

Unity 엔진을 이용한 Aveva Marine 뷰어 구현

장원찬*, 양재균* 김삼성**, 김병석**

*울산대학교

**에이스이엔티

jgyang@ulsan.ac.kr, sskim@aceent.co.kr, bsk0077@aceent.co.kr

Implementation of an Aveva Marine viewer using Unity engine

Jang Won Chan*, Yang Jae Gun*, Kim Sam Sung**, Kim Byung Suk**

*University of Ulsan

**ACEENT

요 약

3차원 CAD를 이용한 선박 설계는 완료 시점에서 부품간의 간섭여부를 판별할 수 있으나 작업순서에 따른 파트의 회전이나 이동 중의 간섭에 대한 판단은 어렵다. 현재는 생산 현장의 작업반장이나 숙련인력이 파트에 대한 회전, 이동시의 간섭 여부를 판단하여 작업순서를 조절한다. 이에 본 논문에서는 Unity 엔진을 이용하여 의장 작업을 실행하기 전에 대상 파트를 손으로 직접 회전, 이동시켜서 실제 작업이 가능한지 여부를 확인할 수 있는 AM 뷰어를 구현하고자 한다.

1. 서론

조선 해양플랜트 생산시물레이션은 디지털 가상시물레이션 기술들로 생산 공정 및 공법 검증, 설비 및 배치의 최적화 및 검증, 생산 관리의 최적화 등의 모사를 실행하는 시스템이다[1]. 3차원 CAD를 이용하여 설계함으로써 의장, 기장, 전장품이 장착완료 시점에서 상호간의 간섭여부를 판별할 수 있으나 작업순서에 따른 파트의 회전이나 이동 중의 간섭에 대한 판단은 설계 시점에서는 불가능하다. 현재는 생산 현장의 작업반장이나 숙련인력이 파트에 대한 회전, 이동시의 간섭여부를 판단하여 작업순서를 조절하거나 기존에 설치한 파트를 분리한 후 재작업을 해야 한다. 즉, 현재의 조선 해양플랜트 의장 작업은 숙련인력의 전문가 지식에 의존하고 있다.

조선소에서 추가적인 비용 및 시수가 발생하는 요소는 파손 3%, 작업미숙 3% 등 선박 건조현장에서 발생하는 것보다 블록의 간섭 50%, 설계정보 오류 30%, 정보누락 10% 등 정보의 오류로 인한 요인이 큰 비중을 차지하고 있다. 이는 설계 수정 과정에서 발생하는 원자재 비용 및 생산 공수, 유희장비, 운송비, 경비 등 경쟁력 저하를 유발한다[2]. 이에 본 논문에서는 의장 작업을 실행하기 전에 대상 파트를 손으로 직접 회전, 이동시켜서 실제 작업이 가능한지 여부를 확인하는 뷰어를 구현하고자 한다.

2. AM 데이터 파일

AM Data 파일 구조

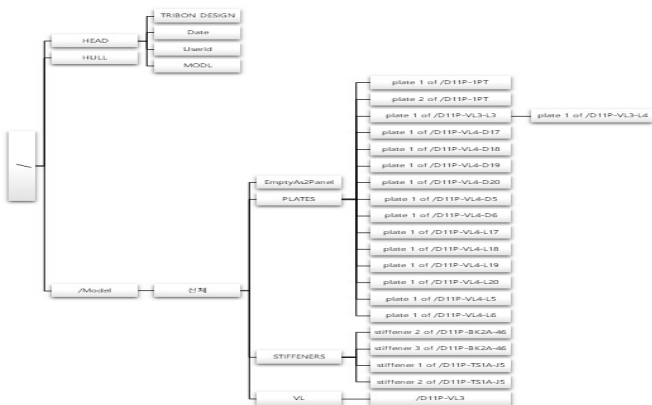
Aveva Marine은 팀 작업을 지원하기 위하여 3차원 형상 정보와 속성 정보를 DBMS를 이용하여 저장하고 관리한다. 한편, 특수한 목적을 위하여 설계 파일을 외부로 내보내기 할 수 있는데 이 경우에 별도로 추출된 파일을 리뷰 파일이라고 한다. 리뷰 파일에는 rvm, rvz, rev 등이 있다. 이 중에서 rvm과 rvz는 바이너리 형식이고 rev는 텍스트 형식이다.

| | | | | |
|----------------------|---------|-------------|-------------|-------------|
| 선체 | 0.00000 | 0.00000 | 0.00000 | |
| 0 | | | | |
| CNTB | | | | |
| 1 | 1 | | | |
| EmptyAs2Panel | | 0.00000 | 0.00000 | |
| 0.00000 | | | | |
| 0 | | | | |
| CNTB | | | | |
| 1 | 1 | | | |
| PLATES | | 0.00000 | 0.00000 | |
| 0.00000 | | | | |
| 0 | | | | |
| CNTB | | | | |
| 1 | 1 | | | |
| plate 1 of /D11P-1PT | | 50150.00000 | 13450.00000 | 14706.00000 |
| 5 | | | | |
| PRIM | | | | |
| 1 | 1 | | | |
| 11 | | 0.00100 | 0.00000 | 0.00000 |
| | | 0.00000 | 0.00100 | 0.00000 |
| | | 0.00000 | 0.00000 | 0.00000 |
| | | 0.00000 | 0.00100 | 0.00000 |
| | | 50050.00000 | 13200.00000 | 14700.00000 |
| | | 50250.00000 | 13700.00000 | 14712.00000 |
| 6 | | | | |
| 4 | | | | |
| 50050.00000 | | 13200.00000 | 14700.00000 | |
| 0.00000 | | 0.00000 | -1.00000 | |
| 50050.00000 | | 13700.00000 | 14700.00000 | |
| 0.00000 | | 0.00000 | -1.00000 | |
| 50250.00000 | | 13700.00000 | 14700.00000 | |
| 0.00000 | | 0.00000 | -1.00000 | |
| 50250.00000 | | 13200.00000 | 14700.00000 | |
| 0.00000 | | 0.00000 | -1.00000 | |

(그림 1) 샘플 rev 파일 일부

AM 속성 Data 구조

Aveva Marine의 3D CAD 모델에 정의된 속성으로는 식별 속성, 스펙(spec) 카탈로그 속성, 기능 속성, 형상 관련 좌표 및 치수 속성, 객체 간의 연결 속성이 있다[3]. 객체 식별 속성으로는 객체의 고유 아이디(id)를 담고 있는 Name 속성, 객체의 고유 데이터베이스 아이디(id)를 담고 있는 REFNO 속성, 및 형상 파일 이름 정보를 담고 있는 SAT 속성이 있다. 스펙 카탈로그 속성으로는 배관의 카탈로그 정보를 담고 있는 PSPE 속성, 피팅에 대한 카탈로그 정보를 담고 있는 SPRE 속성, 절연 처리용 자재 (insulation) 정보를 담고 있는 ISPE 속성이 있다. 기능 속성으로는 설계 온도 및 설계 압력 정보를 담고 있는 TEMP 및 PRES 속성이 있다. 형상 속성으로는 객체의 좌표 정보를 담고 있는 HPOS, TPOS, POS, APOS, LPOS, 및 P3POS 속성, 객체의 직경 정보를 담고 있는 HBOR, TBOR, ABOR, LBOR, 및 P3BO 속성, 객체의 방향 정보를 담고 있는 HDIR, TDIR, 및 P3DIR 속성이 있다. 마지막으로 연결 속성으로는 객체간의 연결 정보를 담고 있는 HREF 및 TREF 속성이 있다.



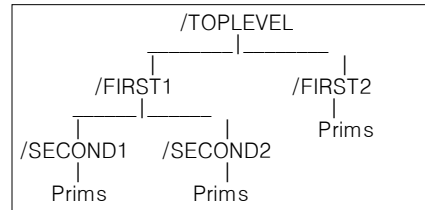
(그림 2) 속성 정보 구조도(일부)

3. AM 데이터 파일 파서

AM binary data file format

리뷰 파일은 여러 레코드 또는 청크로 구성되며 첫 번째는 헤더 청크이다. 청크는 4 개의 문자 식별자, 청크의 끝에 대한 포인터, 주 및 부 개정 번호 및 청크와 관련된 데이터를 포함하는 데이터 세트이다. 청크의 끝에 대한 포인터는 청크를 건너 뛰는데 사용될 수 있으며 개정 번호는 이전 버전과의 호환성을 제공한다. 리뷰 파일에는 다음과 같이 6 가지 유형의 청크가 있다. 헤더 청크(HEAD), 모델 파일 청크(MODL), 컨테이너 시작 청크(CNTB), 컨테이너 엔드 청크(CNTE), 프리미티브 청크(PRIM), 엔드(END) :

모든 리뷰 파일의 처음 두 청크는 HEAD 및 MODL 청크이며 모델 파일에 대한 유용한 정보가 포함되어 있다. 작성된 날짜, 파일 노트, 프로젝트, MDB. 이 두 청크 다음에는 END : chunk로 표시된 파일 끝까지 CNTE, CNTB 및 PRIM 청크가 있다. CNTB 및 CNTE 청크는 '아래'에 해당하는 CNTB와 '위'에 해당하는 CNTE를 갖는 PDMS 데이터베이스 계층 구조를 설명한다. 아래 그림은 계층 구조를 재현하는 방법을 보여준다.



(그림 3) Hierachy

AM Data file parser

다음 과정으로 rvm 파일을 읽고 메모리로 로딩하는 프로그램을 작성하였다. 개발언어는 C#을 사용하였고 리뷰파일의 포맷에 따라서 각 청크에 해당하는 클래스를 선언하였다. 아래는 CNTB 청크에 대응하는 클래스의 일부이다.

```
class CNTB : Chunk
{
    private string ContainerName;
    private float[] Origin = new float[3];
    private int Colour;

    public CNTB()
    {
        ContainerName = "";
        Origin[0] = 0.0F;
        Origin[1] = 0.0F;
        Origin[2] = 0.0F;
        Colour = 0;
    }
}
```

(그림 4) CNTB 청크 클래스

아래의 그림은 청크의 데이터가 정수인 경우에 파일의 특정 위치에서 데이터를 읽어 오는 메소드이다.

```
public unsafe int getIntfromBuff(byte[] buff, ref int pos)
{
    int result = 0;
    fixed (byte* ps = &buff[pos])
    {
        result = *ps * 16777216 + *(ps + 1) * 65535 + *(ps + 2) * 256 + *(ps + 3);
    }
    pos += 4;
    return result;
}
```

(그림 5) 파일에서 정수를 반환하는 메소드

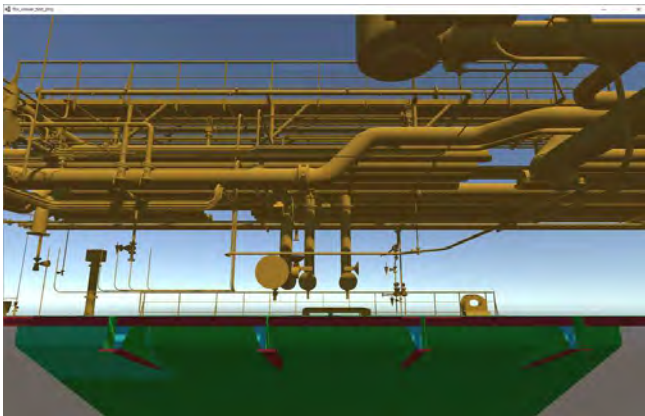
4. Unity 엔진의 동적 Asset 로딩

일반적인 Unity 엔진 사용 방법에서는 프로그램을 개발하는 과정에서 필요한 자원을 Asset으로 등록한 후 빌드하여 배포한다[4]. 3차원 도면 뷰어의 경우에는 다양한 도면을 필요시에 파일형태로 로딩해야 하기 때문에 사전에 Asset으로 등록할 수 없다. 따라서 외부 도면 파일을 프로그램 실행 시점에서 Unity 엔진으로 로딩해야 한다[5]. 아래는 run-time에 외부 파일을 Unity로 로딩하는 메소드의 일부이다.

```
public GameObject LoadFromFileWithTextures(string filename, AssetLoaderOptions options = null, GameObject wrapperGameObject = null, string basePath = null, AssimpInterop.ProgressCallback progressCallback = null)
{
    var fileData = FileUtils.LoadFileData(filename);
    var extension = FileUtils.GetFileExtension(filename);
    if (basePath == null)
    {
        basePath = FileUtils.GetFileDirectory(filename);
    }
    InternalLoadFromMemoryAndZip(fileData, extension, basePath, options, wrapperGameObject != null, null, null, null, progressCallback);
    var loadedGameObject = BuildGameObject(options, extension, wrapperGameObject);
    ReleaseImport();
    return loadedGameObject;
}
```

(그림 6) run-time에 파일을 로딩하는 메소드

아래는 Unity엔진을 이용하여 AM 데이터 파일을 run-time에 읽어서 표현한 최종 결과이다.



(그림 7) 뷰어 최종 결과

참고문헌

- [1] 이필립,황인혁. "조선해양 공정 상호검증 시뮬레이션과 생산관리 플랫폼" 大韓造船學會誌 VOL.48 NO.4 (2011):10-13
- [2] 최영태(Yeong-Tae Choi),서흥원(Heung-Won Suh),이순섭(Soon-Sup Lee). "선체설계와 의장설계 간의 정보인터페이스 기법 연구" 大韓造船學會 論文集 VOL.50 NO.6 (2013):458-465
- [3] 손명조(Myeong-Jo Son),강형우(Hyungwoo Kang),김태완(Tae-wan Kim). "조선 선체 블록 모델의 재사용을 위한 AVEVA Marine Scheme 기반 모델링" 한국CDE학회 논문집 VOL.19 NO.1 (2014):41-49
- [4] 최형욱 (Hyeoung Wook Choi), 강수명 (Soo Myung Kang), 김경준 (Kyung Jun Kim), 김동영 (Dong Young Kim), 정윤재 (Yun Jae Choung). "Unity 3D 엔진을 활용한 강우레이더 자료 시각화 프로토타입 개발" 한국지리정보학회지 VOL.18 NO.4 (2015):131-144
- [5] Unity Manual, <https://docs.unity3d.com/Manual/LoadingResourcesatRuntime.html>

ACKNOWLEDGMENT

이 성과는 2019년도 중소벤처기업부의 재원으로 한국산학 연합회 연구마을지원사업의 지원을 받아 수행된 연구임