편 실제 현업에서는 상업용 S/W의 수정기능에 대한 활용에 대해 보수적이다.

본 논문에서는 설계 이력을 기반으로 CAD 모델의 형상을 수정하는 방법을 제안한다. 설계이력은 특징 형상(Feature) 정보, 작업 이력 트리(Working history tree), 파라메트릭 데이터(Parameterization data) 그리고 구속조건(Constraint)를 포함한다. 설계이력은 설계자가 형상을 생성하는 과정에서 사용한 요소들로서 설계자의 작업 의도가 기록되어 있다. 본 논문에서는 CAD 모델의 오류 사례들에 대한 분석을 통해서, 모델링 명령어들 사이의 패턴 관계와, 개별 파라메트릭 정보들 사이의 상호 의존 관계를 정의하였다. 또한 상용 CAD 시스템에서 생성된 모델로부터 추출된 설계 이력을 구조화 하기 위해서 표준화된 스키마를 정의하였고, 이를 전문가시스템의 오류 추론과정과 설계 이력의 변경에 적용하였다. 제안된 방법을 검증하기 위하여 상용 CAD 시스템인 CATIA V5 환경에서 "Tiny face", "Narrow region", "Non-tangent face", "Narrow step", "Sharp face angle", "Narrow space"의 6개 항목에 대한 오류를 수정하는 응용프로그램을 구현하여 현업에서 만들어진 동력전달장치(Power train) 모델을 대상으로 적용 실험을 하였다.

2. 관련 연구

2.1 B-Rep 기반의 오류 수정

CAD 모델의 오류 수정에 대한 연구는 Exact B-Rep 형상을 이용하는 방법, Faceted B-Rep 형상을 이용하는 방법 그리고 경계곡선(Boundary curve)을 이용하는 방법으로 이루어진다. Exact B-Rep 형상을 이용하는 방법은 형상을 구성하는 위상요소와 기하요소를 표현하는 자료구조에 대한 수학적 연산 과정을 통해 오류를 확인하고 이를 수정하였다. Hoffman^[3]은 기하 공차에 의해 발생된 형상의 오류를 사용자에 의해 제어되는 볼록 매커니즘을 이용해 수정하였다. Gul^[4]은 Complementary model object tree를 이용하여 위상엔티티를 순차적으로 분류하고, 각각의 엔티티에 발생하는 오류를 수정하였다.

Exact B-Rep을 Faceted B-Rep의 다면체로 근사화시켜 오류를 진단하고 수정하는 방법은, 연산 속도가 빠른 장점을 가지고 있다. Barequet^[6]은 복잡한 형상의 트림 곡면을 정렬되지 않은 다면체 형태의 작은 조각(Unordered lists of polygons or surfaces)으로

나눈 후, 순서에 따라 재 구성하는 해쉬 알고리즘(Hashing algorithm)을 제안하였다. 이를 바탕으로 다면체 사이에 발생하는 간격(Gap)을 인접 다면체에 결합(Matching) 하는 방법으로 오류를 수정하였다. 이 방법은 다면체의 면과 면 사이에 발생한 간격에 대해 수정(Stitching)이 가능하다.

경계곡선을 이용하는 방법은 곡면으로만 이루어진 형상에 대해서만 제한적으로 적용할 수 있다. Steinbrenner^[7]는 차수가 다양한 곡면으로 이루어진 모델에 대해 인접 곡면 사이의 간격과 중첩을 진단하고 수정하는 연구를 하였다. 이 방법은 경계곡선을 에지(Edge) 형태의 작은 곡선으로 나눈 뒤, 인접한 에지들 간의 관계를 비교하여 에지를 절단하고 합치는 과정(Edge splitting and merging)을 반복 수행하여 간격과 중첩 같은 오류를 수정하였다. Volpin^[8]은 G1 연속성을 갖는 곡면으로 수정하기 위해, 곡면의 면적과 곡률의 변화에 따라 곡면의 경계곡선을 구성하고 있는 곡선을 나누고, 이를 바탕으로 곡면을 재 구성하는 방법을 사용하였다.

대부분의 연구가 B-Rep 모델을 대상으로 수행되었고 오류 수정에 있어서 형상의 간격과 중첩과 같은 제한적인 항목에 대해서만 수정이 가능하다. 또한 형상을 단순화 하는 과정에서 데이터 손실을 감수해야만 한다.

2.2 상용 프로그램 분석

Table 1은 7개의 상업용 응용프로그램에 대한 비교를 보여준다. CATIA V4.x의 내부 자료 구조를 공유하는 Q-Checker를 제외하고 나머지 프로그램은 별도의 커널을 기반으로 만들어 졌다. 따라서 CAD 모델에서 프로그램 고유의 자료구조 형태로 변환하는 과정에서 데이터 손실을 초래할 수 있다. 또한, 이들 상업용 프로그램들은 B-Rep 모델을 기반으로 오류 진단과 수정이 이루어진다. 비록 특징형상 기반의 CAD 시스템에서 만들어진 설계이력을 포함하고 있는 모델이라도 B-Rep 데이터 만을 프로그램 내부의 자료구조로 변환해서 오류의 위치를 파악하고 위상구조와 기하구조에 대한 변경을 통해 수정이 이루어진다. 위상구조 변경을 통한 오류 수정은 비교적 단순한 형상에 제한적으로 적용할 수 있다.

Fig. 1은 위상구조의 일부를 변경해서 Tiny face를 제거하는 예를 보여 준다. Tiny face가 발생한 Face 4를 인접 페이스인 Face 1과 병합하기 위해서, 두 개의 면 사이에 있는 경계에지(Common edge)를 제거하고, 이 경계에지에 포함된 End point(vertex)를 근기

Table 1. Comparison of seven commercial applications

Application	Vender	Kernel	# of error types	Feature
CADIQ v3.4	TranscenData (US)	Proprietary kernel, Parasolid	19	-Error checking tool -Applicable to surface and solid modes.
CADFix v6.0	TranscenData (US)	Proprietary kernel, Parasolid	99	-Error repairing tool for the downstream application. -A user can interactively repair and manipulate CAD model using various options.
GeometryQA v4.1	Prescient (US)	ACIS	46	-Error checking tool. -Provide the capabilities to automatically check, and the faster geometry computation process.
CADHealer	Theorem (UK)	Proprietary kernel	12	-Error repairing tool. -Applicable to surface, solid, and wireframe models
Q-Checker v1.9	TransCAT (Germany)	CATIA kernel	19	-Plug-in for CATIA V4 system and use its internal data structure.
CADDoctor	Elysium (Japan)	Parasolid	18	-Error checking tool. -Have the advantage of repairing a surface model
spGate	Armonicos (Japan)	Proprietary kernel	46	-Error repairing tool. -Using InterOp translators of Dassault Systemes. -Support the requirements of a quality standard such as SASIG JAMA, and VDA.

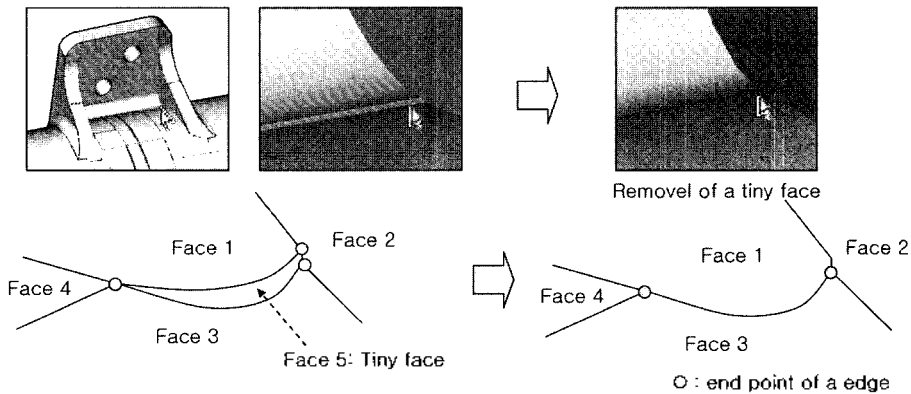


Fig. 1. Repair of a tiny face by adjusting topological elements.

리에 있는 End point(vertex)와 동일한 위치에 놓이도록 한다. 상용 프로그램의 커널 내부로 읽어 들이는 과정에서 데이터 손실로 인해 복잡한 형상의 CAD 모델이 불안정한 위상구조로 유지되는 경우에는, 오류 수정 후 설계자가 원하지 않는 뒤틀림 형상으로 발견하거나, 형상 자체가 붕괴되는 경우가 발생한다. 따라서 오류 수정이 가능한 항목은 Small edge, Tiny face, 곡면 법선벡터의 방향 조정 등과 같은 일부 항목들에 제한된다. 또한 사용자가 오류 수정 과정에 직접적으로 개입하기 때문에 자동화 수준이 낮다. 이로 인해, 현업에서는 오류수정 프로그램에서 제공하는 수정 기능을 활용하지 않고, 개발업체에서도 오류의 수정보다는 오류 진단에 초점을 맞춰 개발하고 있다.

이들 상용 프로그램에서 수정된 결과는 설계 정보가 배제된 B-Rep 모델을 대상으로 하기 때문에, 수정된 결과가 설계 정보를 포함하고 있는 수정 전의 CAD 모델이 반영되기는 어렵다. 즉 상위 설계부서에서 생성된 CAD 모델을, 형상의 꺾이기인 B-Rep 모델만을 수정해서, 다시 상위 설계부서로 전달하는 것은 무의미하기 때문이다.

3. 설계이력 기반의 오류 수정

3.1 연구 범위

CAD 모델 오류의 종류는; (1) 형상의 기하요소와 위상요소에 대한 수치적인 부정확성을 의미하는 오

류, (2) CAD 모델을 전달받는 하위 CAD 시스템의 사용 목적에 맞지 않는 오류, 그리고 (3) 설계자가 모델링 하는 과정에서 경험 부족으로 인해 발생하는 설계 방법론적인 오류로 나눈다.

(1) 수치적으로 부정확한 형상의 오류는, 위상요소와 기하요소에 대한 수학적 해석을 통해 C0/G1/G2 연속성, Small/Void face, Self-intersection, 등과 같이 정량적인 입력 값을 통해 오류를 찾고, 작업하는 CAD 시스템의 자료구조에서 적합하도록 위상요소와 기하요소를 변경해서 수정한다. 오류의 발생 원인은 CAD 데이터를 교환하는 두 개의 CAD 시스템의 정밀도 또는 CAD 시스템의 엔진에 해당하는 형상 커널의 차이에 따라, 데이터 변환과정에서 데이터 손실에 의해 오류가 발생한다. 또한 형상을 생성하는 CAD 시스템 자체에 문제가 있는 경우에도 발생한다.

(2) 하위 CAD 시스템의 사용 목적에 맞지 않는 오류는, 상위 CAD 시스템에서 생성된 형상이 하위 CAD 시스템에서 활용할 수 없는 경우이다. 상위 CAD 시스템에서 미려한 형상을 만들기 위해 7차 이상의 곡률을 가진 곡면을 사용했지만, 하위 CAD 시스템에서는 5차 이상의 곡면을 표현할 수 없어 곡면이 소실될 수 있다. 이러한 오류를 회피하기 위해 상위 CAD 시스템에서 형상을 생성할 경우에 설계자는 해석 또는 가공 조건을 고려한 설계 방법을 적용한다 (DFEM, Design for Engineering & Manufacturing).

(3) 설계방법론적인 오류는, 설계자가 모델을 생성하는 과정에서 경험부족으로 인해 잘못된 설계 프로세스를 수행하여 발생한다. 최근 연구결과에서는 CAD 모델의 품질이 저하되는 주요한 원인으로 설계자의 경험부족을 얘기한다^[1,12].

본 논문에서 정의하는 CAD 모델의 오류는, 정량적인 입력 값에 따라 위상요소와 기하요소에 대한 수학적 해석을 통해, 국제표준 또는 회사 내 표준으로 설정된 허용 범위를 벗어난 것을 의미한다. 즉, 곡선의 C0 연속성에서 사내 표준에서 규정된 최소 허용 값이 0.01 인 경우, 두 개의 연속된 곡선 사이의 간격이 0.01보다 큰 경우는 오류가 된다. 그러나 곡면 형상에서 설계자에 의해 의도된 작은 곡률 반경을 가지고 있더라도, 규정된 곡률 반경 보다 작은 경우에는 오류로 검증 될 수 있다. 본 논문에서는 오류의 위치를 파악하기 위해서 B-Rep 정보를 이용하였고, 오류 수정은 설계이력 정보와 전문가시스템의 추론 방식을 이용하였다.

3.2 설계이력 정보의 추출

제품개발에 사용하고 있는 대부분의 CAD 시스템

은, B-Rep 정보뿐만 아니라 설계이력을 포함하는 자료구조를 가지고 있고, 이를 특징형상 기반의 파라메트릭 CAD 시스템이라 부른다.

Fig. 2는 특징형상 기반의 파라메트릭 CAD 시스템에서 Pad 명령어와 Fillet 명령어를 이용해서 형상을 생성한 후에 CAD 시스템 내부의 자료구조상에 저장된 설계이력 정보(History information)와 B-Rep 형상 정보(B-Rep result shape)를 보여준다. Pad 특징형상 명령어는 높이(H)와 Pad의 단면 엔터티(Sketch)에 대한 파라메트릭 데이터를 설계자로부터 입력 받아서 B-Rep 형상(B-Rep for Pad)을 만든다. Fillet 특징형상 명령어는 이전에 만들어진 Pad의 결과 형상으로부터 선택된 예지와 반경(radius of Fillet)을 입력 받아서 B-Rep 형상(2D geometry for Fillet)를 만든다.

본 논문에서는 B-Rep 형상 정보를 통해 CAD 모델의 오류를 찾고, 설계이력 정보의 제 구성을 통해 오류를 수정하였다.

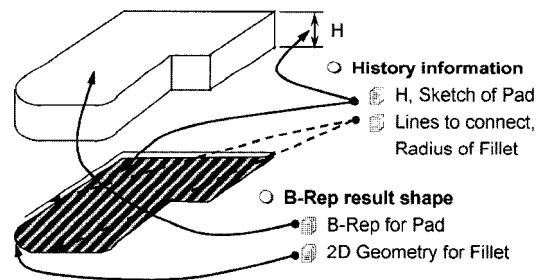


Fig. 2. Representation of the B-Rep data and the design history information.

설계이력 정보를 이용해서 CAD 모델의 오류를 수정하기 위해서는, 설계자가 CAD 시스템에서 형상을 생성하기 위해 사용한 특징형상 명령어 이력(Feature commands history), 특징형상을 생성하기 위해 사용한 파라메트릭 데이터(Parameterization data) 그리고 구속조건을 추출하는 것이 필요하다. 또한 추출된 설계이력 정보를 다수의 CAD 시스템에서 공유하기 위해서 표준화된 구조로 표현하는 것이 필요하다. 본 논문에서는 XSD(XML Schema Definition)를 이용해서 표준화된 설계이력을 정의하였다.

설계이력 스키마는 Fig. 3에서 보는 바와 같이 형상을 생성하는 특징형상 명령어 그룹(FeatureCmds)과, 구속조건을 포함해서 참조 데이터를 생성하는 그룹(ReferenceItems)로 구성된다. 솔리드 형상을 생성하는 21개의 특징형상 명령어들은 매크로파라메트릭 방법론^[13-15]에서 제안된 Working draft V2.0의 핵심 명

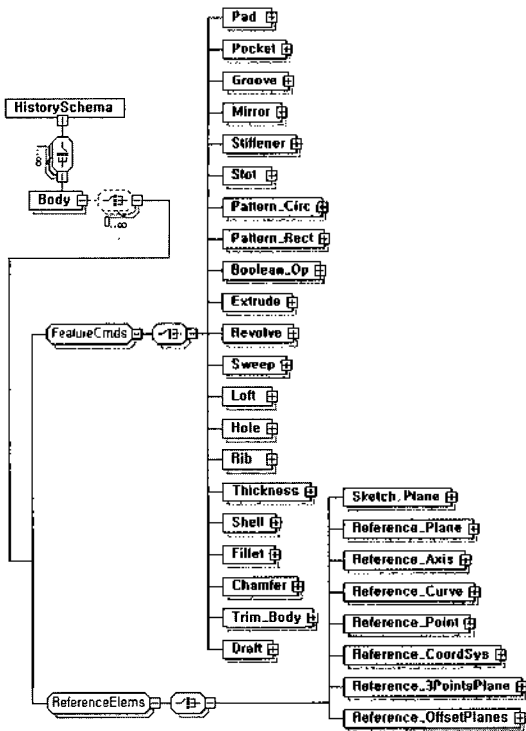


Fig. 3. Design history schema representation with the feature and reference command groups.

명어 그룹(Core commands)를 바탕으로 정의되었다. 각각의 명령어는 특징형상을 인스턴스 시키기 위한 파라메트릭 데이터를 포함한다. 기준평면(Plane), 기준축(Axis), 좌표시스템 등과 같이 8개의 참조 엘리먼트를 생성하는 명령어들로 이루어진 ReferenceElems 그룹은, 솔리드 형상을 만들기 위해 보조적으로 필요한 엘리먼트를 정의할 수 있다. CAD 모델로부터 추출된 설계이력은 하나 이상의 Body 노드들로 구성되고, 각각의 Body는 다수의 특징형상 명령어 또는 참조 엘리먼트를 포함한다.

Fig. 4는 7개의 파라미터로 구성된 Pad 명령어를 정의한 XML 스키마를 보여준다. 각각의 엘리먼트 이름이 정의하는 내용은 다음과 같다.

- profile_curves: Pad의 윤곽 곡선. 윤곽 곡선이 둘 이상일 경우, 첫 번째가 외곽 경계곡선이고, 두 번째부터는 내부 경계곡선이다.
- first_limit_type: profile_curves를 own_direction 방향으로 돌출(protrusion)할 때의 멈춤 조건.
- second_limit_type: profile_curves를 own_direction의 반대 방향으로 돌출할 때의 멈춤 조건.
- first_length, second length: first와 second의 길이.

```

- <xs:element name="Pad">
- <xs:complexType>
- <xs:sequence>
~ <xs:element name="profile_curves"
maxOccurs="unbounded">
- <xs:complexType>
<xs:attribute name="profile_name" type="xs:string"
use="required" />
<xs:attribute name="parent_name" type="xs:string"
use="required" />
</xs:complexType>
</xs:element>
<xs:element name="first_limit_type" type="end_type" />
<xs:element name="first_length" type="xs:double" />
<xs:element name="second_limit_type" type="end_type" />
<xs:element name="second_length" type="xs:double" />
<xs:element name="reference_plane" type="xs:string" />
<xs:element name="own_direction" type="direction"
minOccurs="0" />
</xs:sequence>
<xs:attribute name="result_solid" type="xs:string"
use="required" />
</xs:complexType>
</xs:element>
    
```

Fig. 4. XML schema definition of the Pad feature command.

- reference_plane: Pad가 생성되는 기준면.
- own_direction: first_limit의 진행 방향.

Profile_curve와 같이 특징형상 명령어의 파라미터가 외부 엘리먼트를 참조하는 경우에는 문자열(string) 형태의 엘리먼트 이름 (Identification name)을 이용하게 된다. 이때 중복된 문자열로 인해 모호성을 피하기 위해서, Fig. 4에서 화살표로 표시된 부분과 같이 참조되는 엘리먼트에 대한 부모 엘리먼트의 이름을 함께 정의하였다. 문자열에 의한 외부 엘리먼트의 참조는 설계이력 정보간에 상호 의존 관계에 대한 해석과 수정된 설계이력을 생성하는 과정에서 해당 엘리먼트에 접근하기 위해 사용된다.

Fig. 5는 Fig. 2의 형상을 CATIA V5로 생성한 모델로부터 설계이력 스키마에 따라 XML 포맷으로 설계이력을 추출한 결과를 보여 준다. 결과 형상의 이름이 Pad.1인 Pad 특징형상 명령어는 Dimension 타입 (Pad의 Length를 정의해서 형상을 생성)으로서 첫 번째 Length는 30 mm 그리고 두 번째 Length는 5 mm로 정의되었다. 2D Profile은 PartBody 아래에 위치한 Sketch.1이 참조되었다. 기준면이 되는 Reference plane은 전역좌표계의 XY plane으로 선택되었다. Fillet.1은 이전 명령어로 수행된 결과 솔리드 형상 (Pad.1)에서 선택된 1개의 Edge(Edge.5)에 대해 2 mm의 Fillet 특징형상 명령어를 접선(Tangency) 조건과 함께 수행된 결과 형상이다.

이와 같이 XML 형태로 추출된 설계이력 정보는 DOM(Document Object Model) Interface를 이용해

```

<?xml version="1.0" encoding="UTF-8" ?>
<HistorySchema
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-
  instance" xsi:schemaLocation="D:\HistorySchema.xsd">
  <Body body_name="PartBody">
  - <Pad result_solid="Pad.1">
    <profile_curves profile_name="Sketch.1"
      parent_name="PartBody" />
    <first_limit_type>Dimension</first_limit_type>
    <first_length>30.000</first_length>
    <second_limit_type>Dimension</second_limit_type>
    <second_length>5.000</second_length>
    <reference_plane>xy plane</reference_plane>
  </Pad>
  - <Fillet result_solid="Fillet.1">
    <targets>Edges</targets>
    <fillet_type>Constant</fillet_type>
    <radius>2.000</radius>
    <propagation_type>Tangency</propagation_type>
    <target_edges target_name="Edge.5"
      parent_name="Pad.1" />
  </Fillet>
  </Body>
</HistorySchema>
  
```

Fig. 5. Example of the design history file extracted from a CAD model.

서 전문가시스템의 추론 과정과 오류 수정을 위한 설계이력 정보의 변경에 사용된다.

3.3 설계이력 정보의 해석

CAD 모델로부터 추출된 설계이력 정보를 통해 설계이력들간에 상호 의존관계(Inter-dependency)를 해석하고, 설계이력을 변경해서 오류를 수정할 수 있는

명령어 패턴관계를 정의해야 한다. 의존관계를 해석하는 부분은 추출된 설계이력 리스트로부터 상위 명령어와 하위 명령어간의 종속관계에 대한 패턴을 해석하는 부분(Commands pattern analysis)과 각각의 명령어에 포함된 파라미터 또는 구속조건 등이 다른 명령어들과의 의존관계를 해석하는 부분(Parametric data analysis)으로 구분된다.

Commands pattern analysis는 Fig. 6에서 보는 바와 같이 6개의 오류 항목에 대해 오류발생에 직접적으로 영향을 주는 13개의 명령어와, 이들 13개의 명령어에 종속되어 오류 발생에 영향을 주는 21개의 명령어들의 관계에 대한 해석을 한다. 즉, CAD 모델로부터 추출된 설계이력 정보에서 특정 명령어를 수행한 이후의 결과 형상에 대해 위상요소와 기하요소들에 대한 수학적 해석을 통해 오류의 위치를 확인하고, 이 명령어가 수행되기 위해 종속적으로 연결되는 21개의 명령어 집합과의 패턴 관계를 정의 하였다.

한 예로서 Fig. 7에서 보여주는 Narrow space 오류는, Fig. 6에서 굵은 직선으로 표시된 Narrow Space: Pad(Pad.2)→Draft(Draft.2) 형태의 명령어 패턴 형식으로 표현 되었다. XML로 표현된 설계이력에서 보는 바와 같이 오류가 발견된 위치는 Pad 명령어를 수행한 결과 형상에서 발견되었고, 이 Pad 명령어는 이전 명령어인 Draft 명령어의 결과 형상을 입력 받았다.

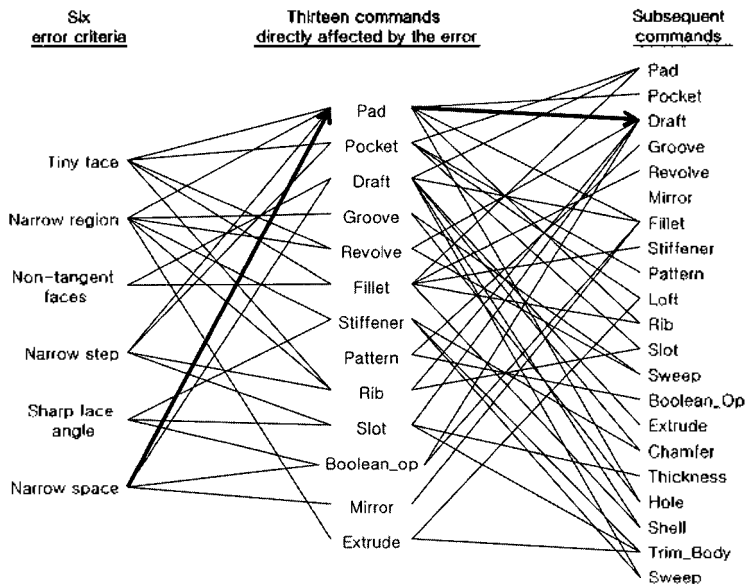


Fig. 6. Commands pattern relationship with the types of CAD model errors.

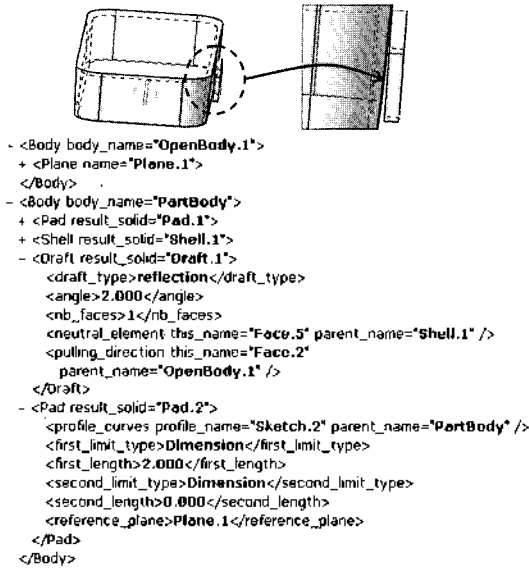


Fig. 7. Example of the narrow space in the design history.

오류 항목과 명령들간의 패턴 관계는 현대기아차 동차, 다임러크라이슬러, 일본자동차공업협회(JAMA), ITI Solutions에서 제공 받은 43개의 오류 사례에 대한 연구를 통해 구성되어 지식베이스로 구축하였다. 본 논문에서는 NASA Johnson Space Center에서 개발된 전문가시스템인 CLIPS 6.0¹⁶⁾을 기반으로 48개의 명령어 패턴 룰(rule)과 39개의 설계이력 변경들을 구축하였다. 구축된 지식베이스를 기반으로 CAD 모델로부터 설계이력 정보가 전문가시스템으로 입력되면 지식베이스의 추론 과정을 통해 오류를 회피할 수 있는 새로운 설계이력을 생성할 수 있다.

특정형상 명령어 내부에 포함되어 있는 파라미터 또는 구속조건을 무시하고 단순히 Commands pattern analysis를 통해 오류가 있는 명령어들의 순서만을 변경해서 모델을 재 구성할 경우 전혀 다른 형상이 만들어지거나 붕괴되는 경우가 많다. Parametric data analysis는 각각의 명령어에 포함된 파라미터 또는 구속조건 등이 다른 명령어들과의 의존관계를 이루고 있는지 여부를 해석한다. Fig. 8은 이러한 파라미터 상호간의 의존관계를 해석하는 프로세스를 보여 준다. Feature_A의 Attribute data와 Feature_B의 Attribute data를 비교해서 서로 의존관계를 유지하고 있는 Attribute data를 해석한다. 이 해석 결과는 CAD 모델로부터 추출된 XML 형식의 이력 파일 내에 저장된다.

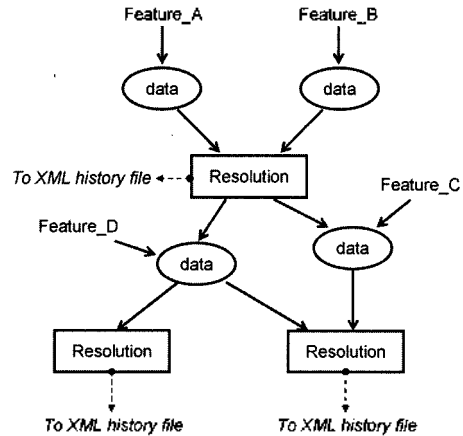
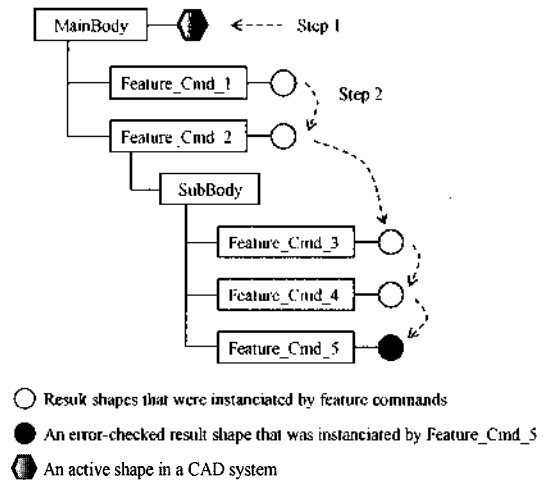


Fig. 8. Parametric data analysis for maintaining the coincidence among the parameters or constraints.

3.4 오류 수정

본 논문에서는 오류를 수정하기 위해 설계이력 정보를 이용하고, CAD 모델에 있는 오류의 위치를 파악하기 위해 기존에 연구된 방법과 마찬가지로 B-Rep 정보에 대한 위상 구조와 기하구조의 수학적 연산 방법을 이용하였다. Rogniac Complex Cell¹⁷⁾ 자료 구조를 가지고 있는 CATIA V5 모델을 대상으로 6개의 오류항목(Tiny face, Narrow region, Non-tangent face, Narrow step, Sharp face angle, Narrow space)에 대한 검증을 한다.

이들 6개의 오류 항목은 Table 1에서 소개한 상용 프로그램에서 수정 기능이 취약한 항목을 선택 하였



- Result shapes that were instantiated by feature commands
- An error-checked result shape that was instantiated by Feature_Cmd_5
- ◐ An active shape in a CAD system

Fig. 9. Error checking process looking for the location of errors among the feature commands.

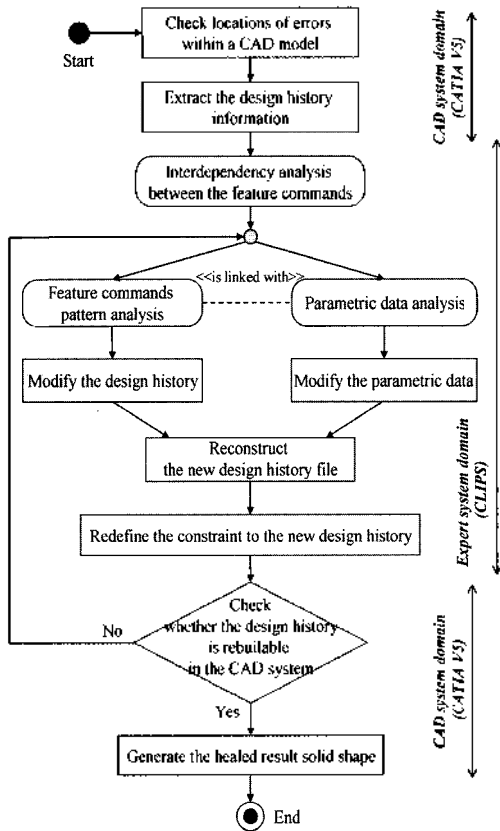


Fig. 10. Healing process.

고, 이들 항목에 대한 용어와 개념 정의는 SASIG Product Data Quality Documentation V1.0을 참고하였다^[18].

오류 수정을 위해 가장 먼저 해야 하는 일은 CAD 모델로부터 오류가 발생한 위치를 찾는 일이다. Fig. 9에서 보는 바와 같이 우선 전체 형상(MainBody)로부터 6개의 오류 항목에 대한 검증 작업을 진행한다. 만약 MainBody로부터 특정 오류가 발견되면, 해당 오류가 설계이력의 어느 위치에서 발생되었는지 순차적으로 각각의 특징형상 명령어에 대한 결과 형상의 오류를 진단한다.

CAD 모델로부터 오류가 확인된 결과는 Fig. 10에서 보는 바와 같이 추출된 설계이력 정보와 함께 전문가시스템으로 입력된다. 전문가시스템에서는 이미 구축된 지식베이스의 룰에 따라 설계이력 정보 사이에 발생된 의존관계에 대한 해석과 각각의 명령어 패턴 분석, 그리고 파라메트릭 데이터 분석을 위한 해석 프로세스가 진행된다. 이를 바탕으로 지식베이스의 수정 룰에 따라 설계이력 파일이 재 생성되고, 변경된

설계이력에 대한 새로운 구축조건이 정의된다. 수정된 설계이력에 대해 CAD 시스템이 내부적으로 재구성(Re-building)이 가능한지 확인한 후 형상을 생성한다.

3.5 구현 및 실험

본 논문에서 제안된 설계이력 기반의 오류 수정 방법을 검증하기 위해, CATIA V5R11 기반의 프로그램을 개발하였다. 전문가시스템은 NASA에서 개발된 CLIPS 6.0의 소스코드를 수정해서 CATIA 모듈 내부에 포함시켰다.

개발된 시스템의 구조는 Fig. 11에서 보는 바와 같이 4개의 모듈로 구성된다. 설계이력을 추출하는 모듈(Design history extraction module)은 CATIA V5 인터페이스 API인 CAA를 이용해서 CAD 모델로부터 설계이력 정보를 추출하여 3.2절에서 소개된 설계이력 스키마에 따라 XML 구조의 파일을 생성한다.

오류의 위치를 찾는 모듈(Geometry checking module)은 CATIA 모델의 B-Rep 정보를 CAA를 이용해서 읽어와서 6개 오류 항목에 대한 사용자의 기준 값을 바탕으로 수학적 연산을 통해 설계이력 정보에서 오류가 발생한 위치를 찾는다.

CLIPS 6.0의 소스코드를 일련의 리스트(list) 형태의 설계이력 정보를 Rule로 정의하고, 추론(Backward reasoning) 기능을 수행할 수 있도록 수정하여 CATIA 내부 모듈로 포함하였다. 구축된 CLIPS의 지식베이스는 48개의 명령어 패턴 매칭 Rule, 39개의 수정 Rule, 그리고 명령어들간에 의존관계를 해석하는 37개의 Rule로 구성되었다.

수정 물의 결과에 따라 설계이력 정보를 재 구성하는 모듈 (History reconstruction module)은 변경된 설계이력이 CATIA 환경에서 재구성이 가능한지 여부

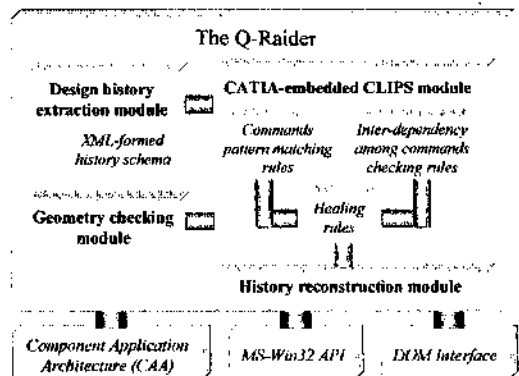


Fig. 11. The structure of the CAD model healing system.

를 확인하고, 변경된 설계이력 에 대한 구속조건을 재정의한다.

오류 수정 시스템의 외부 인터페이스 모듈은 닷소 시스템의 CAA, 사용자 인터페이스를 위한 MS-Win32 API, 그리고 XML 스키마와 설계이력 파일에 접근을 위해 DOM 인터페이스를 이용하였다.

Fig. 12는 개발된 시스템 Q-Raider의 초기화면과 오류 수정을 위한 컨트롤 패널을 보여준다. 수정에 필요한 기능은 컨트롤 패널을 통해 이루어진다. 사용자는 컨트롤 패널을 이용해서 오류 항목별 기준 값 설정, 적용 룰의 설정, 오류 엘리먼트에 대한 그래픽 설정, 그리고 6개 오류 검증 항목에 대한 허용치를 지정할 수 있다.

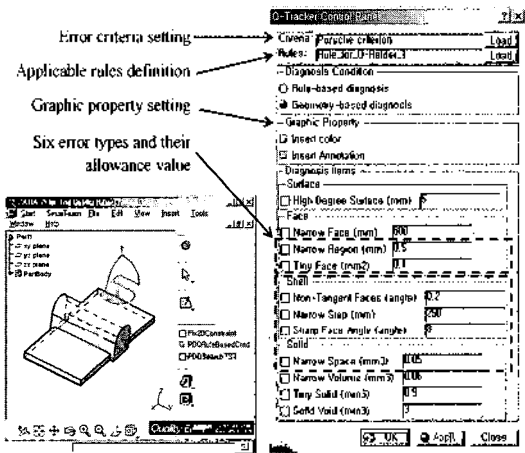


Fig. 12. The main window and control panel of Q-Raider.

Fig. 13은 Crankshaft에 대한 모델로서 면적인 0.094 mm^2 인 "Tiny face" (왼쪽 화살표로 표시된 그림)와 폭이 0.09 mm 이고 연속된 에지 길이의 비가 484:1(장축:단축)인 "Narrow step"이 발견되었다. CAD 모델로부터 추출된 설계이력 정보는 MainBody인 Crankshaft와 5개의 SubBody, 그리고 Reference element로 구성되어 있다.

설계자는 Crank 부분을 모델링 하면서 화살표로 표시된 Profile을 Shaft의 축을 중심으로 360° 회전하면서 Remove 시켰다(설계이력에서 B). Crank 원판의 오른쪽은 정상적인 형상이 만들어 졌지만, 왼쪽은 Profile 경로가 지나가는 부분에서 Tiny face가 만들어 졌다. 이것은 Crank 원판의 오른쪽과 왼쪽의 회전축이 일치하지 않기 때문이다. Tiny face를 제거하기 위해 설계이력에서 Tiny face가 발생한 명령어를 찾고(Body.4의 Shaft.3), Crank 원판의 왼쪽 부분에 대

한 파라메트릭 수정을 한 뒤 (ReferenceElems에 Curve.12를 생성), Shaft(Shaft.12)를 하였다.

Narrow step이 발생한 부분은 위쪽 원판(C에서 Pad.1)과 아래쪽 원판(C에서 Pad.2)에 의해 솔리드를 생성한 후, Circular Pattern의 실린더(A)를 솔리드 형상에 대해 Remove(C에서 Remove.3) 하였다. 이때, Remove 명령어가 적용된 아래쪽 원판에서 Narrow Step이 발생하였다. 잘못된 모델링 순서에 의해 발생한 오류로서 위쪽 원판을 만든 후 아래쪽 원판을 만들기 전에, 위쪽 원판을 기준으로 실린더를 Remove하고 아래 원판을 붙여야 한다. 전처리 과정을 통해 Narrow Step의 발생 위치가 Remove명령어를 수행한 솔리드 형상에서 발생하였음을 확인하고, 설계이력에서 Remove 명령어(C에서 Remove.3)와 Pad 명령어(Pad.2)의 위치를 변경하여 오류를 수정한다.

4. 결 론

B-Rep 형상을 기반으로 오류를 수정하는 것은 다음과 같은 두 가지 단점을 가지고 있다.

첫 번째는 위상요소와 기하요소 에 대한 변경을 통해 수정된 형상은 설계자의 직관적인 의도가 반영되어야 하는 모델링 과정에서 설계자의 의도와 다른 형상으로 왜곡되거나 형상 자체가 붕괴되는 위험이 있다.

두 번째는 비록 오류가 정확히 수정된다라도 B-Rep 형상은 깎이기 형태(Dumb solid)이기 때문에 수정된 결과가 설계 정보와 같이 생성된 Master CAD 시스템으로 반영되거나 이전 설계부서로 전달되기 어렵고, 단지 기준모델(Master model)이 아닌 참조모델(Reference model)로만 사용이 가능하기 때문에 활용도가 떨어진다.

본 논문에서는 이를 해결하는 방법으로 기존에 연구되지 않은 설계이력 기반의 오류 수정 방법을 제안하였다. 이를 위해 3단계의 접근 방법을 보여준다.

첫 번째는 추출된 설계이력 정보를 표현하기 위해 XML 기반의 설계이력 스키마를 정의하였다. 이 스키마는 본 논문에서 구현을 위해 사용한 CATIA V5뿐만 아니라 모든 상용 CAD 시스템에서 생성된 모델에도 동일하게 적용할 수 있다.

두 번째는 오류 수정을 위해 설계이력 정보 사이에 발생한 명령어 상호간에 의존관계에 대한 해석 방법을 제시하였다. 이를 위해 오류 발생 원인에 따른 명령어 패턴과 파라메트릭 데이터를 분석하여 오류 수정에 적용할 수 있었다.

세 번째는 설계이력을 기반으로 오류를 수정할 때

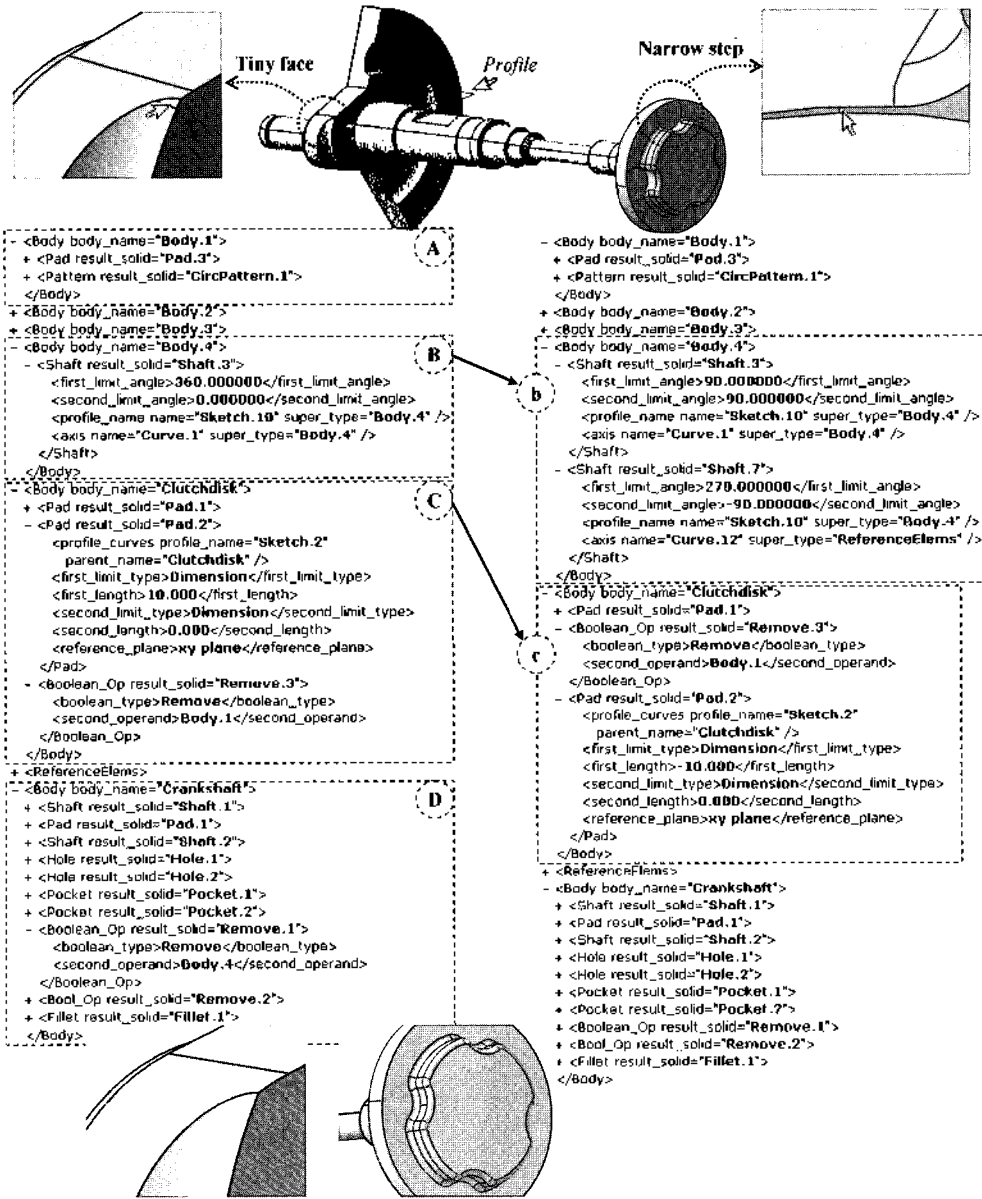


Fig. 13. Example of healing a crankshaft model.

필요한 프로세스를 제안하였다. 설계이력을 수정하기 위해서 각각의 설계이력 명령어의 의존관계에 대한 해석 방법과 이를 수정된 설계이력을 CAD 시스템에서 형상을 재 구성할 때 필요한 과정을 제시하였다.

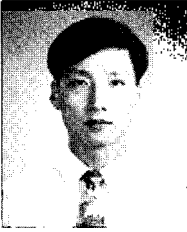
본 논문에서 구축된 오류 수정 툴을 활용해서 현업에서는 설계자가 모델링 하는 과정에서 실시간으로 설계방법론적인 오류를 검증할 수 있는 지능형 설계 시스템 개발에 응용할 수 있다.

향후 설계이력 기반의 오류 수정을 확장하기 위해

서는 많은 사례연구를 통해 지식베이스 구축이 필요하다. CAD 모델을 생성하는 것은, CAD 시스템에서 제공하는 기능의 문법적 이해를 바탕으로 설계자의 설계의도(Design intent)에 따라 전체 형상에 대한 설계상의 문맥적 의미(Contextual meaning)를 완성해 나가는 과정이다. 모델링 과정에서 설계자의 경험과 직관적 판단이 형상에 반영된다. 향후 추가로 구축될 지식베이스는 이러한 설계의도와 설계상의 문맥적 의미를 포함하는 연구가 필요하다.

참고문헌

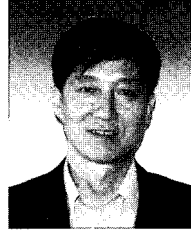
1. Tassey, G., "Interoperability Cost Analysis of the U.S. Automotive Supply Chain - Final Report", RTI Project Number 7007-03, Research Triangle Institute, 1999.
2. Japan Automobile Manufacturers Association, "PDQ (Product Data Quality: 모델 데이터 품질)", JAMA (<http://www.jama.or.jp>), 2003.
3. Hoffman, C. M. and Robert J. A., "CAD and the Product Master Model", *Computer-Aided Design*, Vol. 30, No. 11, pp. 905-918, 1998.
4. Gu, H., Chase, T. R., Cheney, D. C., Bailey, T. and Johnson, D., "Identifying, Correcting, and Avoiding Errors in Computer-aided Design Models Which Affect Interoperability", *Journal of Computing and Information Science in Engineering*, Vol. 1, pp. 156-166, 2001.
5. Deshpande, V., Fornasier, L., Gerteisen, E. A., Hilbrink, N., Mezentssev, A., Merazzi, S. and Wöhler, T., "Virtual Engineering of Multi-disciplinary Applications and the Significance of Seamless Accessibility of Geometry Data", *Future Generation Computer Systems*, Vol. 16, pp. 435-444, 2000.
6. Barequet, G., "Using Geometric Hashing to Repair CAD Objects", *IEEE Computational Science & Engineering*, pp. 22-28, 1997.
7. Steinbrenner, J. P., Wynman, N. J. and Chawner, J. R., "Procedural CAD Model Edge Tolerance Negotiation for Surface Meshing", *Engineering with Computers*, Vol. 17, pp. 315-325, 2001.
8. Volpin, O., Sheffer, A., Bercovier, M. and Joskowicz, L., "Mesh Simplification with Smooth Surface Reconstruction", *Computer-Aided Design*, Vol. 30, No. 11, pp. 875-882, 1998.
9. 박상호, 박종욱, 한순홍, 최 영, 양정삼, "IGES CAD 데이터의 교환에서 오류 진단 시스템", 한국정밀공학회논문집, 제20권, 제10호, pp. 218-225, 2003.
10. Yang, J., Han, S. H. and Park, S. H., "A Method for Verification of CAD Model Errors", *Journal of Engineering Design*, Vol. 16, No. 1, 2005. (To appear)
11. 양정삼, 한순홍, "STEP AP214 자동차 설계 데이터 정리 시스템", 한국CAD/CAM학회논문집, 제7권, 제3호, pp. 170-176, 2002.
12. AIAG D-15, Product Data Quality: Guidelines for the Global Automotive Industry, Technical Report, Automotive Industry Action Group Publications (<http://www.aiag.org/publications/d15.asp>), 2001.
13. Choi, K. H., Mun, D. H. and Han, S., "Exchange of CAD Part Models Based on the Macro-parametric Approach", *International Journal of CAD/CAM* (www.ijcc.org), Vol. 2, No. 2, pp. 23-31, 2002.
14. Mun, D., Han, S. and Oh, Y. C., "A Set of Standard Modeling Commands for the History-based Parametric Approach", *Computer Aided Design*, Vol. 35, pp. 1171-1179, 2003.
15. 양정삼, 한순홍, 김병철, 박찬국, "CAD 모델 교환을 위한 메트로 파라메트릭 정보의 XML 표현", 대한기계학회 논문집 A권, 제27권, 제12호, pp. 2061-2072, 2003.
16. CLIPS: A Tool for Building Expert Systems, <http://www.ghg.net/clips/CLIPS.html>
17. Rossignac, J. and O'Connor, M., "SGC: A Dimension Independent Model for Point Sets With Internal Structures and Incomplete Boundaries", *Geometric Modeling for Product Engineering*. Eds. M. Wosny, J. Turner, K. Preiss, North Holland, pp. 145-180, 1989.
18. SASIG PDQ Documentations V1.0, ISO TC184-SC4: SC4 N-DOCS, http://www.tc184-sc4.org/SC4_Open/SC4_and_Working_Groups/SC4_N-DOCS/1250-1499/maindisp.cfm?bk=7480.
19. CADFix: TranscenData, http://www.transcendata.com/products_cadfix.htm
20. Q-Checker: TransCAT Co., <http://www.q-checker.com/>
21. spGate: Armonicos, <http://www.armonicos.co.jp/spgate/index.html>
22. GeometryQA: Prescient Solutions, <http://www.prescientcorp.com/products/geometryqa.html>
23. CADHealer: Theorem Solutions, <http://www.theorem.co.uk/products/cadhealer.htm>
24. CADDdoctor: Elysium, <http://www.elysium.co.jp/products/caddoctor/index.html>



양 정 삼

2004년 한국과학기술원 기계공학과 박사
 1997년~2000년 고등기술연구원 주임연구원
 2001년~현재 (주)부품디비 연구개발팀장
 2002년 Clausthal University of Technology (Germany) 방문연구원
 2005년~현재 University of Wisconsin-Madison, Postdoctoral Associate

관심분야: Product data quality (PDQ), Product data exchange (PDE), Product data management (PDM), Geometric modeling, Virtual manufacturing



한 순 흥

한국과학기술원 기계공학과 교수이며, 웹저널인 International Journal of CAD/CAM(www.ijcc.org)의 편집장으로 활동하고 있다. 2003년까지 STEP센터(www.kstep.or.kr)의 회장과 전자거래학회(www.calsec.or.kr)의 회장을 맡았으며, 관심분야는 STEP, 가상현실 응용, 지능형 CAD이다. 연락처는 shhan@kaist.ac.kr, 홈페이지 <http://icad.kaist.ac.kr>, 미국 미시간 대학에서 1990년 박사학위