

마인크래프트에서 양방향 디지털 트윈의 구현을 위한 IoT 데이터 동기화 시스템

양병헌*, 정현준**¹, 이석훈**²

IoT Data Synchronization System for Implementation of Bi-directional Digital Twin in Minecraft

Byeongheon Yang*, Hyunjun Jung**¹, and Sukhoon Lee**²

이 논문은 정부(과학기술정보통신부)의 재원으로 한국 연구재단의 지원을 받아 수행된 연구임(No. NRF-2022R1G1A1008493), 본 연구는 환경부/한국환경산업기술원 지중환경오염위해관리기술개발사업 (2022002450002)으로 수행되고 있습니다

요 약

일반적으로 디지털 트윈을 구현하려면 가상 세계 구축에 필요한 물리적 모델에서부터 현실 세계 간 상호작용까지 모든 것을 개발해야 한다. 만약 메타버스 기반으로 디지털 트윈을 구현한다면 메타버스에서 구현된 가상 세계를 사용함으로써 개발 부담이 줄어든다. 마인크래프트는 대표적인 상용 메타버스 게임 중 하나로 캐릭터가 다양한 동작을 통해 세계를 구축해갈 수 있으며 모드 혹은 플러그인을 통해 기능을 추가하는 것이 가능하다. 따라서 본 논문은 마인크래프트에서 양방향 디지털 트윈의 구현을 위하여 IoT 데이터 동기화 시스템을 제안한다. 이를 위하여 본 논문에서는 IoT 플랫폼과 마인크래프트 간의 사물 및 오브젝트의 자료구조 정의, 동기화 알고리즘 설계, 액추에이팅 시퀀스 다이어그램 설계를 기술한다. 또한, 제안 시스템의 검증을 위하여 세 가지 시나리오를 통하여 마인크래프트 내에 스마트오피스를 구현하고 각 시나리오가 잘 동작함을 확인한다.

Abstract

In general, implementing a digital twin requires developing everything from the physical model required to build a virtual world to the interaction between the real world. If digital twin is implemented based on metaverse, the development burden is reduced by using the virtual world implemented on metaverse. Minecraft is one of the representative commercial metaverse games, where characters can build the world through various movements and add functions through mode or plug-in. Therefore, this paper proposes the IoT data synchronization system for the implementation of a bi-directional digital twin in Minecraft. This paper describes defining the data structure of objects and objects between IoT platform and Minecraft, designing a synchronization algorithm, and designing an actuating sequence diagram. For the verification of the proposed system, this paper implements a smart office within Minecraft through three scenarios and confirm that each scenario works well.

Keywords

metaverse, digital twin, IoT, minecraft

* 군산대학교 소프트웨어학과 학사
- ORCID: <https://orcid.org/0000-0002-0788-721X>
** 군산대학교 소프트웨어학부 교수(교신저자)
- ORCID¹: <https://orcid.org/0000-0002-6717-1395>
- ORCID²: <https://orcid.org/0000-0002-3390-5602>

· Received: Dec. 13, 2022, Revised: Jan. 07, 2023, Accepted: Jan. 10, 2023
· Corresponding authors: Sukhoon Lee and Hyunjun Jung
School of Computer Science and Engineering, Kunsan National University,
Korea
Tel.: +82-63-469-8914, Email: {leha82, junghj85}@kunsan.ac.kr

1. 서 론

디지털 트윈은 “컴퓨터에 현실 속 사물의 쌍둥이를 만들고, 현실에서 발생할 수 있는 컴퓨터로 시뮬레이션함으로써 결과를 예측하는 기술”이다[1]. 디지털 트윈은 다양한 산업과 분야에서 현실 세계의 문제를 시뮬레이션하여 예측 분석 최적화가 목적으로 인기 키워드가 되었다[2].

R. Saracco 교수는 디지털 트윈 기술 발전을 5단계로 제시하며 디지털 트윈에 대한 다양한 개념을 제시하였다[3]. 각 단계는 물리적 개체 모델, 현실 세계의 미러링, 물리적 개체와 디지털 트윈 간 양방향 상호작용, 단일 디지털 트윈에서 최소한의 지능과 로컬 데이터 분석, 자율 에이전트로 이루어져 있으며 향후 디지털 트윈이 가야 할 로드맵을 제시한다. 이와 유사하게 [4]에서는 3단계인 양방향 동기화가 가능한 디지털 트윈을 양방향 디지털 트윈, 최종 단계인 5단계에 도달한 물리 및 가상 환경을 통합적으로 인지, 분석, 상황 인지를 자율적으로 동작하는 디지털 트윈을 자율 트윈이라는 용어로 정의하고 있다.

이러한 디지털 트윈은 가상 세계의 구현, 가상 세계와 물리적 세계의 인터페이스, 가상 세계에서의 데이터 분석, 시뮬레이션 및 예측 등으로 구성된다. 가상 세계의 구현에는 가상 세계에 구축에 필요한 현실 세계 모델링, 객체들 사이의 관계 정의 등의 기술이 요구되며, 가상 세계와 물리적 세계의 인터페이스에는 가상 세계와 물리적 세계 간 상호작용을 위한 것으로, IoT와 가상 현실, 증강현실 등과 같은 기술들이 필요하다. 마지막으로 가상 세계에서 데이터 분석, 시뮬레이션 및 예측은 다양한 문제 해결을 위한 모델링, 데이터 분석, 처리 및 예측 등의 기술이 사용된다[5].

한편, 메타버스는 “현실의 나를 대리하는 아바타를 통해 일상 활동과 경제생활을 영위하는 3D 기반의 가상 세계”로 정의된다[6]. 이는 어떤 대상을 모사하여 현실 문제를 해결하려는 디지털 트윈과 달리, 현실을 닮은 또 다른 세계를 창조하는 것을 목적으로 한다[7].

상용화된 메타버스는 발현과 접속 목적에 따라

게임, 소셜, 생활·산업 기반 메타버스와 같은 3가지 유형으로 분류된다. 그중 게임 기반 메타버스로는 마인크래프트, 로블록스, 포트나이트 등이 대표적으로 있다.

디지털 트윈과 메타버스는 가상 세계라는 공통점을 가지고 있으며, 이러한 공통점을 기반으로 서로 융합하기 위한 다양한 연구들이 제시되었다[4][7]. 융합을 제시한 연구들이 있었다. [7]에서는 데이터와 지능을 통합하기 위해서는 디지털 트윈을 활용하며 디지털 트윈으로 효과적인 전달 체계를 위해 메타버스를 내세워 융합을 제시한다. [4]에서는 가상 세계인 메타버스에서 다양한 디지털 트윈들이 자율적으로 상호 동작을 수행하는 초현실적인 개념으로 ‘메타버스 자율트윈’을 제시한다.

이러한 메타버스 자율트윈의 실현을 위해서는 물리적 세계를 디지털화할 수 있는 사물인터넷 플랫폼이 필수적이며, 각 사물인터넷(IoT) 플랫폼과 가상 세계를 연결하기 위해서는 데이터 및 통신 프로토콜 등의 상호운용성 보장이 반드시 요구된다.

한편, IoT 플랫폼 등에서 상호운용성을 보장하거나 향상하기 위한 다양한 기술들이 개발되었다 [8]-[11]. [8]은 IoT에서 객체의 메타데이터 정보를 관리하기 위해서 디바이스 레지스트리를 활용하였다. [9]는 제조 클라우드에서 이기종 간 상호운용성을 위해 oneM2M 표준을 맞추어 디지털 트윈을 구현하였다. [10]은 가상 세계에서의 충돌 실험 시나리오를 위하여 게임엔진을 사용하여 디지털 트윈 프로토타입을 개발하였으며, [11]은 이기종 IoT 플랫폼 간 상호운용성을 위하여 oneM2M 아키텍처 기반의 IoT 플랫폼을 개발하였다.

또한, 메타버스를 기반으로 IoT 플랫폼의 데이터 동기화하는 연구들 역시 진행되었다[12]-[14]. [12]는 메타버스를 구현하여 IoT 플랫폼을 연결하여 데이터 동기화를 하는 방법을 제시하였으며, [13]은 마인크래프트에서 IoT 플랫폼을 연결하여 출석시스템을 개발하였다. [14]는 IoT 플랫폼과 마인크래프트에서 연동을 위한 API를 넣은 시스템 및 구현하였다.

하지만, 이러한 메타버스를 기반으로 한 연구들은 양방향 디지털 트윈을 위한 액추에이팅, 현실 사물과 메타버스 사물 간 연결성 및 상호운용성 등을

고려하지 않는다는 문제를 지닌다.

양방향 디지털 트윈의 구현을 위해서는 일반적으로 다양한 엔진이나 툴을 이용하여 가상 세계를 직접 구축한다. 하지만 메타버스는 게임 등과 같이 이미 만들어진 가상 세계를 재구축하여 현실 세계와 동기화시킬 수 있으므로, 그래픽 구현, 동작 설정, 인터페이스 개발 등의 부담은 줄어든다.

메타버스를 구현하는 다양한 상업용 게임 중 하나인 마인크래프트는 PC에서 혼자 게임을 하거나 개인이 운영하는 비공식 서버에 접속할 수 있다. 개인이 직접 서버를 개설하여 운영할 수 있도록 오픈 소스로서 코드가 제공되기도 한다. 또한, 마인크래프트는 독창적인 게임의 커스터마이징을 위해 디자인이나 아이템을 추가할 수 있도록 모드(Mod) 혹은 플러그인(Plugin) 방식으로 개발이 가능하다.

따라서, 마인크래프트를 이용한 다양한 시스템 및 연구들이 진행되었다[15][16]. [15]는 소프트웨어 교육 교수학습을 위하여 마인크래프트 모드를 활용하였다. 또한, [16]은 전력 계통을 직접 체험을 위하여 배포 중인 마인크래프트 모드인 'IndustrialCraft'를 이용하여 마인크래프트의 가상 세계에서 가상 전력망을 구축하였다. 이렇듯, 마인크래프트는 간단한 모드 혹은 플러그인 개발 등을 통하여 다양한 모습이 표현될 수 있다.

따라서, 본 논문은 마인크래프트에서 양방향 디지털 트윈의 구현을 위한 IoT 데이터 동기화 시스템을 제안한다. 먼저 IoT 플랫폼과 마인크래프트의 동기화 시스템 구현을 위해 사물 정의 및 자료구조, 동기화 알고리즘, 액추에이팅 시퀀스 다이어그램을 설계하고 마인크래프트의 플러그인을 개발한다. 또한, 상호운용성의 향상을 위해 oneM2M의 SDT(Smart Device Template) 표준을 제안 IoT 플랫폼에 적용하여 메타데이터를 등록하는 규칙을 정의한다. 이후, 시나리오를 제시하여 적용되는지 확인한다.

본 논문의 구성은 다음과 같다. 제2장에서는 관련 연구를 기술하고, 제3장에서는 제안 시스템을 기술한다. 제4장에서는 제안 시스템을 구현한 결과를 보이고 시나리오를 기술한다. 마지막으로, 제5장에서는 결론 및 향후 연구에 관해 기술한다.

II. 관련 연구

이 장에서는 메타버스와 IoT 플랫폼의 동기화를 위하여 IoT 상호운용성과 메타버스 내 IoT 데이터의 동기화에 관련된 연구를 기술한다.

2.1 IoT 상호운용성 관련 연구

[7]은 이기종 서버 시스템을 이용하는 제조 클라우드 CPS(Cyber-Physical System)의 연동, 데이터교환, 시스템 통합 등의 어려움을 해소하고자 oneM2M 표준을 적용을 시킨 디지털 트윈을 개발하였다. CPS 시스템 요구사항 바탕으로 oneM2M의 요구사항을 매핑한 후 oneM2M 표준을 따르는 모비우스와 상호운용성 테스트를 통해 요구사항 지원성 검증을 하였다.

[8]은 IoT 표준 중 하나인 oneM2M를 따르는 플랫폼 기반으로 프로토타입 디지털 트윈을 구현하였다. 상호운용성을 고려하여 IoT 플랫폼으로 모비우스, 시뮬레이션을 위한 물리 엔진을 대신할 게임 엔진을 사용하여 항만 대형 크레인 충돌 시나리오를 하였다.

[9]는 기존 IoT 플랫폼들의 문제점인 타 플랫폼 간 상호운용성을 위해 oneM2M 아키텍처 기반의 연동 시스템을 제안했다. OCEAN(Open Alliance for IoT Standard)에서 제공하는 IoT 플랫폼 기반으로 oneM2M 아키텍처를 적용한 IoT 플랫폼을 OCEAN 리소스 모니터링 프로그램과 oneM2M 테스트 케이스를 적용을 시켜 표준이 잘 적용이 되었는지 확인하였다.

이러한 연구들은 IoT 플랫폼에서 상호운용성을 고려하기 위하여 oneM2M 표준을 적용하고 있다.

2.2 메타버스에서 IoT 데이터 동기화 관련 연구

[10]은 unity 엔진을 사용하여 IoT 플랫폼과 연결하고 센서 데이터의 표현 및 사물 제어를 연구하였다. 하지만 이 시스템은 상호운용성에 대한 부분이 고려되지 않아 새로운 시스템이나 디바이스를 개발하려 할 때 큰 비용이 요구된다.

[11]은 메타버스는 마인크래프트를 이용하여 출력 체크를 할 수 있는 디지털 트윈을 구현하였다. 이 시스템은 센서 데이터를 이용해 수업 참여자 참석 여부를 메타버스에서 확인할 수 있다. 하지만 이 시스템은 메타버스 내에서 현실 세계 액추에이터를 연계하지 못하며 현실과 메타버스 사물 간 양방향성이 부족하다.

[12]는 모비우스를 사용하여, 마인크래프트 내에서 환경 센서 데이터를 표현할 수 있도록 구현하였다. 현실 사물 객체화 및 연산 수행을 위한 가상 세계 소프트웨어 제시를 통해 IoT 플랫폼과 마인크래프트 간 인터페이스를 제시하였으나, 액추에이터에 대한 부분을 연동시키지는 못하였다.

이처럼 기존의 연구들은 상호운용성 및 현실 사물과 메타버스 내 사물의 동기화에 대해 구현하였으나, 액추에이팅의 연동까지는 고려하지 않는다.

따라서 본 논문은 양방향 디지털 트윈의 구현을 위하여 상호운용성을 보장할 수 있는 oneM2M 표준을 적용하고, 현실 사물과 메타버스 사물 간 데이터를 모니터링하고 액추에이팅 할 수 있도록 데이터를 동기화하는 방안을 제시한다.

III. 제안 시스템

이 장에서는 제안 시스템에 대한 구조를 설명한

다. 먼저, 제안 시스템의 전체를 설명하고, 마인크래프트와 IoT 플랫폼을 연결하는 모듈에 대해 설명한다. 그리고 모듈에 필요한 모델과 동기화 방법 등을 보여주고, 현실 사물과 마인크래프트 사물을 연결할 자료구조를 제시한다. 또한 상호운용성을 위하여 oneM2M의 SDT를 IoT 플랫폼에 적용하기 위한 규칙을 제시한다.

3.1 제안 시스템 구조

그림 1은 제안 시스템의 구조를 보인다. 빨간 선은 센서 데이터의 흐름을, 파란 선은 액추에이터의 제어 데이터 흐름을 표현한다. 제안 시스템 구조는 왼쪽의 IoT 디바이스와 서버로 구성되는 IoT 플랫폼 영역 그리고 오른쪽의 마인크래프트 서버와 클라이언트로 구성되는 마인크래프트 영역으로 나뉜다. IoT 플랫폼은 현실 세계의 디바이스 스펙 정보와 수집된 데이터를 저장한다. 마인크래프트는 API 모듈을 통해서 마인크래프트와 IoT 플랫폼 간의 인터페이스 및 정보를 관리한다. 플러그인은 현실 사물과 마인크래프트 사물 간 연결성을 위한 사물 관리를 위한 등록 관리자, 데이터 동기화를 위한 동기화 관리자, 액추에이팅을 위한 액추에이터 센터를 통해 IoT와 마인크래프트 간의 인터페이스로서 역할을 하게 된다.

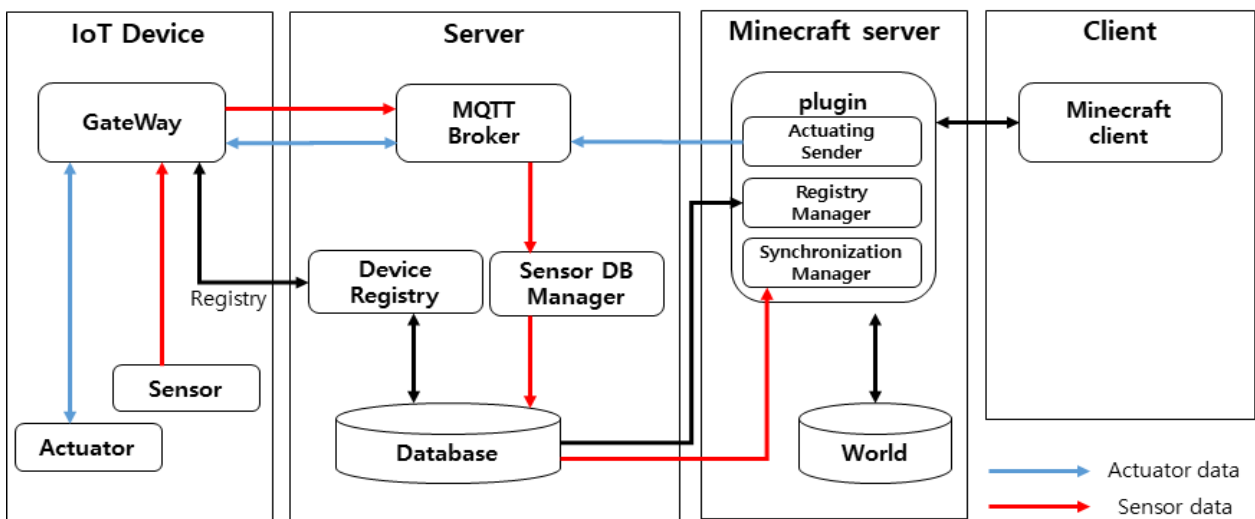


그림 1. 제안 시스템의 구조
Fig. 1. Architecture of proposed system

3.2 데이터 동기화를 위한 관리 시스템

데이터 동기화를 위해서는 마인크래프트의 사물에 현실 세계에서 어떤 사물과 연관되는지 알기 위해서 사물 관리 시스템이 필요하다.

Virtual Object는 현실 세계 사물의 메타데이터와 마인크래프트 사물 정보를 담겨있는 정보 객체이다. Sensor Object 마인크래프트의 사물과 물리 센서의 정보를 가지고 있는 추상 객체이다. Actuator Object 마인크래프트의 사물과 물리 액추에이터의 정보를 가지고 있는 추상 객체이다. Sensor Object와 Actuator Object는 상속받는 구현 클래스에서 현실 사물의 데이터를 가지고 마인크래프트에서 표현하는 방법을 가지고 있다. Actuator Object에는 추가로 액추에이팅에 대한 정보가 포함되어 있다. 그림 2는 Virtual Object의 클래스 다이어그램과 그 예시를 보인다. Virtual Object는 Sensor Object와 Actuator Object를 인터페이스로 가지고, Sensor Object는 가지고 문감지 객체인 DoorObject, 단순 모니터링을 위한 SensorFanObject를 상속하며, Actuator Object는 액추에이팅 기능이 들어간 FanObject, LED 제어가 가능한 LEDObject를 상속해 준다.

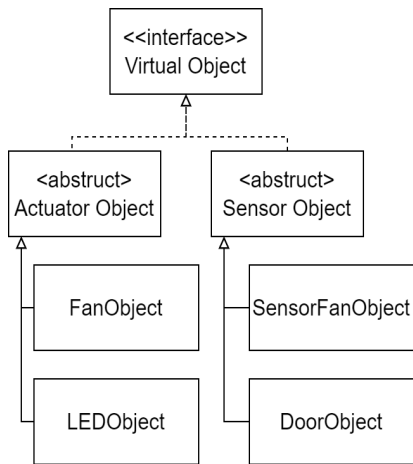


그림 2. Virtual object 클래스 다이어그램
Fig. 2. Class diagram of virtual object

그림 3은 IoT 플랫폼에 현실 사물이 등록되어 있다는 전제하에서 마인크래프트 내의 사물에 현실 사물과의 연관성을 부여하기 위하여 등록하는 과정을 나타낸다.

등록 과정은 크게 6단계로 플레이어가 사물을 요청하는 것으로 시작이 된다. API가 IoT 플랫폼에서 등록된 사물들 정보를 받아와서 마인크래프트 내 UI에 맞게 플레이어에게 보여준다. 플레이어가 원하는 사물을 선택하면 Virtual Object를 생성하고 플레이어에게 설치할 수 있도록 권한을 준다. 플레이어가 원하는 장소에 해당 객체를 설치하면 해당 사물을 동기화 시스템에 등록 및 초기화를 하여 최종적으로 플레이어에게 해당 사물이 보이도록 설치가 된다.

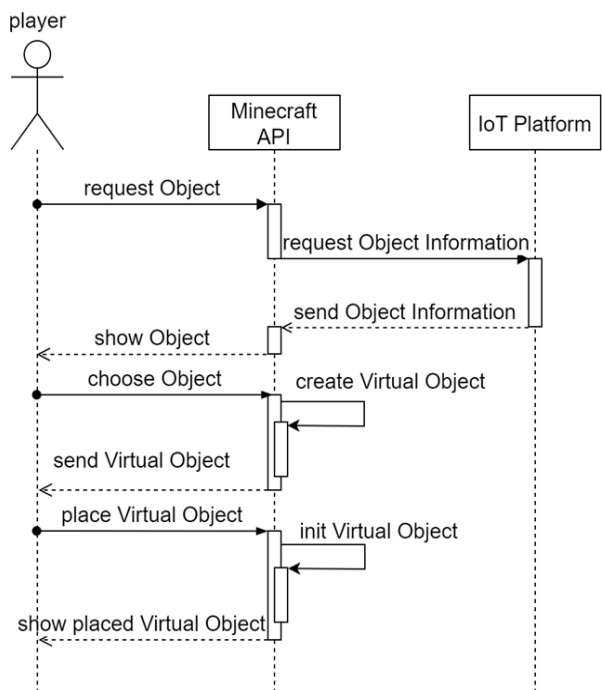


그림 3. 사물 등록의 시퀀스 다이어그램
Fig. 3. Sequence diagram of object registration

그림 4는 마인크래프트에서 현실 사물의 데이터를 일정 시간마다 동기화하는 알고리즘이다. 알고리즘은 7단계로 이루어져 있다. 1단계, Virtual Object 리스트를 준비한다. 2단계, Virtual Object를 하나 꺼낸다. 3단계, Actuator Object인지 Sensor Object인지 확인한다. 4단계 각 Object에 있는 데이터를 가지고 현실 사물의 데이터를 가져온다. 5단계 가져온 데이터를 가지고 마인크래프트 내 사물에 각 로직에 맞게 작동을 시킨다. 6단계 리스트 내에 모든 요소에 반복한다. 7단계 일정 주기로 계속해서 동기화 작업을 진행한다.

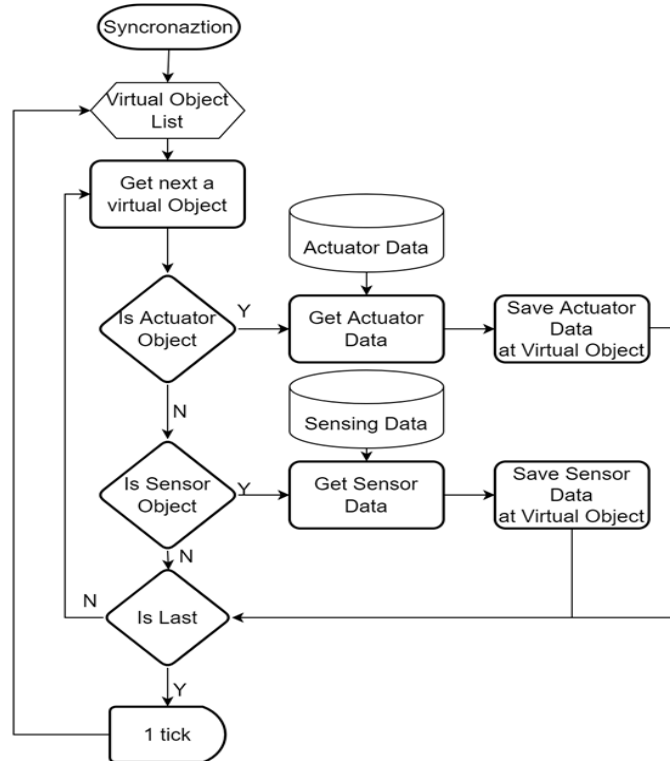


그림 4. 데이터 동기화 알고리즘
Fig. 4. Algorithm of data synchronization

그림 5는 마인크래프트와 IoT 플랫폼 간 액추에이팅을 위해서 어떤 과정을 통해 액추에이팅이 되는지 보여준다. 시작은 플레이어가 관리 시스템에 등록된 사물과 상호작용이다. 모듈이 액추에이팅을 위한 UI를 보여줘서 플레이어가 어떻게 제어할 수 있을지 보여준다. 플레이어가 맞는 형식으로 상호작용을 완료하면 IoT 플랫폼에 디바이스 액추에이터를 제어하기 위한 메시지 포맷에 맞춰서 보내준다.

마지막으로 IoT 플랫폼은 액추에이팅 메시지를 통해 해당 액추에이터에 전달하여 작동하게 된다.

3.3 사물 연동을 위한 자료구조

현실 세계의 사물과 마인크래프트 내의 사물 간 연결성을 부여하기 위해서는 각 사물을 구조화하고 매핑시키는 작업이 요구된다. 즉, 현실 세계, IoT 플랫폼, Virtual Object, 마인크래프트 사물을 연결하기 위해서는 이들을 연속하여 매핑해야 한다.

그림 6은 현실 세계 사물과 마인크래프트 사물을 연결하기 위한 자료구조를 보인다. 현실 사물은 IoT 메타데이터로 매핑이 된다. IoT 메타데이터와 마인크래프트 사물이 Virtual Object로 매핑이 되면서 양쪽 끝이 있었던 현실 사물과 마인크래프트의 사물의 연결성이 생긴 것을 볼 수 있다. LED를 현실 사물 예시로 IoT 플랫폼에 메타데이터로 'device 01'이 등록이 되어 있고 LED Object를 통해 'device 01'과 'Redstone_lamp'가 의미상 연결이 되어 LED와 'Redstone_lamp'가 연결이 된다.

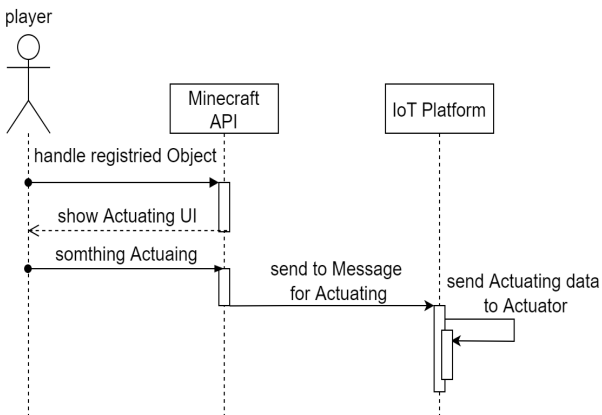


그림 5. 액추에이팅의 시퀀스 다이어그램
Fig. 5. Sequence diagram of actuating

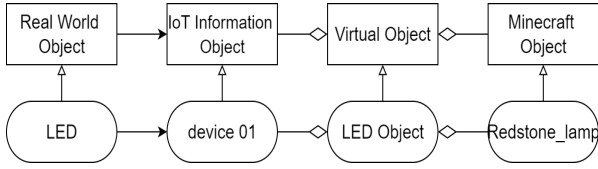


그림 6. 현실 사물과 마인크래프트 사물 간 자료구조
Fig. 6. Data structure between real world object and minecraft object

3.4 oneM2M 기반의 디바이스 등록 규칙

제안 시스템은 상호운용성 향상과 유연한 정보 표현을 위하여 디바이스 레지스트리 기반의 IoT 플랫폼을 사용한다. 이 절은 디바이스 레지스트리에서 디바이스를 등록할 때, 상호운용성의 보장 및 향상을 위하여 oneM2M SDT를 활용하고 이에 기반한 디바이스 등록 규칙을 제시한다. oneM2M SDT이란 oneM2M에서 IoT Device를 고유의 기능과 데이터의 정보를 모델링한 것 일종의 템플릿이다. 표 1은 oneM2M SDT에 정의된 용어 중 본 논문의 이해에 필요한 용어에 대한 정의를 보인다.

표 1. SDT 용어와 설명[17]

Table 1. Terms and descriptions of SDT

Term	Description
Domain	Unique name, or "wrapper" which acts like a namespace, set by the organization, company, or project creating the template, allowing reference to a package of definitions for the contained ModuleClasses and device definitions.
DeviceClass	A physical, addressable, identifiable appliance, sensor, or actuator.
SubDevice	A device which may be embedded in or attached to a device. It is not designed to be operated as a standalone device.
ModuleClasses	Specification of a single service with one or more service methods, the involved abstracted data model and related events.
DataPoint	Datapoints which aspects of a functionality that can be read and/or written.
Actions	Actions which consist of more complex sequences of operations.
Property	Property are used to append to ProductClasses DevicesClasses and their ModuleClass elements with arbitrary additional information.
Event	Events which can be signaled ("published") by devices asynchronously.

또한, 표 2는 제안 시스템이 사용하는 IoT 플랫폼의 중요 용어들도 표 2에 대한 정의를 보인다.

SDT를 디바이스 레지스트리에 적용하기 위해선, 각 요소를 디바이스 레지스트리에 어떻게 적용할지에 대한 기준이 필요하다. 표 3은 SDT를 디바이스 레지스트리에 등록하기 위한 규칙을 보인다.

표 2. 디바이스 레지스트리 용어와 설명

Table 2. Terms and descriptions of device registry

Term	Description
Device deployment information	Deployed device information (like device_id, device_name, item_id, system, deployment_tiem, deployment_location, latitude, longgitude)
Item	A information model defining specifications of device.
Item common information	Common elements of device(model_name, device_type, manufacturer, category)
Item specific information	Specific information on devices is defined data item by key-value format and can be a module by group.

표 3. SDT를 device registry에 등록하기 위한 규칙

Table 3. Rules for registering SDT to device registry

Rule no.	Description
1.1	Domain is category at item common information.
2.1	Deviceclass is item.
2.2	Deviceclass name is device type at item common information.
2.3	Deviceclass instance name is model name at item common information.
3.1	Moduleclass is existed in item specific information.
3.2	DataPoints is followed Rule no. 4.
4.1	A Row in DataPoints can describe a group at item specific information.
4.2	Each group have moduleclass name for connecting to other groups. - Key is moduleclass, value is moduleclass name
4.3	R/W and Name is a pair, the key is sensor if R/W value is R or actuator if R/W value is RW, value is Name.
4.4	For changing a value in actuator of device, add actuating request form by "request_type"
4.5	Other columns are keys, data in a row is value. - Naming for key is followed snake case. - When several rows are described about one feature, they should be described one group. - Unit should rename "unit_of_measure". - A value of type has no namespace.

규칙 1.1은 Domain에 관한 내용으로 디바이스 레지스트리의 item common information category와 매칭된다. 규칙 2.1~2.3은 Deviceclass에 관한 항목으로 Deviceclass가 어디에 매칭되고 추가적인 정보를 어떻게 등록해야 하는지 하위 규칙으로 정의된다. 규칙 3.1~3.2는 Moduleclass에 관한 내용으로 해당 정보들이 어느 부분에 적용이 되고 하위 요소가 어떤 규칙을 따르는지 보여준다. 규칙 4.1~4.5는 DataPoint에 관한 내용으로 item specific information에 등록하는 규칙을 세부적으로 기술한다.

IV. 구현 결과

이 장에서는 제안 시스템을 구현하고 세 가지 시나리오를 통하여 제안 시스템이 잘 동작하는지 검증한다.

4.1 구현 환경

제안 시스템을 위한 구현 환경은 IoT 디바이스 게이트웨이, IoT 플랫폼 서버, 마인크래프트 서버 및 클라이언트와 같이 구성된다. 먼저 IoT 디바이스 게이트웨이는 IoT 디바이스로부터 발생하는 데이터를 서버로 전송하고 액추에이터를 제어하는 역할을 하며 Raspberri Pi 3 Model B+를 사용한다.

표 4는 IoT 플랫폼 서버의 스펙을 보이며, MySQL 데이터베이스와 MQTT 브로커인 Mosquitto 그리고 파이썬으로 개발된 데이터 수집 에이전트가 설치되었다. 디바이스 레지스트리는 JSP 기반의 웹서비스로 이 서버에 함께 운영된다. IoT 플랫폼으로 사용되는 게이트웨이는 라즈베리를 사용하였고, 서버는 리눅스 서버로 작동된다. 표 4는 IoT 서버의 스펙이다.

표 4. IoT 서버 환경

Table 4. Environment of IoT server

Feature	Specification
OS	Ubuntu 18.04.6
CPU	Intel Core i5-7640x
RAM	32GB
SSD	500GB
Hard disk	8TB

표 5는 마인크래프트 플러그인이 작동할 서버와 클라이언트가 실행된 컴퓨터를 보인다.

표 5. 마인크래프트 클라이언트 및 서버 환경

Table 5. Environment of client and server on minecraft

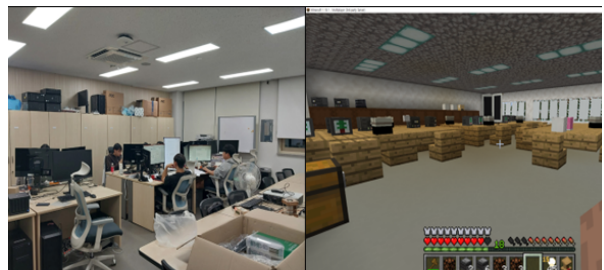
Feature	Specification
OS	Window 10(x64)
CPU	Intel core i7-6700
RAM	24GB
SSD	1012GB

4.2 시나리오

본 논문은 제안 시스템의 검증을 위하여 실제 IoT 디바이스를 개발하고 3가지 시나리오를 설정하고 IoT 플랫폼과 API 연동을 통해 마인크래프트에서 모니터링 및 제어하는지를 검증한다.

본 논문에서 제시하는 시나리오는 스마트오피스 환경을 가정한다. 스마트오피스에서는 IoT 플랫폼에 공기청정기, 전등, 에어컨 등과 같은 다양한 사물들이 존재한다. 이 사물들은 마인크래프트 내에서도 연계되어 현실 사물과 게임 내 오브젝트와 연동된다. 그림 7은 현실 세계와 마인크래프트 내의 스마트오피스 구현 결과를 보인다.

현실 세계의 스마트오피스에서 각 사물은 제안하는 IoT 플랫폼과 연동되어 디바이스 레지스트리에 그 정보들이 저장된다. 또한 서버로 데이터들이 수집되어 실시간 모니터링이 가능하다. 마인크래프트 내에서도 실제 스마트오피스 현장을 그대로 구현해 놓은 후 마인크래프트 플러그인을 통해 내 특정 오브젝트를 현실 세계의 IoT 디바이스와 연동시킨다.



(a) 현실 세계 (a) Real world (b) 마인크래프트 (b) Minecraft

그림 7. 스마트오피스의 구현 결과
Fig. 7. Implementation of a smart office

본 논문은 IoT 디바이스와 마인크래프트의 동기화 시스템이 정상 작동하는지 검증을 위하여 문열림 감지 센서, 미세먼지 측정기, 공기 청정기의 세 가지 디바이스를 개발한다.

이러한 IoT 디바이스들은 각 센서 및 액추에이터를 포함한 모든 디바이스 정보가 oneM2M SDT에 기반하여 모델링 되고 IoT 플랫폼 내의 디바이스 레지스트리에 정보화되어 등록된다. 등록된 디바이스들은 제안 시스템에서 정의된 자료구조에 의해 마인크래프트의 플러그인을 통하여 오브젝트와 동기화된다. 이에 따라 IoT 디바이스에서 수집된 데이터를 마인크래프트 내에서 모니터링하고 게임 내 행동을 통해 현실 사물인 IoT 디바이스를 제어할 수 있다.

첫 번째 시나리오는 센서 모니터링 검증을 위한 것으로, 문열림 감지 센서가 있는 문을 사람이 여닫을 때 마인크래프트에서도 문열림이 제대로 표현되는지를 확인한다. 문열림 감지 센서는 AMS-10C 센서를 이용한 아두이노 기반으로 구현하였으며 스마트오피스의 문에 부착하였다. 그림 8은 문열림 감지 센서의 구현 결과를 보이며, 현실에서 사람이 지나다니면서 문을 여닫을 때 마인크래프트에서도 동일하게 동기화됨을 확인하였다.



(a) 현실 세계 (a) Real world (b) 마인크래프트 (b) Minecraft

그림 8. 문열림감지 센서의 구현 결과
Fig. 8. Implementation of door open sensor

두 번째 시나리오는 액추에이팅 통한 양방향 제어 검증을 위한 것으로, 미세먼지 측정기에 부착된 RGB LED를 마인크래프트에서 제대로 제어할 수 있는지 확인한다.

구현된 미세먼지 측정기는 아두이노에 기반하여 GP2Y 미세먼지 측정 센서, LED 디바이스는 RGB

LED, CDS 조도 센서, DHT11 온습도 센서를 부착하고 있으며, HC-06 블루투스 모듈을 통하여 게이트웨이와 통신한다. 이 디바이스는 종합적인 스마트 오피스 환경을 모니터링할 수 있으나 두 번째 시나리오에서는 LED의 제어만을 테스트한다. 그림 9는 미세먼지 측정기의 구현 결과를 보인다. 마인크래프트 내에서 사용자가 미세먼지 측정기에 부착된 LED의 전원을 켜기 위해 해당 오브젝트에 동작하면 마인크래프트의 플러그인에서 제안 IoT 플랫폼의 MQTT에 메시지를 보내 현실에 있는 미세먼지 측정기의 LED를 켜도록 한다. 이는 실시간으로 IoT 플랫폼이 해당 디바이스의 현재 LED 상태를 모니터링하여 다시 데이터베이스에 현재 상태를 수집하게 되고 이는 마인크래프트 내에서 해당 오브젝트가 실시간 모니터링에 의해 LED가 켜져 있음을 반영하는 방식으로 동기화한다.



(a) 현실 세계 (a) Real world (b) 마인크래프트 (b) Minecraft

그림 9. 미세먼지 측정기의 구현 결과
Fig. 9. Implementation of fine dust sensor

세 번째 시나리오는 공기청정기 디바이스를 oneM2M SDT에 기반하여 디바이스 레지스트리 등록시키고 이 모델이 제안 시스템에서 제대로 동작하여 동기화되는지를 검증한다.

공기청정기 디바이스는 DHT11 온습도 센서, PM-D4 미세먼지 센서, LCD 모니터, 모터가 부착되어 스마트오피스의 공기 질에 따라 공기청정기를 동작시킬 수 있는 디바이스이다. 그림 10은 공기청정기의 구현 결과를 보인다. 공기청정기의 경우 현재 공기 질 및 온습도 등 주변 환경에 대한 데이터를 수집할 수 있으며 공기 질에 따른 공기청정기 팬을 동작시킬 수 있다. 공기청정기의 팬은 0~255 사이의 값으로 설정할 수 있으며, 이는 마인크래프트

트 내의 해당 오브젝트에 0~255 사이의 값을 설정하는 것으로 동기화시킬 수 있다.

한편, 그림 11은 SDT에 기반하여 디바이스 레지스트리에 등록된 공기청정기의 정보를 보인다.



(a) 현실 세계 (a) Real world (b) 마인크래프트 (b) Minecraft

그림 10. 공기청정기의 구현 결과
Fig. 10. Implementation of air purifier

Common Information

Attribute	Value
Item id	11
Registration time	2022-10-12 12:45:32
Model name	deviceAirPurifier
Device type	customDeviceAirPurifier
Manufacturer	DSEM
Category	Home

Specific Information

Group	Key	Value
2	moduleclass	airQualitySensor
2	sensor	pm25
2	type	integer
2	unit_of_measure	µg/m3
3	moduleclass	airQualitySensor
3	sensor	humidity
3	type	float
3	unit_of_measure	pct
4	moduleclass	airQualitySensor
4	sensor	temperature
4	type	float
4	unit_of_mesure	°C
5	moduleclass	airFlow
5	actuator	speed
5	type	integer
5	min_value	0
5	max_value	255
5	request_type	range_of_numbers

그림 11. 디바이스 레지스트리에 등록된 공기청정기 정보에 대한 스크린샷

Fig. 11. Screenshot of information about air purifier registered in the device registry

V. 결 론

기존에는 상용화된 메타버스 서비스를 기반으로 양방향 디지털 트윈을 구현할 때 상호운용성, 센서 모니터링, 액추에이터, 동기화 시스템을 모두 고려한 시스템은 없었다.

본 논문은 상용화된 메타버스 게임인 마인크래프트에서 양방향 디지털 트윈의 구현을 위한 IoT 데이터 동기화 시스템을 제안하였다. 이를 위하여, oneM2M SDT와 디바이스 레지스트리를 활용하여 상호운용성을 향상했으며, IoT 디바이스와 마인크래프트의 오브젝트를 연결하고 제어하기 위해 자료구조, 동기화 알고리즘, 액추에이팅 시퀀스 다이어그램을 설계하였다. 제안 시스템은 마인크래프트의 플러그인 통해 구현하였으며 세 가지 시나리오를 통하여 제안 시스템이 잘 동작함을 검증하였다.

본 연구는 기존의 상용화된 메타버스를 활용하는 연구들이 다루지 않았던 상호운용성[12] 및 액추에이팅[13][14]을 모두 고려하여, 마인크래프트 내에서 실시간 동기화 시스템을 구축함으로써 실시간 모니터링 및 액추에이팅을 가능하게 한다는 장점을 지닌다.

향후 연구로는 양방향 디지털 트윈의 다음 단계인 지능형 트윈을 구현하기 위하여 마인크래프트 내에서 데이터 분석 및 인공지능 학습을 통한 예측 시뮬레이션을 설계하고 구현할 예정이다.

References

[1] S. Choi, J. Woo, J. Kim, W. Choi, J. Kim, and J. Lee, "A Survey and Analysis of Research on Digital Twin in Korea", Korean Journal of Computational Design and Engineering, Vol. 26, No. 1, pp. 59-69, Mar. 2021. <https://doi.org/10.7315/CDE.2021.059>.

[2] D. Jeong, S. Kim, and Y. Kim, "Technical definition and detailed development five-level model of digital twin", IITP weekly technology trend, Vol. 1983, No. 2, pp. 15-27, Apr. 2022.

[3] R. Saracco, "DIGITAL TWINS' FUTURE", <https://cmte.ieee.org/futuredirections/2021/01/26/digital-twins-future> [accessed: Oct. 05, 2022]

- [4] S. Yang, J. Cho, and J. Kim, "Interworking the physical world and virtual world, direction of metaverse autonomous twin technology", *Broadcasting and Media Magazine*, Vol. 26, No. 3, pp. 20-23, Jan. 2022.
- [5] J. Yoo, J. Lee, and W. Choi, "Technology and standardization trend of digital twin", *The Journal of The Korean Institute of Communication Sciences*, Vol. 38, No. 9, pp. 40-47, Aug. 2021.
- [6] S. Go, H. Jeong, J. Kim, and Y. Shin, "Concepts and direction of Metaverse", *Korea Information Processing Society Review*, Vol. 28, No. 1, pp. 7-16, Mar. 2021.
- [7] C. W. Ahn, "Metaverse and digital twin", *Industrial Engineering Magazine*, Vol. 28, No. 4, pp. 23-27, Dec. 2021.
- [8] W. Jang, D. Jeong, and S. Lee, "A Method of IoT Object Control based on MQTT for Enhancing Object Mobility", *The Journal of Korean Institute of Information Technology*, Vol. 20, No. 3, pp. 107-119, Mar. 2022. <https://doi.org/10.14801/jkiit.2022.20.3.107>.
- [9] S. Yun, H. Kim, H. Shin, H. S. Chin, and W. T. Kim, "A Novel Reference Model for Cloud Manufacturing CPS PlatformBased on oneM2M Standard", *KIPS transactions on computer and communication systems*, Vol. 8, No. 2, pp. 41-56, Feb. 2019. <https://doi.org/10.3745/KTCCS.2019.8.2.41>.
- [10] J. Kim, W. Choi, M. Song, and S. Lee, "Design and Implementation of IoT Platform-based Digital Twin Prototype", *Journal of Broadcast Engineering*, Vol. 26, No. 4, pp. 356-367, Jul. 2021. <https://doi.org/10.5909/JBE.2021.26.4.356>.
- [11] S. Kang, J. Park, and K. Chung, "Interworking System based on oneM2M Architecture for Interoperability between Heterogeneous IoT Platform", *Proc. of the Korean Information Science Society Conference 2017*, pp. 440-442, Dec. 2017.
- [12] A. A. Simiscuka, T. M. Markande, and G. M. Muntean, "Real-Virtual World Device Synchronization in a Cloud-Enabled Social Virtual Reality IoT Network", in *IEEE Access*, Vol. 7, pp. 106588-106599, Aug. 2019. <http://doi.org/10.1109/ACCESS.2019.2933014>.
- [13] S. Son, S. Ham, M. Kim, J. Rho, J. W. Lee, M. Kang, and J. Seo, "Digital Twins of IoT Motion and Vibration Sensors for Attendance Confirmation and Notification in the Metaverse", *Proc. of Symposium of the Korean Institute of communications and Information Sciences*, pp. 327-328, Jun. 2022.
- [14] Y. Lee, H. Kwon, H. Moon, H. B. Choi, H. Lee, J. Seo, and C. Lee, "Real-Time Interworking between IoT Environmental Sensors and Their Digital Twins in the Metaverse", *Proc. of Symposium of the Korean Institute of communications and Information Sciences*, pp. 325-326, Jun. 2022.
- [15] M. Lee, "The Development of Instruction Model for SW Education using the Minecraft Platform", *Journal of the Korea Society of Digital Industry and Information Management*, Vol. 15, No. 3, pp. 119-128, Sep. 2019. <https://doi.org/10.17662/ksdim.2019.15.3.119>.
- [16] J. Y. Moon, T. M. Yeo, S. I. Choi, W. Ko, "Research and Realization of a Regional Power System Based on Minecraft", *Proc. of the 53th KIEE Summer Conference 2022*, Vol. 53, pp. 50-251, Jul. 2022.
- [17] Andreas Kraft, "SDT", <https://git.onem2m.org/MAS/SDT> [accessed: Nov. 09, 2022]

저자소개

양 병 헌 (Byeongheon Yang)



2017년 3월 ~ 2023년 2월 :
군산대학교 소프트웨어학과(학사)
관심분야 : 사물인터넷, 메타버스,
IoT

정 현 준 (Hyunjun Jung)



2008년 : 삼육대학교
컴퓨터과학과(학사)
2010년 : 송실대학교
컴퓨터과학과(공학석사)
2017년 : 고려대학교
컴퓨터·전파통신공학과(공학박사)
2017년 8월 ~ 2020년 8월 :
광주과학기술원 블록체인 인터넷경제연구센터
2021년 ~ 현재 : 군산대학교 소프트웨어학부 조교수
관심분야 : 블록체인, 데이터 사이언스, 센서 네트워크,
사물인터넷, 머신러닝

이 석 훈 (Sukhoon Lee)



2009년 2월 : 고려대학교
전자및정보공학부(학사)
2011년 2월 : 고려대학교
컴퓨터·전파통신공학과(공학석사)
2016년 2월 : 고려대학교
컴퓨터·전파통신공학과(공학박사)
2016년 3월 ~ 2017년 3월 :
아주대학교 의료정보학과 연구강사
2017년 4월 ~ 현재 : 군산대학교 소프트웨어학부 부교수
관심 분야 : 사물인터넷, 데이터 공학, 에지 컴퓨팅,
디지털트윈, 연합학습