

JT를 이용한 CAE 해석결과 가시화

이옥렬[†] · 김재정^{*}

Visualization of CAE Analysis Results using JT

Ok-Lyeol Lee, Jay-Jung Kim

Key Words: CAE, JT, viewer, LOD, binary file, translator

Abstract

In the manufacturing industries, viewing CAE analysis results is frequently required during the product development process for design verification. CAE data which include all related information of an analysis is, however, not efficiently shared among designers because CAE data size is in general large to deal with. In order to increase collaboration among designers this paper introduces the development of a CAE visualization system based on JT format exploiting for a large model visualization with a scene graph-based toolkit. Since CAE analysis results and JT format have different structure we developed a translator to convert the CAE result in binary format to the JT format. To show the effectiveness of JT format in showing the CAE result we also developed a prototype viewer offering basic functions provided by commercial systems. By using JT format we are able not only to reduce the size of analysis results, but to store a series of analysis results with several LOD in a data file.

1. 서 론

최근 항공, 선박 및 자동차 등의 제조 산업분야에서 3차원 CAD의 보급이 확산되면서, 설계 품질을 향상시키기 위한 수단으로 설계 대상물의 물리적 특성을 시뮬레이션으로 평가하는 CAE 시스템을 도입하는 기업이 늘어나고 있다. 이러한 CAE 시스템의 도입은 제품 개발 초기에 발생하는 잦은 설계 변경 등으로 인한 문제점을 최소화할 수 있으며, 테스트 기간 단축 및 비용 절감효

과를 가져올 수 있다. 개발 초기에 설계 품질의 향상을 위해서는 해석 결과 시 발생하는 문제에 대한 효율적인 검토와 모델 수정이 이루어져야 한다. 이를 위해서는 CAD 설계자와 CAE 설계자 간의 유기적인 연결을 통해 해석 데이터 공유가 원활이 이루어 질 수 있는 협업 시스템이 구축되어야 한다.

하지만 CAE 시스템의 경우 CAD 시스템과는 달리 협업 시스템이 미흡하다. CAE 데이터 파일은 형상 정보 이외에 CAE 시스템에서 다루는 모든 해석 정보를 포함하고 있어 데이터양이 방대하다. 이러한 대용량 CAE 데이터 파일은 원활한 데이터 전송이 어려우며, CAE 뷰어를 통한 해석 데이터 공유를 어렵게 만드는 요인이 되고 있다. 이로 인해 현재 기업 내 모델 해석 결과에 대한 검토는 해석된 모델의 이미지를 복사하거나 혹은 수치적 데이터로만 확인하고 있다. 또한 설계 변경 시 CAD 설계자는 수치적 결과만을 바탕으로

[†] 한양대학교 대학원 기계공학과

E-mail : king123400@hotmail.com

TEL : (02)2220-0572 FAX : (02)2292-7758

^{*} 한양대학교 기계공학과

제품 모델을 수정하는 비합리적인 작업이 이루어지고 있는 실정이다.

따라서 대형 CAE 데이터 파일의 경량화와 해석 데이터의 가시화는 상호간(CAE 설계자와 검토자 혹은 CAD 설계자와 CAE 설계자)의 해석 결과에 대한 검토 및 모델 수정에 있어 매우 효율적이라 할 수 있으며, 여러 형식의 정보를 공유해야 하는 기업 환경의 구현에 필요한 부분이라 할 수 있다.

1.1 관련 연구

과거 CAD 시스템의 협업 환경 구축 관한 연구들은 데이터 교환의 목적으로 DXF(Data Exchange File), IGES(Initial Graphics Exchange Specification), STEP(Standard for the Exchange of Product model data)등의 중립 데이터 형식에 초점을 맞추어 진행되어져 왔다. 이들 중립 포맷은 다양한 CAD 시스템간의 데이터 교환이 원활히 이루어지는 않는 문제점과 직접 번역기를 이용한 데이터 교환의 문제점을 해결하고자 제안되었다.[1][2]

중립 포맷을 이용한 시스템은 기업 내 여러 분야에서 정보를 공유를 통한 협업 시스템을 구축하는데 활용되었다.[3][4] 하지만 이와 같은 중립 포맷들은 다음과 같은 한계를 지니고 있다. DXF는 설계도면 데이터를 교환하는 목적으로 개발되었으므로, 3차원 솔리드 모델 데이터에 적용할 수 없다는 단점이 있다. IGES는 3차원 모델 데이터의 교환이 가능하지만, 사용자가 원하는 만큼의 데이터 변환이 이루어지지 않으며, 형상데이터 이외의 정보들은 포함할 수 없다는 단점이 있다. 최근 가장 많은 연구가 이루어지고 있는 STEP의 경우 IGES보다 안정된 데이터 변환이 이루어지지만, 해석 정보들을 정의하고 있는 STEP AP209의 방대한 양과 복잡성, 또한 아직 모든 해석정보들의 표준이 정립되어 있지 않아 CAE 시스템의 협업 구축에 적용시키기는 어렵다.[6][7]

다른 연구로는 사출 금형 협업 설계 구현을 위해 CAE 가시화 시스템을 진행한 연구가 있다. 이 연구는 중립포맷을 이용한 시스템의 한계를 인지하고 상용 CAE 시스템에서 직접 아스키(ASCII) 데이터를 추출하여 가시화 시스템을 개발하였다.[8] 하지만 아스키 데이터는 Pre/Post Processing 정보에 대한 풍부한 정보를 얻을 수

없으며, CAE 시스템의 업데이트로 아스키 데이터의 정보구조 변경 시에 다시 개발해야 하는 단점이 있다.

1.2 연구 목적

본 논문에서는 CAE 해석 결과의 경량화와 가시화 기능의 강화를 통하여 결과의 공유성과 관리의 편리함을 증대하는데 그 목적을 두고 있다. 이를 위하여 CAD 분야에서 뷰어의 중립 포맷으로 사용되고 있는 JT를 CAE 해석결과를 표현하는 포맷으로 채택하여 공유를 통한 협업 환경 구축을 위한 방법으로 적용시켰다. 또한 공유를 위해 예제로 사용된 CAE 해석결과는 구조해석에 널리 사용되는 ANSYS 결과를 사용하였다.

JT는 대형의 모델을 신속하고 효율적인 가시화의 목적으로 개발되어 세계적으로 널리 사용되는 경량화된 3차원 모델 데이터 형식이다. 또한 모델의 형상 정보 및 여러 속성 정보 등의 풍부한 정보를 담을 수 있는 관계로 현재 CAD 산업에서 널리 사용되고 있다. 그리고 CAE 산업에서도 NX Nastran 5.0경우에 JT 포맷이 내장되어 사용되어지고 있다.[9] JT에 관한 자세한 설명은 다음장에서 하기로 한다.

본 논문에서는 아스키 데이터 대신 ANSYS Binary 데이터를 활용하여 JT 포맷으로 변환하였다. ANSYS 데이터와 JT 데이터는 서로 다른 구조로 이루어졌기 때문에 JT 포맷으로의 변환에 있어 별도의 번역 과정이 필요하다. Fig.1은 JT 포맷으로의 번역 과정의 개략 도를 나타낸 것이다.

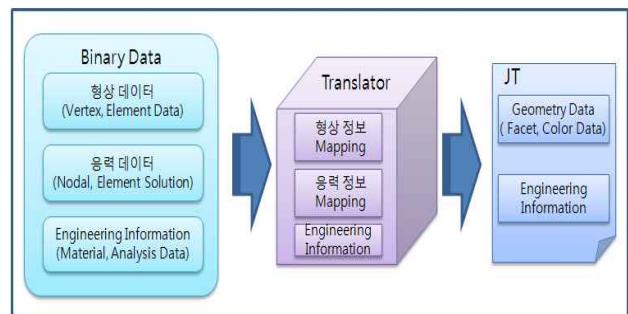


Fig.1 Translation Process into JT Data

JT를 활용함으로써 첫째, 상호간(CAE 설계자와 검토자 혹은 CAD 설계자와 CAE 설계자)에 필요한 정보만을 JT에서 제공하는 LOD 공간에 담음

으로써 해석 데이터를 경량화 할 수 있다. 둘째, 한 제품에 대해 여러 차례 해석한 결과를 JT 파일 생성 방법을 이용해 하나의 JT파일로 생성할 수 있다. 셋째, CAE 해석 결과를 신속하고 효율적으로 가시화 할 수 있다.

2. 배경

2.1 JT

JT는 1998년 효율적인 가시화의 목적으로 HP와 Engineering Animation Inc.의 “Jupiter Project”에 의해 개발되어 세계적으로 널리 사용되는 경량화된 3차원 모델 데이터 형식으로써, 대형 모델을 신속하고 효율적으로 가시화할 수 있는 장점이 있다. JT 데이터 형식 대형 모델을 신속하고 효율적으로 가시화시킬 수 있는 이유는 Jupiter 장면 그래프(Scene graph)의 Occlusion, Visibility Culling, Open GL Optimizer, 데이터 압축 기술이 지원되기 때문이다. 또한 Level of Detail(LOD)를 지원하므로 각 단계별로 빠른 가시화를 제공한다.

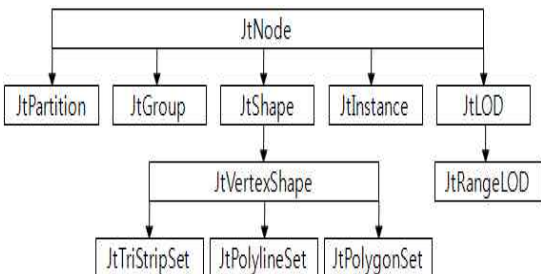


Fig.2 Jupiter Node Class Hierarchy

Jupiter 장면 그래프는 Fig.2에서 보듯이 다양한 노드(Node)로 구성되어 있다. 이 노드들은 객체(Object) 또는 그룹 모델의 파티션(Partition)을 나타내는데 사용된다. Fig.3은 Jupiter 장면 그래프 모델이 다양한 노드들로 구성된 예이다. 이를 통해 JT 모델은 단계(Level)별 LOD 그룹을 갖고 있음을 확인할 수 있다.

Fig. 4는 Jupiter 장면 그래프내의 파이프라인 에이전트들(agents)간의 기능 수행에 관련된 프로세스를 나타낸다. 에이전트들의 요청에 따라 각자의 기능을 수행하는데 render agent (Jt Range LOD Agent), range level of detail agent, (Jt View Frustum Culling Agent), frustum culling agent,

occlusion Culling agent(Jt Occlusion Agent)등으로 분류된다.[10]

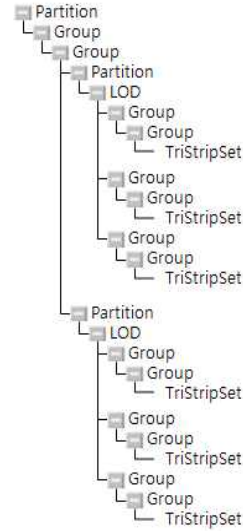


Fig.3 Jupiter Scene Graph

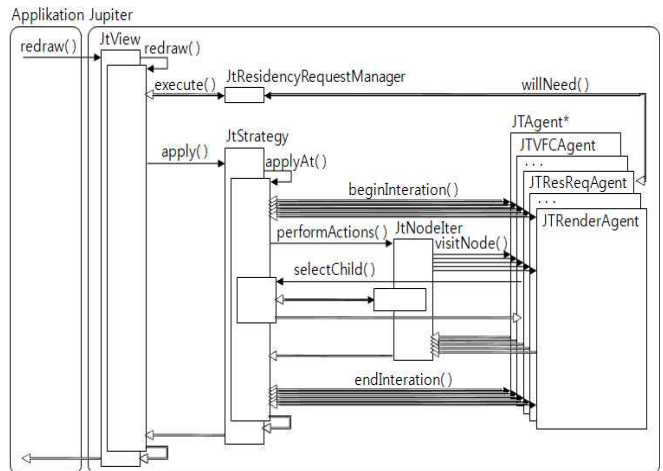


Fig.4 A Strategy calls a Set of Pipelined Agents Which process the Scene Graph

3. CAE 데이터를 JT 데이터로 변환

3.1 정보 추출 방법

CAE 데이터를 JT 데이터로 변환 과정에 대해서는 서론에서 이미 소개했듯이, CAE 시스템에서 형상 및 여러 속성 정보를 추출하여 JT 포맷으로 매핑 및 변환하는 과정이 필요하다. CAE 시스템은 아스키와 Binary의 파일 형식으로 데이터를 저장하며 이 데이터들을 활용해 정보를 추출할 수 있는 방법이 다음 두 가지가 있다.

첫 번째로 CAE 시스템에서 아스키 데이터를 활용하여 JT 포맷으로 변환하는 방법이 있다. 아

스키 데이터는 사용자가 해독 가능한 문자 형태여서 코어(Core)가 없어도 필요한 정보의 추출이 가능하며, 해석 결과에 대해 명확한 해석이 가능하다. 하지만 여러 개의 해석 결과 파일을 사용자가 직접 관리하며, 해석 결과 파일들을 얻기 위해서는 해당 CAE 시스템을 구동시켜야 된다. 또한 한 제품에 대해 여러 번의 해석을 하기 때문에 해석 결과 변경 시 새로운 해석 결과에 대한 파일들을 사용자가 일일이 관리해야 하는 단점이 있다.

두 번째로 CAE 시스템에서 Binary 데이터를 활용하여 JT 포맷으로 변환하는 방법이 있다. 이러한 방법은 하나의 파일로 모든 해석 결과 정보의 추출을 가능하게 하며, Pre/Post Processing 정보도 담고 있어 아스키 데이터에 비해 풍부한 정보를 얻을 수 있게 해준다. 또한 모델의 해석과 동시에 해석 결과 파일이 Binary 파일로 자동 저장된다. 따라서 해당 CAE 시스템의 구동이 필요 없으며 해석 결과 변경 시 사용자는 변경되어진 하나의 파일만 관리하면 된다. 하지만 Binary 형태가 포트란(FORTRAN)으로 되어있어 필요한 정보를 추출하기 위해서는 포트란에 대한 이해가 필요하다. 또한 Binary 구조가 복잡해 구조 분석에 많은 시간이 걸린다는 단점이 있다. Fig. 5는 Binary 데이터의 일부분을 나타낸 것이다.

Node=1804	xyz=	0.97653	0.14815	0.50000E-01	0.00	0.00	0.00
Node=1805	xyz=	0.93199	0.13365	0.15000	0.00	0.00	0.00
Node=1806	xyz=	0.91183	0.12135	0.50000E-01	0.00	0.00	0.00
Node=1807	xyz=	0.91183	0.12135	0.10000	0.00	0.00	0.00
Node=1808	xyz=	0.77500	0.00000	0.10000	0.00	0.00	0.00
Node=1809	xyz=	0.77500	0.00000	0.15000	0.00	0.00	0.00
Node=1810	xyz=	0.80000	0.00000	0.50000E-01	0.00	0.00	0.00
Node=1811	xyz=	0.77909	0.12952	0.10000	0.00	0.00	0.00
Node=1812	xyz=	0.77909	0.12952	0.15000	0.00	0.00	0.00
Node=1813	xyz=	0.78216	0.17204	0.50000E-01	0.00	0.00	0.00
Node=1814	xyz=	0.85795	0.14205	0.10000	0.00	0.00	0.00
Node=1815	xyz=	0.85795	0.14205	0.15000	0.00	0.00	0.00
Node=1816	xyz=	0.88387	0.15351	0.50000E-01	0.00	0.00	0.00
Node=1817	xyz=	0.35000	-0.20000	0.20000	0.00	0.00	0.00
Node=1818	xyz=	0.40000	-0.50000E-01	0.20000	0.00	0.00	0.00
Node=1819	xyz=	0.40000	-0.10000	0.20000	0.00	0.00	0.00
Node=1820	xyz=	0.75000	-0.50000E-01	0.50000E-01	0.00	0.00	0.00
Node=1821	xyz=	0.75000	-0.50000E-01	0.10000	0.00	0.00	0.00
Node=1822	xyz=	0.75000	-0.50000E-01	0.15000	0.00	0.00	0.00
Node=1823	xyz=	0.10000	-0.25000	0.15000	0.00	0.00	0.00
Node=1824	xyz=	0.50000E-01	-0.25000	0.50000E-01	0.00	0.00	0.00
Node=1825	xyz=	0.50000E-01	-0.25000	0.10000	0.00	0.00	0.00


```

*** Elements
IElen= 1 nNode= 8 ElenData= 1 2 1 1 0 0 1a 0 1 0
Nodes= 1 3 41 40 462 474 582 579
IElen= 2 nNode= 8 ElenData= 1 2 1 1 0 0 1a 0 2 0
Nodes= 462 474 582 579 463 475 583 580
IElen= 3 nNode= 8 ElenData= 1 2 1 1 0 0 1a 0 3 0
Nodes= 463 475 583 580 464 476 584 581
IElen= 4 nNode= 8 ElenData= 1 2 1 1 0 0 1a 0 4 0
Nodes= 464 476 584 581 366 368 406 405
IElen= 5 nNode= 8 ElenData= 1 2 1 1 0 0 1a 0 5 0
Nodes= 3 4 45 41 474 477 594 582
IElen= 6 nNode= 8 ElenData= 1 2 1 1 0 0 1a 0 6 0
Nodes= 474 477 594 582 475 478 595 583
IElen= 7 nNode= 8 ElenData= 1 2 1 1 0 0 1a 0 7 0
Nodes= 475 478 595 583 476 479 596 584
IElen= 8 nNode= 8 ElenData= 1 2 1 1 0 0 1a 0 8 0
Nodes= 476 479 596 584 368 369 410 406
IElen= 9 nNode= 8 ElenData= 1 2 1 1 0 0 1a 0 9 0
Nodes= 4 5 49 45 477 480 606 594
    
```

Fig.5 Structure of the Binary Data

3.2 Binary 데이터의 정보 번역

CAE 시스템에서 추출되어진 Binary 데이터를 활용해 해석 결과를 가시화하기 위해선 크게 두

부분으로 나눌 수 있다. 모델의 형상 가시화와 해석 결과에 대하여 결과 값의 영역별로 색상을 이용하여 표시해 주는 색상 가시화이다. CAE 시스템에서 모델의 형상은 유한개로 구성되어 있는 노드(Node)와 각 노드들 사이에 서로 유기적인 관계를 맺어 주는 메쉬(Mesh)들의 집합으로 나타내어진다. 메쉬는 2D 모델과 3D 모델에 따라 다른 형태들로 되어있으며, CAE 시스템에서 정의한 방향과 순서에 의해 구성되어있다. 2D 모델일 경우 삼각형과 사각형 형태이며, 3D 모델일 경우 사면체와 육면체 형태로 구성되어있다. fig.6은 메쉬를 구성하는 노드들의 상관관계를 나타낸 것이다.

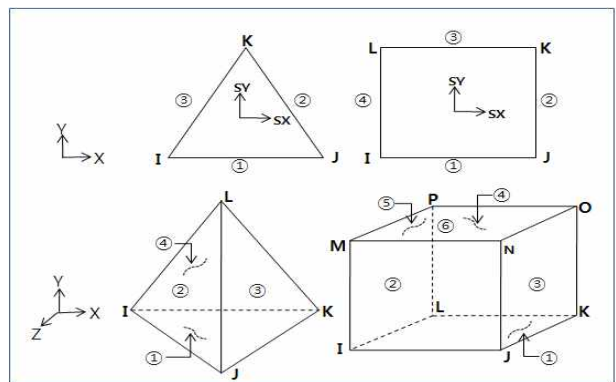


Fig.6 Element Description of Meshes

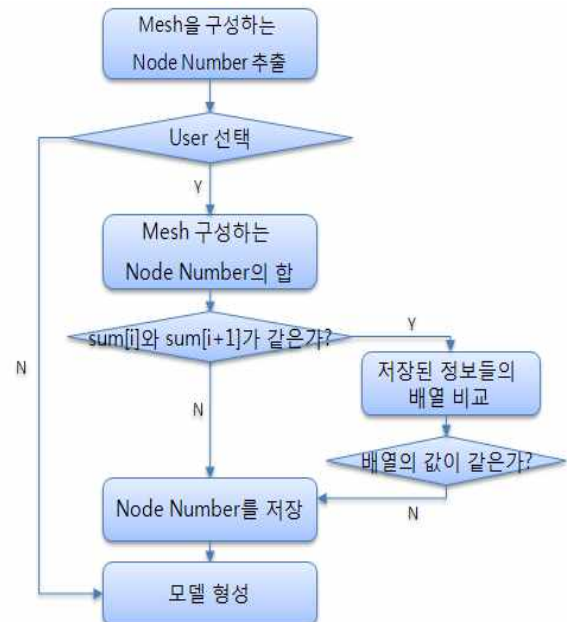


Fig. 7 Flowchart of the Removing Inside Meshes

Fig. 7 모델의 형상 내부 메쉬 제거는 순서도를 나타낸 것으로 모델의 형상 내부에 있는 메쉬를 사용자의 선택에 의해 제거할 수 있게 구현하였

다. 해석 결과의 검토에 있어서 모델의 내부보다는 외부 형상이 필요하다. 반면에 모델 수정에 있어서는 외부 형상뿐 아니라 모델 내부의 해석 결과도 필요로 한다. 따라서 CAE 설계자는 상황에 따라 해석 데이터 파일을 생성함으로써 파일의 크기 및 불필요한 메모리 공간을 줄일 수 있다.

결과 검토에 있어서 사용자가 해석을 통해 계산된 결과들을 분석해야 할 경우, CAE 시스템은 구조물 전체에 걸친 응력이나 변위 등을 일정한 색상으로 표현하여 시각적으로 분석할 수 있게 해준다. 본 논문에서 색상 가시화를 위해 다음과 같은 과정으로 진행하였다.

1. Binary 데이터에서 추출한 모든 종류의 결과 값들을 임시 공간에 저장한다.
2. 사용자의 선택에 의해 필요한 결과 값만 추출한다.
3. 각각의 결과 값들에서 Max값과 Min값을 구분한다.
4. Max값과 Min값 사이를 색상별로 각 구간의 범위 및 단계를 나눈다.
5. 각 구간의 결과 값들을 색상별로 매핑 한다.

3.3 JT 포맷으로 변환

JT는 Fig. 8에 보이는 것처럼, JT 파일을 생성할 때 세 가지 방법으로 생성될 수 있다. 첫 번째는 모델의 트리 구성과 동일하게 파일과 디렉토리를 생성하는 방법이고, 두 번째는 하나의 디렉토리에 모든 파일들을 생성하는 방법이다. 그리고 마지막은 모든 데이터를 포함하는 하나의 파일로 생성하는 방법이다.[11]

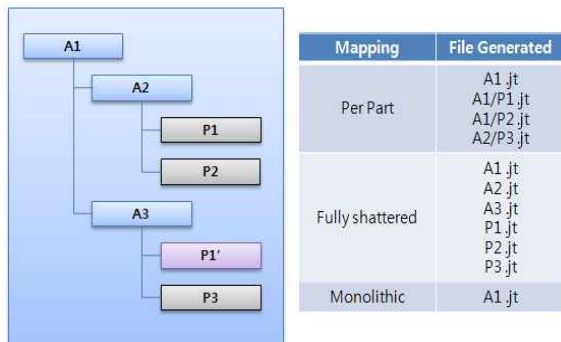


Fig. 8 JT File Structure

본 논문에서는 위의 세 가지 방법 중 Monolithic 방법을 이용해 한 제품에 대해 여러

차례 해석한 결과들을 담고 있는 파티션들을 연결해 하나의 JT파일로 생성하였다. 각 파티션들은 LOD 그룹들로 구성되어 있으며, 각각의 LOD 안에는 Binary 데이터에서 추출되어진 모델의 형상 정보 및 해석 결과 정보들이 담겨져 있다. Fig. 9는 JT파일 안에 담겨져 있는 정보들의 계층 구조를 나타낸다.

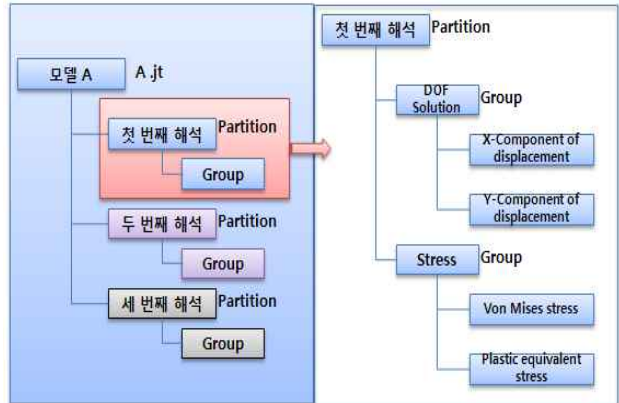


Fig. 9 CAE Result Data Hierarchy

4. 구현

해석 결과 가시화를 위해 ANSYS10.0의 Binary 데이터로부터 정보를 추출하였다. JT데이터의 생성을 위해서는 UGS의 JT Open Toolkit Version 4.0을 이용하였으며, 추가로 JT 포맷으로 변환하는 번역기를 개발하여 이용하였다. 개발환경은 Microsoft Windows XP SP2 이며 사용된 언어는 MS Visual studio NET 2003이다.

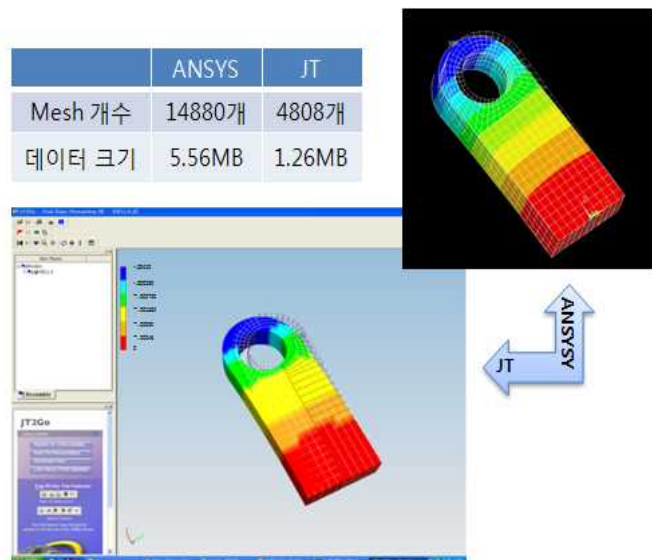


Fig. 10 An Example of CAE Result Data from ANSYS

Fig. 10은 ANSYS에서 구조 해석한 후 DOF (Degree Of Freedom)의 결과 값을 JT 포맷으로 변환하여 가시화 한 것이다. 그림에서 보이는 것처럼 JT 변환 후 메쉬 개수 및 데이터의 크기가 줄어들었다는 것을 알 수 있다. 또한 색상 구간별로 결과 값을 한눈에 볼 수 있는 색상바(Bar)를 표시함으로써 해석 결과의 검토 및 수정을 용이하게 하였다.

5. 결론 및 향후 연구과제

본 연구에서는 CAE 분야에서 협업 구축이 이루어지지 않는 문제점을 인지하고 JT를 이용해 협업 시스템 환경을 구현하였다.

첫째는 해석 모델의 형상과 해석 결과를 색상으로 표현하기 위해 필요한 정보들을 구분하여 정의한 다음 JT에서 제공하는 LOD 공간과 내부 메쉬를 제거하는 알고리즘을 이용하여 대용량 CAE 데이터를 경량화 하였다. 이로써 원활한 데이터 전송을 이룰 수 있으며, 해석 데이터의 공유가 이루어질 수 있다. 둘째는 JT 파일 생성 방법 중 Monolithic 방법을 이용하여 한 제품에 대해 여러 차례 해석한 결과를 하나의 JT 파일로 생성함으로써 설계 지식 누적 및 파일 관리가 용이해진다. 마지막으로 JT 포맷으로 변환되어진 해석 결과의 가시화를 통해 고가의 사용 CAE 시스템 도입 없이도 상호간(CAE 설계자와 검토자 혹은 CAD 설계자와 CAE 설계자)의 효율적인 검토 및 설계 변경을 할 수 있다.

향후 연구 과제로 본 논문에서 사용한 CAE 시스템 이외에 다양한 해석 분야에 사용되어지고 있는 CAE 시스템들의 데이터를 JT 데이터로의 변환에 관한 연구가 필요하다. 또한 JT는 운용 특성상 대형 모델을 가시화에 기대 효과가 큰 만큼, 범용 해석 결과 전용 뷰어의 개발이 필요하다.

참고문헌

(1) Reed, K., Harrod, D. and Conrooy, W., The Initial Graphics Exchanges Specification (IGES) Version 5.0, NIST, 1990.
 (2) Helpenstein, H. J., CAD Geometry Data Exchange Using STEP, Springer-Verlag, 1993.

(3) Lee, Y. J., Ko, K. W. and Yoo, S. B., 1996, "Implementation of CAD Data Translation System using STEP", *International Journal of CAD/CAM*, Vol.1, No.2, pp.87~96.
 (4) Baek, J. H. and Min, S. J., 2007, "STEP-Based Information Exchange for Structural Analysis and Optimization", *International Journal of CAD/CAM*, Vol.12, No.1, pp.8~14.
 (5) Lee, J. C., Kim, J. H. and Han, S. H., 1999, "Visualization of STEP Assembly Model on the Web", *International Journal of CAD/CAM*, pp.295~299.
 (6) Deng, Y. M., Lam, Y. C., Tor, S. B. and Britten, G. A., 2002, "A CAD-CAE Integrated Injection Molding Design System", *Engineering with Computers*, Vol.18, No.1, pp.80~92.
 (7) Tanaka, F. and Kishinami, T., 2006, "STEP-Based Quality Diagnosis of Shape Data of Product Models for Collaborative E-engineering", *Computers in Industry*, Vol.57, No.3, pp.245-260.
 (8) Kim, H. G., Song, I. H. and Chung, S. C., 2007, "Development of the CAE Visualization System for Collaborative Injection Mold Design", *International Journal of CAD/CAM*, pp. 667~673.
 (9) Bang, H. C., Suh, K. C., and Kim J. J., 2007, "Development of a translator from ACIS data to JT data", *International Journal of CAD/CAM*, pp. 343~352.
 (10) Bartz, D., 2001, "Jupiter: A Toolkit for Interactive Large Model Visualization", IEEE Symposium on Parallel and Large Data Visualization and Graphics proceedings.
 (11) UGS, JT Open Toolkit v4.0.0.0 Online Document, 2007.