

# 디지털 트윈 확장성을 위한 계층적 USD 구조: Dream-AI+ Twin 적용 사례와 검증

김강렬, 김종원\*

광주과학기술원 AI 융합학과

kr\_kim99@gm.gist.ac.kr, \*jongwon@gist.ac.kr

## A Hierarchical USD Structure for Digital Twin Scalability: Case Study and Validation with Dream-AI+ Twin

KangRyeol Kim, JongWon Kim\*

Department of AI Convergence, Gwangju Institute of Science and Technology (GIST)

### 요 약

디지털 트윈 구축 시 객체 수 증가는 USD 파일 관리 복잡도와 초기 렌더링 시간 증가로 이어진다. 본 논문에서는 이러한 문제를 해결하기 위해 공간 단위 기반의 계층적 USD 구조를 제안한다. 제안 구조는 Payload 를 활용한 공간별 지연 로딩, Sublayer 기반 카테고리 구성을 통한 개별 자산 참조의 2 단계 계층으로 설계되어 체계적인 파일 관리와 Stage 조직화를 가능하게 한다. 건물 디지털 트윈인 Dream-AI+ Twin 에 제안 구조를 적용한 결과, 캐시 사용 시 초기 렌더링 시간이 24.15 초에서 4.92 초로 79.6% 단축되었으며, 캐시 미사용 시에도 13.5%의 성능 향상을 확인하였다. 이는 제안 구조가 파일 관리 효율성뿐만 아니라 초기 렌더링 성능 개선에도 기여함을 보여준다.

### I. 서 론

디지털 트윈은 실제 세계의 가상 표현과 이들 간의 정보 교환으로 구성되며, 다양한 분야에 적용되어 더 나은 현실 세계를 구현하는 데 기여할 수 있다.[1] 디지털 트윈을 구축하고 활용하기 위해서는 현실 세계를 가상으로 표현하는 목적에 적합한 수준을 맞춘 미리링 과정이 필수적이다.[2] 이러한 미리링을 지원하는 NVIDIA Omniverse 는 Universal Scene Description(USD)를 기반으로 디지털 트윈 구축을 위한 API, SDK 및 서비스를 제공하는 플랫폼이다.[3] 여기서 USD 는 3D 세계의 설계, 구성, 시뮬레이션 및 협업을 지원하는 확장 가능한 개방형 생태계로, 가상 세계 구축을 지원하는 역할을 수행한다.[4]

디지털 트윈 구축에서 USD 는 자산의 통합 표현을 담당하기 때문에, 대상 객체 수가 증가할수록 장면의 규모와 파일 관리 복잡도가 함께 증가한다. USD 기반 시스템에서 객체 수 증가는 관리해야 할 파일 수의 증가로 직결된다. 특히 다수의 협업자가 참여하는 대규모 디지털 트윈 구축 과정에서는 체계적이지 못한 파일 저장 구조와 비직관적인 Stage 구성으로 인해 자산 탐색의 어려움, 초기 렌더링 시간 증가 등의 문제가 발생할 수 있다. NVIDIA 의 Residential Lobby 사례와 같이 USD composition 구조 설계 사례가 존재하지만,[5] 주로 협업 워크플로우에 초점을 맞추고 있어 객체 수 증가에 따른 체계적인 파일 저장 구조 및 Stage 조직화 방안에 대한 구체적인 가이드라인은 제한적이다.

본 논문에서는 건물 디지털 트윈인 Dream-AI+ Twin 을 대상으로, 공간 단위 기반의 계층적 USD 구조를 제안한다. 제안하는 구조는 Payload 와 Sublayer 를 활용한 2 단계 계층 설계를 통해 파일의 체계적 저장과 Stage 조직화를 수행하며, 정적/동적 요소의 분리를 통해 확장성과 초기 렌더링 성능을 개선한다.



그림 1. Dream-AI+ Twin 디지털 트윈 구현체

### II. Dream-AI+ Twin 디지털 트윈의 구조적 문제점

Dream-AI+ Twin 디지털 트윈 구현체는 L40S, RTX A6000, A100 등의 GPU 로 구성된 Omniverse Enterprise 기반 TwinX 클러스터를 활용하여 구현되었다. 그림 1 은 Omniverse 기반의 Dream-AI+ Twin 디지털 트윈 구현체를 보여준다. 해당 구현체는 AI\_Grad\_Building, AI\_Grad\_Datacenter, AI\_Grad\_Extension 의 세 가지 주요 공간으로 구성된다. AI\_Grad\_Building 은 실험실과 사무실로 구성된 건물이며, AI\_Grad\_Datacenter 는 서버 랙과 냉각 통로를 포함하는 데이터센터, AI\_Grad\_Extension 은 로봇 실험실과 소규모 데이터센터를 갖춘 확장 건물이다.

초기 Dream-AI+ Twin 구축 과정에서 다수의 협업자가 참여함에 따라 다음과 같은 구조적 문제가 발생하였다. 첫째, 다양한 자산을 표현하는 USD 파일들이 정돈되지 않은 폴더 구조에 저장되어 특정 자산 탐색에 어려움이 있었다. 둘째, Stage 내 모든 Prim 이 단일 계층에 나열되어 원하는 객체를 찾거나 수정하기 위해 수십 개의 Prim 을 일일이 확인해야 하는

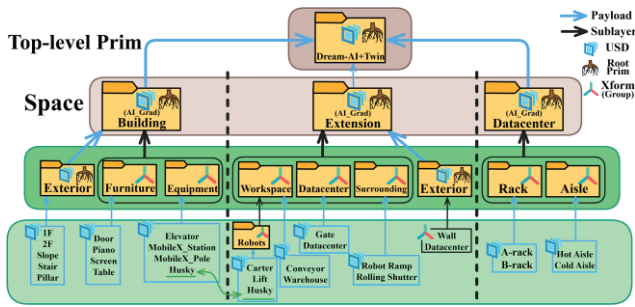


그림 2. Dream-AI+Twin의 계층적 USD 구조

비효율이 존재하였다. 셋째, 동일한 USD 파일을 여러 공간에서 재사용할 때 각 공간별로 변형된 버전이 필요한 경우 의존성 관리가 복잡해지는 문제가 있었다. 이러한 비구조화된 상태는 규모가 확장될수록 파일 관리 복잡성과 초기 렌더링 시간이 증가하는 문제로 이어졌다.

### III. 공간 단위 기반 계층적 USD 구조 설계

본 논문에서는 II장에서 제시한 문제를 해결하기 위해 공간 단위 기반의 계층적 USD 구조를 제안하고, 이를 Dream-AI+Twin에 적용하여 Omniverse Isaac Sim 환경에서 구현하였다. 제안하는 구조에서 각 공간(Space)은 용도(실험실, 사무실, 데이터센터 등) 또는 물리적 범위(건물 단위)를 기준으로 정의되며, 외관(Exterior), 가구(Furniture), 장비(Equipment)와 같은 카테고리별 세부 요소들을 참조하여 구성된다. 이를 통해 정적 요소(건물 구조, 고정 가구 등)와 동적 요소(로봇 등)를 명확히 분리하여 관리할 수 있다.

그림 2는 제안하는 계층적 USD 구조를 나타낸다. 최상위 Root Prim인 Dream-AI+Twin은 AI\_Grad\_Building, AI\_Grad\_Extension, AI\_Grad\_Datacenter의 세 가지 공간 단위를 Payload 방식으로 참조한다. Payload는 필요 시에만 로드하는 지연 로딩을 통해 초기 렌더링 시간을 최적화한다. 각 공간은 다시 하위 카테고리 USD 파일들을 Sublayer 방식으로 구성한다. Sublayer는 여러 USD 레이어를 하나의 Stage로 결합하는 방식으로, 카테고리별 자산을 논리적으로 그룹화하는 데 사용된다. 최하위 계층에서는 개별 자산 USD 파일들이 Payload 방식으로 참조되며, 계층 구조 내에서 Xform 타입의 Prim은 변환(이동, 회전 등) 정보를 담는 컨테이너로 사용되어 여러 객체를 그룹화하는 역할을 한다.

이러한 공간 단위 기반 계층 구조는 다음과 같은 이점을 제공한다. 첫째, 사용자는 Payload의 로드/언로드 기능을 활용하여 관심 있는 특정 공간만 선택적으로 로드할 수 있어 Stage 탐색 복잡도가 감소한다. 둘째, 새로운 공간 추가 시 독립적인 USD 파일을 생성한 뒤 최상위 USD에서 Payload 방식으로 참조만 추가하면 되므로, 기존 구조를 수정하지 않고도 확장이 가능하다. 셋째, 공간별 독립적인 폴더와 USD 파일 관리를 통해 버전 관리가 용이하며, 여러 협업자가 서로 다른 공간을 동시에 작업할 때 발생할 수 있는 충돌을 방지할 수 있다.

또한, 하나의 자산을 표현하는 USD 파일이 두 개 이상의 공간에서 사용될 때 각 공간에서 변형된 버전이 필요한 경우, 본 연구에서는 USD 파일을 복제하여 각 공간 폴더에 별도로 저장하는 방식을 채택하였다. 이는 공간별 독립적인 수정을 가능하게 하며, 한 공간의 자산 변경이 다른 공간에 의도치 않게 영향을 미치는 것을 방지한다.

	구조 적용 전(s)	구조 적용 후(s)
캐시 사용	24.15	4.92
캐시 미사용	222.86	192.67

표 1. 계층적 USD 구조 적용 전후 초기 렌더링 시간 비교

### IV. 제안 구조 기반의 Dream-AI+Twin 성능 검증

제안한 계층적 USD 구조의 효과를 검증하기 위해 구조 적용 전후 Dream-AI+Twin.usd의 초기 렌더링 시간을 측정하였다. 실험은 Omniverse Isaac Sim 환경에서 RTX 4070 GPU를 사용하여 캐시 사용 여부에 따라 각각 10회 반복 측정하여 평균값을 산출하였다.

표 1에서 볼 수 있듯이 캐시 사용 시 초기 렌더링 시간이 24.15초에서 4.92초로 79.6% 단축되었으며, 캐시 미사용 시에도 222.86초에서 192.67초로 13.5% 단축되었다. 이는 Xform 타입의 Prim이 90개 이상 추가되었음에도 불구하고 달성된 결과이다. Payload를 통한 지연 로딩과 공간 단위 기반 계층 구조가 초기 Stage 로드 시 불필요한 자산의 로딩을 최소화하여 초기 렌더링 시간을 효과적으로 감소시킬 수 있음을 보여준다.

### V. 결론

본 논문에서는 대규모 디지털 트윈 구축 시 발생하는 USD 파일 관리 복잡도와 초기 렌더링 시간 증가 문제를 해결하기 위해 공간 단위 기반의 계층적 USD 구조를 제안하였다. 제안 구조는 Payload와 Sublayer를 활용하여 파일의 체계적 저장과 Stage 조직화를 달성하는 2단계 계층 설계이다. Dream-AI+Twin 디지털 트윈에 제안 구조를 적용한 결과, 캐시 사용 시 초기 렌더링 시간이 79.6%, 캐시 미사용 시 13.5% 단축되어 파일 관리 효율성과 렌더링 성능을 개선하였다.

### ACKNOWLEDGMENT

이 연구는 2025년도 산업통상부와 한국산업기술기획평가원(KEIT) 연구비 지원에 의한 연구(RS-2025-25448249)와 2019년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임(No. 2019-0-01842, 인공지능대학원지원(광주과학기술원))

### 참고 문헌

- [1] Dihan, Md Shezad, et al. "Digital twin: Data exploration, architecture, implementation and future." Heliyon 10.5 (2024).
- [2] Deukyong, Jeong. "Korea Digital Twin Technology Roadmap." IITP (Institute for Information & Communication Technology Planning & Evaluation), Feb. 16, 2022, (<https://www.iitp.kr/kr/1/knowledge/openReference.it>).
- [3] Ahmed, Naveed, et al. "A systemic survey of the Omniverse platform and its applications in data generation, simulation and metaverse." Frontiers in Computer Science 6 (2024): 1423129.
- [4] Universal Scene Description. "Introduction to USD." OpenUSD, (<https://openusd.org/release/intro.html>).
- [5] NVIDIA Corporation, "Residential Lobby.", 2025, ([https://docs.omniverse.nvidia.com/usd/latest/usd\\_content\\_samples/res\\_lobby.html](https://docs.omniverse.nvidia.com/usd/latest/usd_content_samples/res_lobby.html)).