# 디지털 트윈 기반 도시 미기후 및 대기질 관리 시스템 분석 및 설계 보고서

**프로젝트명:** S-City 환경 모니터링 및 최적 제어 플랫폼 구축

**작성일:** 2024년 05월 20일

**작성자:** 시스템 아키텍트 팀

**버전:** v1.0

## 1. 개요 (Overview)

### 1.1 배경

급격한 도시화로 인한 고층 빌딩 밀집 현상은 도시 내 바람길을 차단하여 '열섬 현상(Heat Island)'과 '미세먼지 정체'를 유발하고 있음. 기존의 사후 대응 방식(민원 발생 후 살수차 투입 등)은 골든타임을 놓치기 쉬우며 근본적인 해결책이 되지 못함.

### 1.2 목적

현실 세계의 도시 환경을 가상 공간(Digital Twin)에 복제하여 실시간으로 모니터링하고, 물리 엔진(CFD) 기반 시뮬레이션을 통해 문제 상황을 예측하며, 최적의 대응 시나리오를 도출하여 선제적으로 환경을 제어하는 플랫폼을 구축함.

## 2. 유스케이스 시나리오 (Use Case Scenario)

### 2.1 시나리오 명: 폭염 및 대기 정체 선제 대응

**관점:** 사용자(관제 팀장) 및 시스템

| **단계** | **내용** |
| --- | --- |
| **1. 징후 포착** | **[System]** A지구의 풍속이 0.5m/s로 떨어지고 기온이 급상승함을 감지. AI 모델이 2시간 후 '복합 열섬 현상' 발생 확률 92%로 예측 및 경고 발송. |
| **2. 시뮬레이션** | **[User]** 가상 공간에서 대응책 모색을 위해 시뮬레이션 모드 실행.  **[System]** '살수차 단독 투입' vs '살수차 + 옥상 환기팬 가동' 시나리오에 대해 CFD 유체 해석 수행. |
| **3. 의사결정** | **[System]** 복합 처방 시 기온 -4℃, 미세먼지 농도 '보통' 회복 예측 결과 제시.  **[User]** 복합 처방 시나리오(Scenario B) 승인 및 실행 명령. |
| **4. 현실 제어** | **[System]** 현장 IoT 액추에이터(환기팬) 가동 명령 전송 및 자율주행 살수차 호출. 시민들에게 스마트폰으로 주의 알림 발송. |
| **5. 결과 확인** | **[System]** 30분 후 센서 데이터 분석 결과, 기온 하강 및 대기 흐름 회복 확인. 상황 종료 보고서 자동 생성. |

## 3. 요구사항 정의 (Requirements)

### 3.1 기능적 요구사항 (Functional)

1. **실시간 모니터링:** IoT 센서(온도, 습도, 미세먼지, 풍향/풍속) 데이터를 수집하여 3D 지도에 시각화해야 한다.
2. **이상 징후 감지:** AI 모델은 수집된 데이터를 분석하여 설정된 임계치를 넘거나 패턴이 이상할 경우 즉시 경보를 울려야 한다.
3. **시뮬레이션:** 사용자의 요청에 따라 가상 환경에서 바람길 및 오염 물질 확산 모델(CFD)을 수행할 수 있어야 한다.
4. **장치 제어:** 분석 결과에 따라 현장의 제어 장치(살수차, 환기팬, 스프링클러 등)에 원격 명령을 내릴 수 있어야 한다.

### 3.2 비기능적 요구사항 (Non-Functional)

1. **실시간성:** 센서 데이터 수집부터 대시보드 표출까지의 지연 시간(Latency)은 1초 이내여야 한다.
2. **신뢰성:** 시뮬레이션 예측 정확도는 85% 이상을 유지해야 한다.
3. **확장성:** 향후 센서 및 관제 구역 확장에 유연하게 대응할 수 있는 마이크로서비스 아키텍처(MSA)를 적용한다.

## 4. OOA/D 분석 및 설계 (Analysis & Design)

### 4.1 유스케이스 다이어그램 (Use Case Diagram)

시스템의 주요 기능과 액터 간의 상호작용을 정의함.

* **Actors:** 환경 관리자, IoT 센서/액추에이터, 외부 기상청 API
* **Use Cases:** 모니터링, 이상 감지, 시뮬레이션 실행, 장치 제어

@startuml  
left to right direction  
skinparam packageStyle rectangle  
skinparam usecase {  
 BackgroundColor White  
 BorderColor Black  
}  
  
actor "환경 관리자" as Admin  
actor "IoT 센서/액추에이터" as IoT  
actor "기상청 API" as WeatherAPI  
  
rectangle "디지털 트윈 플랫폼" {  
 usecase "실시간 모니터링" as UC1  
 usecase "이상 징후 자동 감지" as UC2  
 usecase "시뮬레이션 및 예측" as UC3  
 usecase "장치 제어 및 대응" as UC4  
}  
  
Admin --> UC1  
Admin --> UC3  
Admin --> UC4  
UC1 <.. UC2 : <<include>>  
IoT --> UC1  
UC4 --> IoT  
WeatherAPI --> UC1  
@enduml

### 4.2 클래스 다이어그램 (Class Diagram)

시스템의 정적 구조 및 데이터 모델 설계.

* **DigitalTwinModel:** 전체 시스템의 진입점.
* **SimulationEngine:** CFD 및 예측 로직 담당.
* **IoTDevice:** 센서 및 액추에이터의 부모 클래스.

@startuml  
skinparam classAttributeIconSize 0  
skinparam class {  
 BackgroundColor White  
 BorderColor Black  
}  
  
class "DigitalTwinModel" as DT {  
 - twinId : String  
 + syncData() : void  
 + visualize() : void  
}  
  
class "SimulationEngine" as Sim {  
 + runCFD(params) : Result  
 + predictFuture(time) : Prediction  
}  
  
abstract class "IoTDevice" as Device {  
 - deviceId : String  
 + connect() : Boolean  
}  
  
class "Sensor" {  
 + measure() : Data  
}  
  
class "Actuator" {  
 + execute(cmd) : Result  
}  
  
DT ..> Sim : Uses  
DT o-- Device : Aggregation  
Device <|-- Sensor  
Device <|-- Actuator  
@enduml

### 4.3 시퀀스 다이어그램 (Sequence Diagram)

'이상 징후 감지 및 대응' 시나리오의 동적 흐름 설계.

@startuml  
skinparam backgroundColor White  
skinparam sequence {  
 LifeLineBorderColor Black  
 ParticipantBorderColor Black  
 ParticipantBackgroundColor White  
}  
  
actor "관리자" as User  
participant "플랫폼" as Platform  
participant "AI 감지기" as AI  
participant "시뮬레이션 엔진" as Sim  
participant "제어 장치" as Device  
  
User -> Platform : 모니터링 시작  
activate Platform  
Platform -> AI : 실시간 데이터 분석 요청  
activate AI  
AI --> Platform : [경고] 열섬 현상 감지  
deactivate AI  
  
Platform -> User : 경고 알림 전송  
User -> Platform : 대응 시나리오 시뮬레이션 요청  
Platform -> Sim : CFD 해석 수행  
activate Sim  
Sim --> Platform : 최적 시나리오 도출  
deactivate Sim  
  
Platform -> User : 결과 보고 및 승인 요청  
User -> Platform : 대응 실행 승인  
Platform -> Device : 가동 명령 전송  
activate Device  
Device --> Platform : 가동 완료  
deactivate Device  
deactivate Platform  
@enduml

### 4.4 시스템 아키텍처 (Package/Deployment)

* **Client:** 웹 대시보드, 3D 뷰어
* **Server:** 모니터링 서비스, 시뮬레이션 서비스, AI 모델
* **Infra:** IoT 게이트웨이, 시계열 DB(InfluxDB), 공간 DB(PostGIS)

## 5. 결론 및 기대 효과 (Conclusion)

본 디지털 트윈 시스템 구축을 통해 S-City는 다음과 같은 효과를 기대할 수 있음.

1. **데이터 기반 과학적 행정:** 경험이나 직관이 아닌, 데이터와 시뮬레이션 결과에 기반한 환경 관리.
2. **선제적 재난 예방:** 문제가 발생하기 전 예측(Prediction)하여 대응함으로써 시민의 안전과 건강 보호.
3. **운영 효율성 증대:** 최적의 시점에 필요한 장비만 가동함으로써 에너지 및 운영 비용 절감.

*End of Document*