

# CAD 뷰어 기반 대화형 3D PDF 문서 생성 자동화

박경호\*, 최 영#, 양상욱\*\*, 송인호\*\*\*

## Automatic Generation of Interactive 3D PDF Document in a 3D Viewer Environment

Kyeong Ho Park\*, Young Choi#, Sang Wook Yang\*\* and In Ho Song\*\*\*

### ABSTRACT

PDF is widely accepted as a standard document format and now it supports 3D contents as well. Within the engineering application areas, this new 3D feature may be used to support sharing of 3D documents and thus collaboration between engineering departments, suppliers and partners. In this paper, we describe a system that automatically generates formatted engineering documents including 3D data converted from 3D applications such as commercial 3D CAD viewer. The system consists of two major modules. One is U3D conversion module and the other is PDF conversion module. U3D conversion module extracts geometry, view data, assembly and disassembly information from 3D viewer and converts to U3D format, currently in IDTF text file format. PDF conversion module generates a PDF file and inserts U3D data, various annotation information, and scripts for custom generated operations such as assembly and disassembly in the PDF document.

**Key Words** : 3D PDF (3 차원 PDF), U3D, Acrobat 3D

### 1. 서론

제품 개발에서 3D CAD의 도입이 증가하고 있으며 엔지니어링 정보에서 3D정보의 공유를 위하여 PDM 및 뷰어를 활용하고 있지만, 도면이나 일부 정보에만 한정된 공유가 가능하다. 대부분의 엔지니어링 정보는 문서로서 만들어지게 된다. CAD와 연계한 문서작업을 위하여 상용 CAD의 HTML 문서 생성 기능이나 XML 기반의 문서 생성기능을

갖는 PTC의 Arbortext<sup>1</sup>의 경우 CAD와 연결하여 문서를 편집하는 기능을 갖지만, 문서상의 3D 설계 결과물을 표현하기 위하여 화면상의 이미지를 캡처하여 편집하는 경우가 대부분이다. 설계 정보의 명확한 공유를 위해서 이미지보다는 실제 3D 모델이 필요한 경우가 있다. Adobe Systems<sup>2</sup>에서 PDF 문서에 3D 데이터를 포함시킬 수 있는 Acrobat3D를 발표하면서 별도의 3D 뷰어 없이 Adobe Reader를 통하여 PDF 문서내의 3D를 검토

접수일: 2007년 3월 29일; 게재승인일: 2007년 11월 19일

\* 중앙대학교 기계공학부, LG 전자

# 교신저자: 중앙대학교 기계공학부

E-mail: yychoi@cau.ac.kr Tel. (02) 820-5312

\*\* 중앙대학교 기계공학부

\*\*\* 아주대학교 기계공학부

할 수 있게 되었다. PDF는 전자문서로서의 보존성 및 보안성을 갖추고 있으며, 국제표준문서 포맷으로서 범용성 및 플랫폼 독립성, 사용자와의 상호작용이 가능한 장점을 가지고 있다. 이러한 PDF 문서를 통하여 엔지니어링 정보의 효율적인 관리 측면뿐만 아니라 관련부서 및 협력업체로의 3D 정보의 직접 배포에 비하여 배포 및 보안관리가 용이하게 되었다. 또한, 문서와 결합된 형태의 3D 설계 정보를 활용하면 더 효율적인 커뮤니케이션이 가능하다고 할 수 있으며, 마케팅, 서비스 등 3D 정보가 필요한 여러 분야에서 3D 설계 정보의 활용이 가능하다.

본 연구에서는 사용자와 인터랙션이 가능한 3D 정보가 포함된 PDF문서를 자동화하는 기술을 개발하고, 상용 뷰어인 이눅스사의 DynaView<sup>3</sup>에 적용하였다. 뷰어에서 다루고 있던 3D 모델에 대한 형상 정보 및 뷰정보, 조립정보, 분해정보를 그대로 PDF로 출력하게 되고, 사용자가 조립체의 분해, 조립에 대한 조작이 가능하도록 하였다.

## 2. 관련 표준 기술

### 2.1 U3D

PDF는 3D와 같이 복잡한 형태의 데이터를 통합할 수 있는 프레임워크를 제공하기 위하여 3D 표준으로 U3D(Universal 3D) 포맷을 채택하였다.

U3D는 여러 종류의 상용 CAD와 3DMax, 그리고 기타 3D 소프트웨어 개발자들이 제안한 각기 고유의 3D 포맷을 단일 포맷으로 대체하기 위하여 2004년도에 Intel을 중심으로 Adobe 등 24개의 회사가 참여한 3D 산업 포럼 (3DIF: 3D Industry Forum)에서 발표한 표준이다. 현재 3번째 버전까지 확장되었으며, 유럽컴퓨터제조업자협회(ECMA)에서 ECMA 363 표준으로 지정하였다.<sup>4</sup>

U3D는 설계에서 사용된 3D Data를 마케팅, 전자문서화, 조립지시, 사용자 훈련, 고객 서비스 등 다양하게 활용할 수 있다.

U3D 특징으로는 3D 데이터의 스트리밍을 지원하여 다운로드하는 동안 점진적으로 뷰잉이 가능하도록 CLOD(Continuous Level of Detail) 메시를 지원하고, 그 외에 파일 포맷의 확장, 파일의 압축, 애니메이션, 텍스처 맵핑, 음영처리기능을 가능하게 한다.<sup>5</sup>

## 3. 3D PDF 자동 생성 시스템

### 3.1 시스템 개요

본 연구는 3D 데이터를 포함하는 PDF문서를 자동으로 생성하면서 동시에 문서 내의 3D 데이터를 사용자가 대화형으로 조작할 수 있는 기능을 자동화하는 기술 개발을 목적으로 하였다. 기술 개발의 구현은 상용 뷰어인 DynaView상의 메뉴에서 호출되며 다음과 같은 두 가지 목표를 가지고 시스템을 개발하였다.

첫째, 3D 뷰어 상에서 3D PDF를 저작하는 모듈을 개발하여, DynaView 에서 현재 다루고 있는 3D 데이터를 PDF 문서로 출력할 수 있도록 하는 기능을 지원하는 모듈을 구현한다. 이로써, 뷰어 사용자는 상용 CAD 시스템이나 Acrobat3D가 없더라도 DynaView만으로도 3D PDF 문서를 저작할 수 있도록 한다.

둘째, 3D PDF의 조립 분해 기능 구현으로, 3D 형상뿐 아니라, 어셈블리 정보를 포함하여 부품의 분해 및 조립을 표현 할 수 있다. 따라서 개발된 모듈 또한 DynaView에서 표현된 어셈블리 정보가 PDF에서 그대로 표현이 가능하도록 적절히 변환할 수 있어야 하며, 문서에서 조립체의 분해 조립이 가능하도록 해야 한다.

위와 같은 3D가 포함된 PDF 문서를 작성하기 위해서는 일반적으로 Acrobat 3D 및 3D Toolkit을 이용하여 3D CAD 데이터의 캡처 및 편집을 하고, Microsoft Office 내에서 ActiveX 컨트롤을 이용하여 3D를 삽입하고, 문서를 작성하여, PDF로 변환하는 과정을 거치게 된다.

그러나 엔지니어링 문서 중에서 작성 빈도가 높은 문서들은 포맷이 정형화되어 있고, 텍스트로 작성되는 내용은 데이터베이스로 관리되는 경우가 많다. 이러한 문서를 매번 Acrobat3D로 편집하는 경우 작업자의 시간적인 낭비뿐만 아니라 작성의 오류가 발생할 수 있다. Fig. 1은 3D 원본 데이터로부터 3D PDF 문서를 생성하는 두 종류의 절차를 보여준다. 한 가지는 Acrobat3D 툴킷을 이용하여 U3D로 변환 하는 과정이지만 본 논문에서는 이 방법을 사용하지 않고 Fig. 1의 점선으로 표기된 절차와 같이 DynaView 의 내부 포맷 데이터를 중립 포맷으로 변환하여 사용자와 인터랙션이 가능한 3D 정보가 포함된 PDF문서 생성을 자동화하는 기술을 개발하였다.

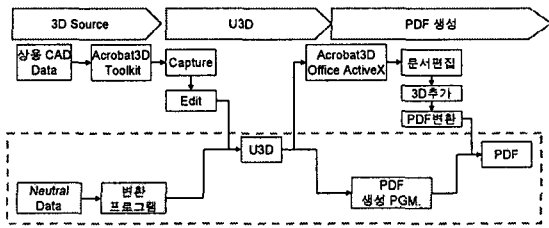


Fig. 1 3D PDF document generation process from 3D source

Fig. 2는 개발된 시스템의 전체 개요를 나타내고 있다. 3D PDF 자동 생성 시스템은 그림의 가운데 보이는 바와 같이 U3D 변환 모듈과 PDF 변환 모듈로 구성된다. 이러한 2개의 변환모듈은 기술적 구분이고, 실제로 사용자의 관점에서는 DynaView에서 3D 데이터를 다루면서 PDF생성 메뉴를 선택하면 하나로 연결된 상태에서 동작되어, U3D변환과 조립체의 분해조립이 가능한 스크립트가 포함된 PDF변환이 자동적으로 진행된다.

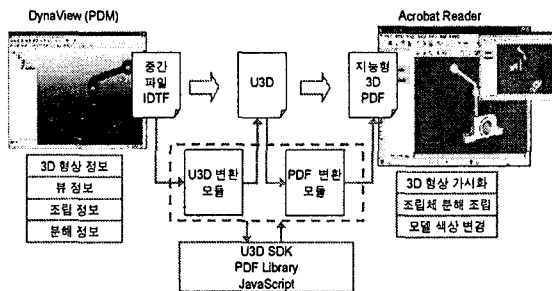


Fig. 2 Overview of PDF document generation system

U3D 변환 모듈은 DynaView의 3D 형상 정보, 뷰 정보, 조립 정보, 분해 정보를 U3D 포맷으로 변환한다. U3D 변환 모듈은 DynaView에서 U3D의 중간 포맷의 텍스트 파일로 출력하고 U3D SDK를 이용하여 U3D로 변환을 한다.

PDF 변환 모듈은 U3D 포맷을 거쳐서 최종적으로 3D 형상 및 조립체의 조립 및 분해가 가능한 대화형 PDF 문서로 변환한다. PDF 변환 모듈은 PDF Library를 이용하여 PDF 문서를 생성하고, 변환된 U3D 파일을 PDF 문서에 추가하며 조립체의 분해, 조립 애니메이션이나 색상 등의 속성 변경을 위한 사용자의 인터랙션을 추가하기 위하여 자바스크립트를 활용한다.

### 3.2 개발 도구

#### 3.2.1 Acrobat PDF Library

PDF를 위한 API (Application Programming Interface)로 Acrobat SDK와 Acrobat PDF Library SDK(PDF Library)가 있다. Acrobat SDK는 PDF 문서에서 Acrobat plug-in의 지원 및 디지털 서명, 코멘트 추가 등 PDF Library에 비해 많은 기능을 가지고 있지만 Acrobat 소프트웨어가 설치된 환경에서 사용이 가능하다. 그러나 PDF Library는 Acrobat이 설치되지 않은 환경의 외부 프로그램에서 PDF 문서의 생성, 편집, 조합, 출력 등이 가능하도록 개발환경을 제공해준다. 본 연구에서는 뷰어나 CAD, PDM 같은 외부 어플리케이션에서 정형화된 PDF 문서를 자동 생성하는 것이 목적이기 때문에 PDF Library를 활용하였다. Fig. 3은 Acrobat SDK와 PDF Library에서 제공하는 API의 개요로서, AV Layer 부분은 PDF SDK에서만 활용 가능한 라이브러리이며, 나머지는 PDF Library에서 함께 활용할 수 있는 라이브러리이다.<sup>6</sup>

|  |                             |   |
|--|-----------------------------|---|
| Acrobat Viewer (AV) Layer<br>(menu selections, other user-level actions)         |                             | Acrobat Support (AS)<br>(file access, platform-independent utilities, Callbacks, exceptions, ...) |
| Portable Document (PD) Layer<br>(bookmarks, pages, thumbnails, annotations, ...) |                             |   |
| PDFEdit<br>(page content)  | PDSEdit<br>(structure info) |   |
| Cos Layer<br>(strings, numbers, dictionaries, ...)                               |                             |   |

Fig. 3 Overview of Acrobat Core API

#### 3.2.2 Acrobat 3D

Acrobat3D<sup>7</sup>는 3D PDF 문서의 작성 기능을 다양하게 제공하고 있지만, 기본적으로 사용자 UI를 제공하는 프로그램으로, 원하는 포맷의 문서로 자동 생성을 할 수는 없다. 본 연구는 3D PDF를 자동 생성하는 것을 목적으로 하기 때문에 템플릿 PDF문서를 작성하기 위한 수단으로만 Acrobat3D를 활용하였다. 참고적으로 Acrobat3D는 Adobe Reader에서 제공되지 않는 3D 단면, 치수 측정 등의 뷰잉 기능이 추가되어 있고 Microsoft의 파워포인트, 엑셀, 워드에서 3D를 삽입하여 PDF로 출력하는 기능 및 툴킷을 이용하여 애니메이션, 질감 설정 등의 편집 기능 등을 수행할 수 있다.

### 3.2.3 U3D SDK

U3D<sup>4</sup> 파일 구조는 Fig. 4와 같이 씬 그래프 구조를 가지며, 노드, 리소스, 노드와 리소스를 연결하기 위한 팔레트로 구성이 된다. 각각의 노드는 부모 노드를 갖는 계층구조로 정의되어 전체의 씬 그래프를 구성하게 된다. 리소스는 노드의 실제 정보를 갖는다.

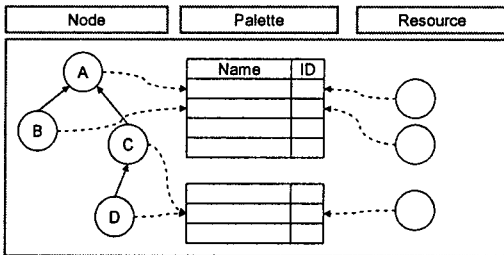


Fig. 4 Scene graph, node, resource and palette of U3D

노드와 리소스의 종류는 다음과 같다. 별도의 리소스를 갖지 않고 변환행렬과 노드 간의 계층정보만을 갖는 그룹 노드, 뷰 정보를 가지는 뷰 노드와 뷰 리소스, 광원 효과에 대한 라이트 노드, 라이트 리소스가 있다.

가시화를 위한 모델 정보를 가지는 모델 노드와 모델 리소스, 모델 리소스의 색상이나 텍스처를 처리하는 material 리소스, shader 리소스, 텍스처 리소스로 구성이 된다. 또한 모델의 애니메이션을 위한 모션 리소스가 있다. 모든 노드는 기본적으로 변환행렬을 가지며, 각각의 리소스와 연결하기 위한 리소스 이름을 가지고 있다.

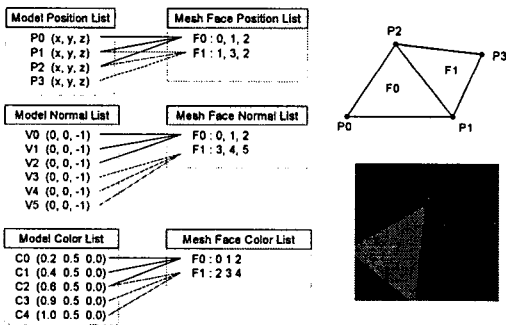


Fig. 5 Face mesh of U3D

형상의 가시화 부분을 정의하는 모델 리소스는 face정의를 위한 메쉬, curve 정의를 위한 라인, 포

인트의 3가지 형태를 지원하고 있다. 메쉬의 구조를 보면 Fig.5와 같이 메쉬를 위한 리스트로 구성되며, 각 포인트 리스트 (model position list)와 각 포인트의 노말 리스트 (model normal list), 셰이드 리스트(shade list) 등을 갖는다, face별로 형상을 위한 3개의 포인트 리스트(mesh face position list), face의 방향을 위한 3개의 노말 포인트 리스트 (mesh face normal list), 각각의 face별로 다른 색상 구현을 위한 shading list를 구성한다. Fig. 5는 face별로 다른 색상을 구현한 사례를 보여주며 CAE 해석결과를 문서화할 때 활용이 가능할 것으로 보인다.

Intel은 U3D포맷을 프로그래밍으로 다룰 수 있는 C++ 라이브러리를 개발하여 3DIF와 SourceFORGE 웹사이트에서 공개하고 있으며, U3D의 기본 프로파일에 대하여 정의하는 Core DL(Dynamic Library), U3D 파일의 출력을 위하여 인코딩을 지원하는 Exporting DL, U3D 파일의 디코딩을 지원하는 Importing DL로 구성이 된다. Fig. 6은 Core DL의 U3D 파일 생성을 위한 컴포넌트 다이어그램을 보여준다.<sup>8</sup>

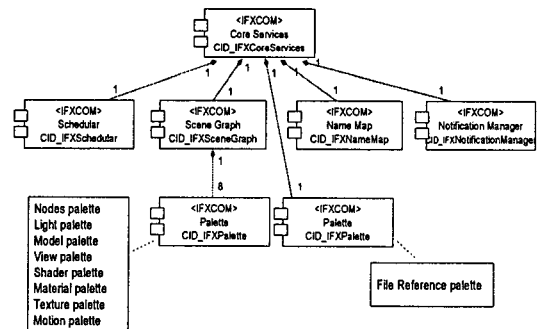


Fig. 6 Core component diagram of U3D

### 3.2.4 IDTF

Intel은 바이너리 형식의 U3D 파일의 생성시 다양한 3D 시스템으로부터 나온 데이터를 원활히 변환 할 수 있도록 텍스트 형식으로 기술할 수 있는 중간 형태로서 IDTF (Intermediate Data Text File) 포맷을 별도로 제시하였으며 이는 U3D 파일과 동일한 구조로 이루어지며 Fig. 7과 같은 형태를 가진다.<sup>8</sup> 이를 이용하여 데이터의 소스 시스템은 직접 U3D 인터페이스를 구현하지 않고 간단한 텍스트 포맷 출력으로 U3D를 위한 데이터를 내보낼 수 있다.

```

FILE_FORMAT "IDTF"
FORMAT_VERSION 100

NODE "VIEW" {
  NODE_NAME "DefaultView"
  PARENT_LIST {
    PARENT_COUNT 1
    PARENT 0 {
      PARENT_NAME "<NULL>"
      PARENT_TM {
        1.000000 0.000000 0.000000 0.000000
        0.000000 0.258819 0.965926 0.000000
        0.000000 -0.965926 0.258819 0.000000
        0.000000 -50.461461 100.704765 1.000000
      }
    }
  }
  RESOURCE_NAME "SceneViewResource"
  VIEW_DATA {
    VIEW_TYPE "PERSPECTIVE"
    VIEW_PROJECTION 34.515877
  }
}

NODE "MODEL" {
  NODE_NAME "FaceNode"
  PARENT_LIST {
    PARENT_COUNT 1
    PARENT 0 {
      PARENT_NAME "<NULL>"
      PARENT_TM {
        1.000000 0.000000 0.000000 0.000000
        0.000000 1.000000 0.000000 0.000000
        0.000000 0.000000 1.000000 0.000000
        -3.336568 -63.002571 0.000000 1.000000
      }
    }
  }
  RESOURCE_NAME "FaceResource"
}
    
```

Fig. 7 An example of IDTF

### 3.2.5 자바스크립트

자바 스크립트는 1995년 넷스케이프사에서 웹 브라우저의 클라이언트 쪽에서 독립적으로 실행되는 프로그램을 작성하기 위하여 만들어진 스크립트 언어이다. 처음 라이브스크립트(LiveScript)라고 하여 발표되었고 후에 자바를 만든 선 마이크로시스템즈와 공동으로 라이브스크립트를 확장시킨 자바스크립트로 변경되었다. 또한 ECMA에서 표준으로 채택하면서 ECMA 스크립트라고도 불린다.

자바스크립트는 웹브라우저의 클라이언트에서 사용자와 대화형으로 동작이 가능한 특징을 가지고 있다. 이러한 특징을 이용하여 PDF내에서 문서와 사용자간의 인터랙션을 위하여 Acrobat 자바스크립트를 제공한다. PDF문서 내에서 Acrobat 자바스크립트를 활용하면 새로운 PDF 문서를 생성하거나 form을 생성하거나 수정하기, 동영상, 음악 등 디지털 미디어 파일의 추가할 수 있고, 코멘트나 문서의 승인 등이 가능하다.<sup>9</sup>

3D PDF는 3D 데이터와 사용자간의 인터랙션을 위해서 Acrobat3D 자바스크립트가 별도로 제공이 되며, U3D에서 정의된 노드, 리소스 등의 정보에 접근하거나 수정할 수 있으며, 애니메이션의 구현, 뷰의 변경, 모델의 이동, 회전 등이 가능하다. Fig. 8은 Acrobat3D 자바스크립트를 이용하여 모델의 회전 사례를 보여준다.<sup>10</sup>

본 연구에서는 Acrobat3D 자바스크립트를 이용

하여 PDF 문서상의 3D 모델의 색상변경, 조립체의 분해 및 조립이 가능하게 하였다.

```

// get index of page containing the Annot3D object (count starts at 0)
pageIndex = this.pageNum;

// index of the Annot3D (count starts at 0)
annotIndex = 0;

// get a reference to the Annot3D script context
c3d = this.getAnnots3D( pageIndex ][ annotIndex ], context3D);

// get a reference to the node in the scene named "Axes"
axes = c3d.scene.nodes.getByNamed( "Axes" );

// rotate the object about the X-Axis PI/6 radians (30 degrees)
axes.transform.rotateAboutXInPlace( Math.PI / 6 );
    
```

Fig. 8 An example of Acrobat3D Javascript

### 3.3 U3D 변환 모듈

DynaView의 3D 형상 정보 및 조립체의 조립 정보, 분해 정보에 대한 인터페이스를 위하여 IDTF를 활용하였으며 변환 대상 및 구조, 노드 및 리소스에 대한 명명 규칙을 별도로 정의하였다. 조립체의 구조 표현을 위하여 어셈블리와 하위 어셈블리는 그룹 노드를 이용하고, 최종 파트에 대해서는 모델 노드를 이용하였다.

각각의 모델 노드에 대해서는 모델의 페이스 표현을 위한 메쉬와 에지 표현을 위한 라인 정보를 동시에 구현하는 구조를 정의하였다. 조립체의 분해도를 위한 각 파트의 위치정보는 Fig. 9와 같이 그룹 노드의 변환 행렬을 이용하여 추가하고, 다른 노드와 구분하기 위하여 노드 명에 특정 접두어를 두어 정의하였다. 분해도를 PDF 문서상에 보여주기 위해서 Acrobat3D자바스크립트를 통하여 구현이 되었다.

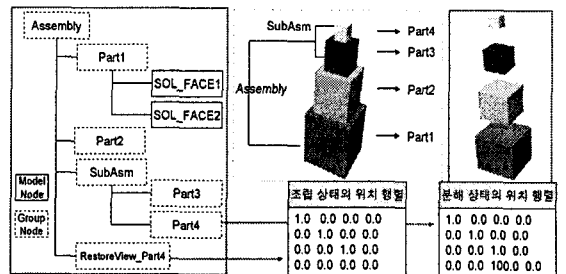


Fig. 9 Node definition for exploded view

Fig. 10과 같이 DynaView에서 조립상태의 모델

을 불러와서 뷰어의 기능으로 분해상태로 만들고 변환을 위한 출력을 하게 되면 Fig. 9과 같은 분해 정보 및 조립정보를 IDTF로 출력한다. 출력된 IDTF 파일은 U3D 변환 모듈을 통하여 U3D파일로 변환 된다.

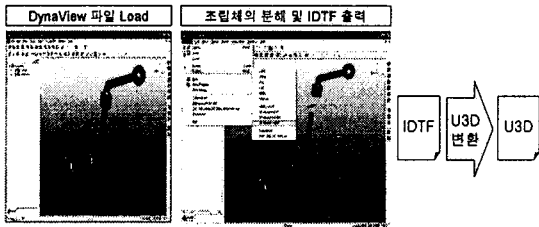


Fig. 10 U3D conversion module

Fig. 11은 U3D 변환을 위하여 프로그래밍에서 수행되는 절차 및 연결 관계를 보여준다. core service, Scene graph, node 및 resource 클래스의 객체를 생성하고 IDTF로 출력된 정보를 각각의 객체에 연결하여 최종적으로 U3D로 인코딩하는 함수를 호출하여 변환하게 된다.

U3D 변환 모듈의 구현은 Microsoft Visual Studio .NET 2003 환경에서 인텔에서 제공한 U3D SDK를 이용하였다.

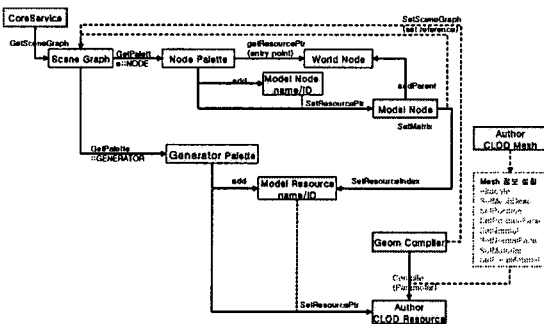


Fig. 11 Programming architecture of U3D conversion

### 3.4 PDF 변환 모듈

정형화된 PDF 문서를 자동화하고 3D에 대한 조작기능을 제공하기 위해서는 템플릿 PDF 문서가 필요하다.

Fig. 12는 템플릿 문서를 보여주며, Acrobat을 이용하여 자동화할 문서의 양식을 미리 정의할 수 있다. 템플릿문서의 구성은 데이터베이스 등 Text

정보를 자동으로 가져올 텍스트 폼 필드, 3D데이터를 조작하기 위한 자바스크립트를 포함하고 있는 버튼, PDF 문서가 처음 열릴 때 수행되는 도큐먼트 자바스크립트로 정의되어 있다.

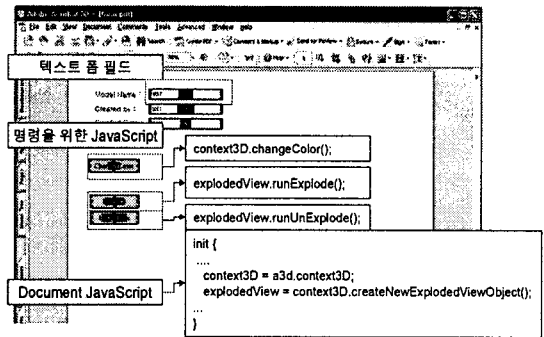


Fig. 12 Template document

텍스트 폼 필드는 약속된 필드 명으로 정의하여야 한다. 버튼과 같이 정의되는 자바 스크립트는 동작되어야 할 명령에 대한 함수 명을 정의하였다. 도큐먼트 자바스크립트는 3D데이터 전체 객체에 대한 변수 명을 정의하고 생성된 문서가 열릴 때 초기에 수행해야 할 스크립트가 정의된다. Fig. 13 에서의 도큐먼트 스크립트는 조립상태 View를 초기에 보여주도록 설정한 예제이다. 실제 동작을 구현한 자바스크립트는 프로그래밍에 의해 U3D와 동시에 정의된다.

```
function init()
{
    if ( waitingFor3DActivated )
    {
        var a3d = getAnnots3D(0)[0];
        if ( a3d.activated )
        {
            waitingFor3DActivated = false;

            context3D = a3d.context3D;
            explodedView =
                context3D.createNewExplodedViewObject();
            explodedView.runUnExplode ();
        }

        if ( timeout.count >= 10 ) // seconds
        {
            app.clearInterval( timeout );
        }
        timeout.count++;
    }
}

timeout = app.setInterval( "init()", 1000 );
timeout.count = 0;
var waitingFor3DActivated = true;

context3D.changeColor();
explodedView.runExplode();
explodedView.runUnExplode();
```

Fig. 13 Document JavaScript

템플릿 문서는 변동 부에 해당하는 3D 데이터와 동작 구현을 위한 자바스크립트, 텍스트 폼 필드에 입력될 문서 내용은 포함하지 않고, 변경될 필요가 없는 고정부만 정의하게 된다.

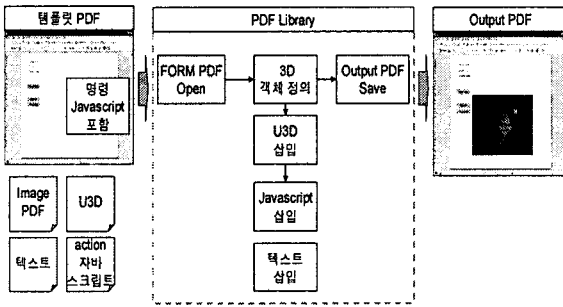


Fig. 14 PDF document generation

PDF 변환 모듈은 프로그램 내부적으로 Fig. 14의 과정으로 수행된다. 먼저 템플릿 PDF 문서를 열어 U3D변환 모듈로부터 생성된 U3D 파일과 조립체의 분해 및 색상변경을 위한 동작이 구현된 자바스크립트를 PDF 문서에 동시에 추가한다. 그리고 문서의 텍스트 폼 필드에 삽입될 문자를 추가한 후 최종 PDF 문서로 저장된다. Fig. 15는 PDF변환을 위한 프로그램의 구성의 주요 부분이다.

PDF 변환 모듈은 Microsoft Visual Studio .NET 2003 환경에서, PDF 라이브러리를 이용하여 구현되었다.

```

char *JsFileName = "action.js";
char *pdImageFilePath = "image.pdf";
char *pdFormFile = "form_eco.pdf";

....중략...

//PDF Template File Open
if ((pdDoc = MyPDDocOpen(pdFormFile)) == NULL) {
    printf ("Sample application error: Unable to open input PDF file\n");
    exit(-1);
}

pdPage = PDDocAcquirePage (pdDoc, 0);

//Text Form Field 내용 삽입
inputForm (pdDoc);

// creat annotation
newAnnot = PDDocAddNewAnnot(pdPage, -2, ASAAtomFromString("3D"),
&annotRect);

// set other entries
PDDocSetFlags(newAnnot, pdAnnotPrint | pdAnnotReadOnly);
//PDDocSetFlags(newAnnot, 68); // value used in adobe sample file

// 3D 및 자바스크립트 삽입
embedStatus = EmbedDataIn3DAnnot(pdPage,
    newAnnot, // 3D annot
    u3dfile, // Path to the U3D file
    JsFileName, // Path to the JavaScript file
    bCreate3DV); // whether to create 3DV dictionary

....중략...

//회편 문서 저장
PDDocSave(pdDoc, PDSaveFull | PDSaveLinearized, ASPathFromPlatformPath (outpd),
ASGetDefaultFileSys(), NULL, NULL);
    
```

Fig. 15 Main function of source

#### 4. 적용 사례

Fig. 16은 자바스크립트를 이용하여 3D 데이터의 색상 변경과 조립체의 분해도를 구현한 사례이다. 좌측의 두 그림은 화면 왼쪽의, 스크립트와 연결된 버튼을 누름으로써 모델의 색상이 바뀌도록 작성된 문서이고, 우측의 두 그림은 화면상의 버튼을 누를 때 동적으로 분해 및 조립의 애니메이션이 되도록 한 문서이다.

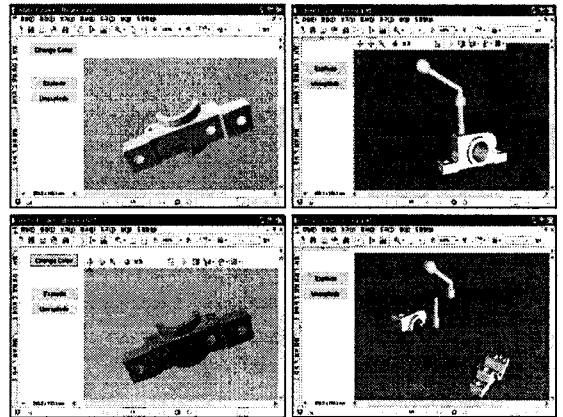


Fig. 16 Change of color(left), Exploded view(right)

Fig. 17은 CATIA의 원본 CAD데이터의 크기가 50MB이상이고, 삼각형 메쉬 수가 각 418,758개, 322,129개로 복잡한 모델에 대해 적용한 사례이다. PDF문서로의 변환은 문제가 없으나 Acrobat Reader를 통하여 3D를 회전, 줌 및 이동 등의 작업 시에 어려움이 있었는데 대용량의 모델에 대해서는 U3D변환 시 메쉬 수 최적화가 필요하다.

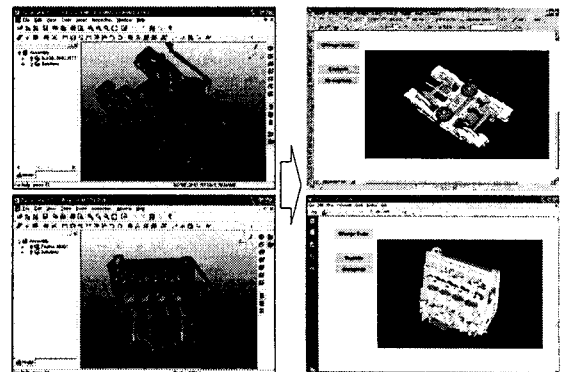


Fig. 17 Large models

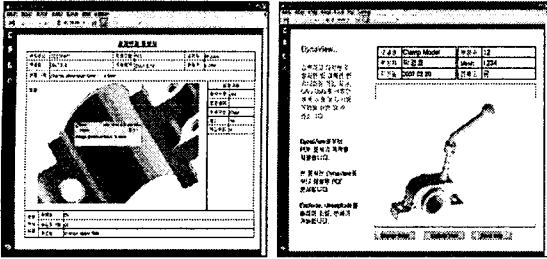


Fig. 18 Formatted documents

Fig. 18은 다양한 양식의 PDF 템플릿을 만들어 용한 사례를 보여준다. Fig. 19는 본 논문의 PDF버전에 실제 3D PDF를 구현한 사례에 대한 시연을 보여준다. 프린트된 버전이 아닌 PDF 문서 버전을 열면 3D데이터의 조작이 가능하다.

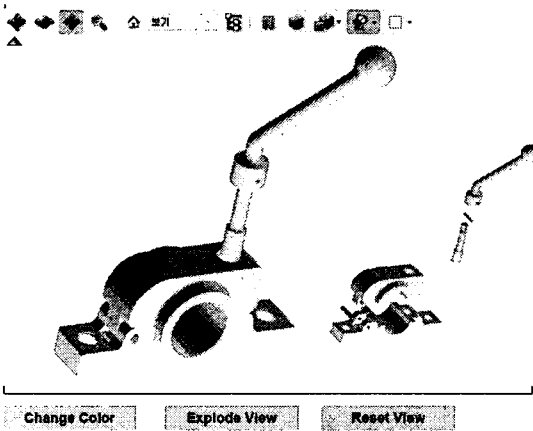


Fig. 19 Demonstration

### 5. 활용 방안 및 결론

본 연구를 통하여 별도의 Acrobat 3D와 같은 3D PDF 저작 도구가 없더라도 정형화된 3D PDF 문서를 자동으로 생성할 수 있는 변환 기술을 개발하였고 상용 뷰어인 DynaView에 적용하였다. 뷰어상의 3D 데이터를 PDF로 출력하게 되면 원하는 형태의 문서로 3D PDF 문서가 자동으로 생성되며, 뷰어에서 분해 상태로 PDF로 출력할 경우 PDF문서에서 분해 및 조립을 사용자가 직접 조작할 수 있는 문서로 생성 되도록 하였다.

본 연구에서 개발된 기술을 확장하면 다양한 3D 시스템상에서 사용자의 요구에 따라 3D 인터랙션이 가능한 여러 형태의 PDF 문서를 자동 생성

하는 것이 가능하다.

또한, Fig. 20과 같이 기업내의 PDM과 연동되면 PDM에서 관리되는 3D 데이터와 BOM (bill of material), 부품 및 제품의 사양 정보, 설계 변경 정보와 연계하여 ECO (engineering change order), 서비스 매뉴얼, 제품 및 부품에 대한 카탈로그 등의 제작에 활용할 수 있다. 기업에서 설계에서 3D를 작성하더라도 설계 변경에 대한 관련부서와의 커뮤니케이션은 2D위주의 문서로 진행하였고, 매뉴얼의 경우 제품의 분해도를 작성하기 위하여 Adobe Illustrator같은 툴을 사용하여 직접 그리는 방식을 활용하고 있다. 이러한 문서들을 3D PDF로 자동 생성한다면 기존의 수작업 작성에 따른 작성 오류 및 시간 손실을 방지할 수 있을 것으로 기대된다.

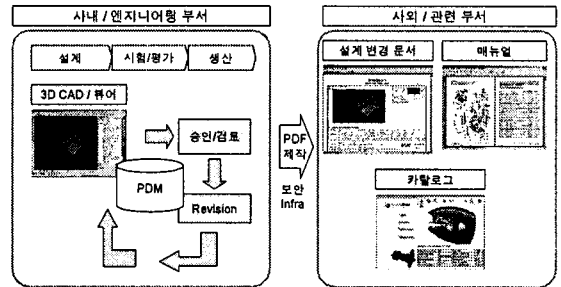


Fig. 20 PDM application

후 기

본 연구는 서울시 산학연 협력사업, 수중운동체특화연구센터 및 한국과학재단 특정기초연구사업(과제번호 R01-2006-000-10327-0)의 지원에 의해서 이루어졌습니다.

### 참고문헌

1. PTC Arbortext, <http://www.ptc.com/>
2. Adobe Systems, <http://www.adobe.com/>
3. INOPS, DynaView, <http://www.inops.co.kr/>
4. Standard ECMA-363 Universal 3D File Format, <http://www.ecma-international.org/publications/standards/Ecma-363.htm>
5. Intel Universal 3D Standard, <http://www.intel.com/technology/systems/u3d/>



6. Adobe – PDF Library SDK,  
<http://www.adobe.com/devnet/pdf/library/>
7. Adobe Acrobat 3D,  
<http://www.adobe.com/products/acrobat3d/>
8. SourceForge.net, <http://sourceforge.net/projects/u3d>
9. JavaScript for Acrobat,  
<http://www.adobe.com/devnet/acrobat/javascript.htm>
10. JavaScript for Acrobat 3D,  
<http://www.adobe.com/devnet/acrobat/javascript3d.html>