

Development of a Fire Detection Framework for Underground Utility Tunnels Using Emulators and Open API

Hyun Seong Kil[†] · Yoo Dong Young^{††}

ABSTRACT

Underground utility tunnels, which house essential urban infrastructure such as communication lines, water pipes, and power lines, are crucial for modern city life. In the event of a disaster, these tunnels can cause significant inconvenience to urban residents and impact public safety. To reduce inconvenience and enhance public safety, it is vital to quickly and accurately identify and assess disasters and share information with relevant agencies. This system collects data from various sensors, uses message queues to asynchronously produce and consume events, and implements a framework and processing pipeline to swiftly and accurately detect and assess fire disasters, utilizing an Open API for sharing.

Keywords : Underground Utility Tunnel, Fire, Detection, Message Queue, Framework

에뮬레이터와 Open API를 활용한 지하공동구 화재탐지 프레임워크 개발

현 성 길[†] · 유 동 영^{††}

요 약

지하공동구는 현대 도시 생활의 필수라 여길 수 있는 통신선, 상수관, 전력선등이 지나가는 곳으로 재난 발생시 도시민 불편을 초래하며 나아가 국민의 안전에 영향을 줄 수 있다. 시민 불편 감소 및 국민 안전 재고를 위해 신속하고 정확하게 재난을 식별·판단하고 유관기관과 정보를 공유하는 일은 대단히 중요하다. 다양한 센서로부터 데이터를 수집한 후 메시지큐를 이용하여 비동기적으로 이벤트를 생산·소비하고 신속·정확하게 화재 재난을 탐지·판단하는 프레임워크 및 처리 흐름(Pipeline), 그리고 공유를 위한 Open API를 활용한 시스템을 구현한다.

키워드 : 지하공동구, 화재, 탐지, 메시지큐, 프레임워크

1. 서 론

2018년도 KT 서울 아현지사 화재 사고, 2018년도 고양 백성동 온수배관 파열 사고등 지하공동구내 대형 피해 사례들이 속속 보고 되고 있는 작금에 “디지털 트윈 기반의 지하공동구 화재·재난지원 통합플랫폼 기술개발”은 다부처 과제로 진행되고 있으며 빅데이터, 인공지능, 모델링 및 시뮬레이션등 지능화 기술을 활용하여 지하공동구에서 발생 가능한 문제점을

분석하고 최적화 방안을 도출하여 현장에 적용하는 기술을 연구하는 과제이다[1]. 지하공동구는 지하매설물(전선로, 통신선로, 상·하수도관, 열수송관)을 공동 수용함으로써 도로중복 굴착방지, 도시미관 개선, 도로구조 보전 및 교통의 원활한 소통을 위해 지중(地中)에 설치하는 시설물이다[2]. 재난 발생시 그 지역의 전력, 통신, 상수등이 피해를 입으며 도시를 마비시키고 국민의 안전에 큰 영향을 미칠 수 있다. 이에, 국토교통부는 기반시설 관리체계 재정비와 인프라 노후화에 대해 적극 대응하기 위해 “지속가능한 기반시설 관리 기본법”을 제정하였다[3]. 지하공동구는 지상과 다르게 환기 조건이 좋지 못하고 소방 접근성도 어려운 편이며 구조적 특성상 내부가 복잡하고 출입구가 제한적이어서 대피와 진압이 어렵다. 좁고 긴 형태의 밀폐·폐쇄된 공간이므로 산소공급 부족, 유독물질 발생, 시야장애등 사람이 직접 진입하는데 어려움이 있어 큰 피

※ 이 논문은 정부(과학기술정보통신부, 국토교통부 및 산업통상자원부)의 재원으로 정보통신기획평가원(IITP) 지원을 받아 수행된 연구임(No. 2020-0-00061, 디지털트윈 기반의 지하공동구 화재·재난 지원 통합플랫폼 기술개발).

† 준 회원 : 바이브컴퍼니 수석 연구원

†† 중신회원 : 홍익대학교 소프트웨어융합과 부교수

Manuscript Received : October 30, 2024

Accepted : November 27, 2024

* Corresponding Author : Yoo Dong Young(ydy@hongik.ac.kr)

해로 확산될 가능성이 높기에 정보 수집, 상황 판단, 재난 상황 전파의 처리 흐름을 통해 피해 확산을 방지해야 한다.

본 논문은 지하공동구의 재난 상황을 대상으로 하며, 특히 화재 재난에 중점을 두었다. 재난 발생시 부정확한 정보에 의한 혼란유발 방지 및 피해 최소화를 위해 실시간으로 정확한 정보 수집, 빠른 상황 파악, 선제적 대응, 신속하고 안정적인 재난 정보 전파를 위한 프레임워크(Framework) 및 처리 절차(Pipeline)를 제안하기 위해 가상 데이터 생성 메시지 큐, Open API(Application Programming Interface), JSON(JavaScript Object Notation)등의 기술을 활용해 구현하였다. IoT 센서에 의해 수집된 데이터는 모두 데이터베이스에 저장하며 지하공동구에서 발생할 수 있는 재난중 화재는 임계값을 설정해 특정 임계값 이상의 데이터만 이벤트를 발생시켜 네트워크 대역폭을 줄이고, 메시지 큐와 JSON형식의 데이터를 이용해 시스템의 제한없이 독립적으로 비동기적으로 메시지를 주고 받을 수 있도록 한다. 탐지 정확도를 높이기 위해 특정 이벤트 발생시 연관 데이터를 종합적으로 계산하도록 하였으며 특정 이벤트와 관련된 센서 데이터의 종류는 표로 정리하여 데이터베이스의 테이블로 저장하여 관리한다. 지하공동구에 직접적으로 재난을 발생시키기 어려운 관계로 가상 데이터 생성기를 구현하여 실험하였으며 여러개의 센서에서 동시 다발적으로 데이터를 전송한다는 가정하에 스레드 기능을 이용해 데이터를 발생시켰다.

2. 선행연구

“디지털 트윈 기반의 지하공동구 화재·재난지원 통합플랫폼 기술”은 빅데이터, 인공지능, IoT(Internet of Things), 네트워크, 지능형 로봇 등 첨단 ICT(Information and Communication Technology) 기술을 개발하는 과학기술정보통신부를 주관부처로 행정안전부, 국토교통부 및 산업통상자원부가 협력하는 다부처 사업으로 2020년 4월부터 진행되는 디지털 트윈 관련된 국내 최초 연구개발 사업이다. 본 사업에서는 현장정보 센싱, 데이터 통합분석, 모델링 및 시뮬레이션, 초정밀 가시화 등 디지털 트윈 핵심요소 기술을 개발하여 지하공동구 화재 재난 관리를 위한 서비스를 제공한다[1]. 지하공동구 환경에서 음향정보를 수집하기 위한 프로토타입 모듈을 개발하여 실험을 하였으며 음향 수집 장치 모듈은 음향을 수집한 후 음향을 RTP(Realtime Transport Protocol)을 이용하여 음향 수신 저장 모듈로 전송하였다. 음향 수신 저장 모듈은 데이터를 파싱하여 시간과 파일 정보를 분리 추출한 후 데이터베이스에 저장하였으며 Rest(Representational State Transfer) API를 이용하여 데이터에 접근할 수 있도록 하였다[4]. OASIS(Organization for the Advancement of Structured Information Standards)에서 국제 표준으로 배포하는 공통경보 프로토콜(CAP-v1.2)을 지하공동구 재난 환경에 최적화되고 신속하고 풍부한 정보를 전파할 수 있도록 하기 위해 세부 규

격을 재정의한 CAP 프로파일을 제안하였다[5]. CAP은 XML(eXtensible Markup Language)형태로 되어 있어 사람이 읽을 수 있으며 개발자에 의해 필요에 따라 요소(Element) 및 속성(Attribute)을 추가/수정할 수 있는 유연성을 가지고 있다. 지하공동구의 다양한 객체들을 토대로 분석 및 시뮬레이션의 기반이 될 수 있는 지하공동구 3차원 데이터 모델을 제시하였다[6]. 지하공동구 3차원 데이터 모델은 컴퓨터 화면상에 가상화된 지하공동구를 표시하며 재난 발생시 재난 정보수신 후 가시화하고 발생 위치로 화면을 이동하여 재난 대응 및 조치에 도움을 줄 수 있다. 공동구에서 발생하는 화재는 케이블 자체에서 발생하는 경우와 외부 발화원에 의한 경우가 있다. 화재 특수성으로는 밀폐공간의 이유로 자연조건이 배제되어 불안심리를 유발, 이성적 판단 곤란, 시각감각을 저하, 활동장애 유발, 강박, 방향감각 상실등이 생길 수 있다고 언급하였다. 또한, 진·출입의 문제점, 동간 구획의 문제점, 배연상의 문제점, 소방시설상의 문제점도 지적하며 개선방향을 제시하였다[7].

정부는 데이터 기반의 상황관리시스템을 구축·운영하고 있지만 재난 상황 발생시 현장 상황 파악이 어렵다는 한계를 지적하며 재난안전 상황실 근무자의 상황관리 업무에 대해 지능화 및 효율화에 필요한 기술을 도출하였다[8]. 자연 재난 발생 빈도가 심각해지고 있고 화재·폭발등 사회 재난이 복잡적으로 일어나고 있다. 재난에 대한 도시의 취약성 및 위험성이 심각해지고 있고 물리적·지리적 환경이 열악한 곳에서 재난이 발생하기 쉽다. 사전 예방측면에서 재해에 대응하기 위한 체계적인 관리가 필요하고 재난유형별 요인을 도출해 취약요인을 비교·분석 하였다[9]. 사회재난 발생 빈도와 규모가 커짐에 따라 사회적·경제적 피해 규모도 커지고 있다. 재난이 대규모화되고 복잡성이 증가하기 때문에 인력으로 대응하는 것에는 한계가 있고, 인력이 위험에 노출되는 단점이 있다. 이를 해결하고자 재난 대응 로봇을 이용하여 재난 현장을 실시간으로 인지 및 대응할 수 있는 방안을 제시하였다[10].

관련된 시설물과 연계된 대량의 이기종 센서 데이터를 수집하였을 때, 고속으로 사용자에게 관제 서비스를 제공하기 위해 데이터를 정규화하는 방안을 제시하고, 이기종 데이터에 대한 동기화와 태깅을 통해 효율적인 데이터 고속 검색 서비스를 위한 방법을 설계하였다[11]. 기존의 예·경보시스템이 가지는 한계를 개선하는 IoT 기반 재난대응 조기경보시스템을 구축하고자 하였으며 각종 센서로부터 계속되는 데이터를 체계적이고 표준화된 구조로 데이터베이스화하여 대상지역에 현실적인 예·경보 기준 마련하는데 활용할 수 있도록 하였다[12]. 악성 댓글 분류를 위해 메시지큐 아키텍처를 이용하고 서비스가 예상치 못한 상황에서 종료되더라도 메시지큐에 대기 중이던 댓글 분석 요청은 유실되지 않고 개별 서비스들을 독립적으로 운영할 수 있음을 실험한 사례도 있다[13].

위에서 언급한 선행연구들을 분석하여 데이터베이스, 메시지 큐, Open API등 관련 기술을 선정하였으며 각 기술의 특

성을 활용해 기능을 구현하였다.

3. 지하공동구 화재탐지 프레임워크

3.1 화재탐지 프레임워크

고성능 실시간 지하공동구의 화재탐지를 위해 Fig. 1과 같은 메시지 큐를 활용한 프레임워크를 구현한다. 프레임워크는 복잡한 문제를 해결하는데 사용되는 기본 개념 구조로 일정한 요소와 틀, 규약을 가지고 무언가를 만들어 내는 일이다.

지하공동구 화재탐지 프레임워크는 이기종의 다양한 센서로부터 데이터를 수집하는 수집기, 수집된 데이터를 이용해 메시지를 생성하는 Producer, 비동기 실시간 메시지 송/수신을 위한 메시지 큐, 이벤트 메시지를 수신하여 재난을 탐지·판단하는 Consumer, 재난 발생시 해당 정보를 다른 시스템에 전파하기 위한 재난 전파 모듈, 데이터 연속성을 위한 데이터베이스, 데이터 조회 및 관리를 위해 Open API를 제공하기 위한 Server로 구성된다. 메시지 큐는 시스템 및 플랫폼에 독립적이고 비동기 처리가 가능한 구조이기 때문에 실시간 처리에 적합하다. 또한, 데이터의 연속성 및 시스템을 위한 메타정보 관리를 위해 데이터베이스도 같이 사용한다. 지하공동구내의 각종 센서로부터 생성된 데이터는 주기적으로 메시지 큐로 전달되며 Consumer는 실시간으로 메시지를 수신하여 처리한다. 메시지 큐와 주고 받는 메시지는 텍스트 형식으로 사람이 읽을 수 있고 기계도 처리할 수 있는 JSON형식을 이용한다. 추가적으로 Open API도 제공하여 필요시 해당 데이터를 조회하여 업무에 활용할 수 있도록 한다. 이 프레임워크는 사용자와의 상호작용이 필요없는 백엔드의 기능이며 항상 메모리에 상주하여 실행되어야 한다.

재난 정보를 가시화하는 방법중에 하나로 디지털 트윈 기술이 있다. 지하공동구 내부를 디지털 트윈으로 실제와 유사하게 모델링하고 3차원 객체는 Blender등의 툴을 이용해 제작하며 Unity 등을 통해 가시화 하면 손쉽게 위치 정보와 재난 관련 메타 정보를 표시할 수 있다. 수신된 데이터와 모델링된 데이터에는 좌표 정보가 포함되어 있어 컴퓨터 화면상에 재난이 발생한 위치를 정확하게 표시할 수 있다. 디지털 트윈을 위한 3차원 모델링은 연구 범위를 벗어나기 때문에 자세하게 설명하지는 않는다.

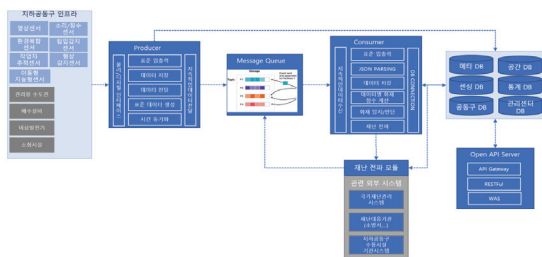


Fig. 1. Fire Detection Framework Conceptual Diagram

3.2 지하공동구

지하공동구는 좁고 긴 복도 형식의 구조물이기에 데이터의 전달, 위치 식별등의 불편함이 존재할 수 있고 각종 센서의 센싱 정확도등이 떨어질 수 있으며 관리의 불편함도 따른다. 이를 해결하기 위해 Fig. 2처럼 공동구 공간을 작은 단위의 논리적인 구분인 셀(Cell), 섹터(Sector), 박스(Box)로 나누어 처리한다. 셀(Cell) 개념은 200M 간격의 논리적 구분이며, 셀 안에는 섹터(Sector) 개념이 존재하며 50M 간격으로 구분한다. 한 셀안에는 4개의 섹터가 존재하며 섹터에는 저조도카메라, 열화상카메라가 있으며 RTSP(Real Time Streaming Protocol)를 이용하여 실시간으로 영상 정보를 전송하고 있다. 섹터안에는 가장 작은 단위인 박스(Box) 개념이 존재하며 10M 간격으로 구분한다. 한 섹터안에는 5개의 박스가 존재하며 박스 단위에는 1개의 조명과 1개의 환경센서가 있고 환경센서는 온도, 습도등 8종으로 구성되어 있다. 환경센서의 데이터는 약 2초 간격으로 서버로 전송된다. 그리고, 지하공동구의 기타 장비로는 레일로봇, 음향센서, 부분 방전 센서, 침수센서등이 존재한다. 레일로봇은 사용자와의 상호작용에 의해 동작하며 영상 카메라가 달려 있어 이동시의 영상을 사용자에게 실시간으로 전송할 수 있다. 음향센서는 비인가침입등을 판단할 때 사용되며, 스파크에 의한 화재 발생시에도 활용된다. 스파크 발생시 소리도 같이 발생하는 경향이 있다[4]. 부분 방전 센서는 전력선이나 통신선에서 발생하는 부분 방전을 감지한다. 공동구의 논리적인 구분을 간략하게 이미지로 표시하면 아래와 같다.

3.3 메시지 큐(Message Queue)

메시지 큐는 큐(queue)라는 자료구조를 채택해서 메시지를 전달하는 시스템이며, 메시지 지향 미들웨어(Message Oriented Middleware, 이하 MOM)을 구현한 시스템을 말한다. 김은미 외 10인(2024)은 클라우드에 재해가 발생했을 때 데이터 복구를 위한 DR(Disaster Recover)과정에서 이루어지는 메시지 전달을 효율적으로 관리하기 위해 메시지 큐 기반의 통신 방법을 제안하였다[15]. 서비스들이 직접적으로 통신하지 않고 매개체를 사용하여 통신하도록 설계된 구조는 각 서비스들에게 자율성을 보장한다.

큐는 자료구조중 하나로 가장 먼저 들어온 데이터가 가장 먼저 나가는(First in First out) 터널 형태의 자료구조이다. 비교되는 자료구조로는 스택(stack)이 있으며 가장 나중에 입력

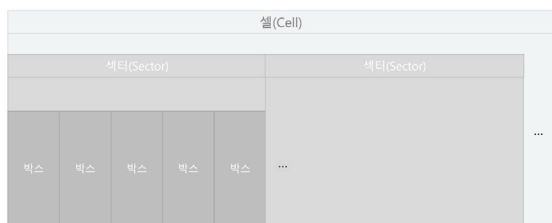


Fig. 2. Logical Demarcation of Underground Utility Tunnel

된 데이터가 가장 먼저 나가는(Last in First out) 형태이다. 메시지 지향 미들웨어는 서로 다른 응용 프로그램간에 메시지를 교환할 수 있게 해주는 소프트웨어 유형이다. 하드웨어나 소프트웨어 플랫폼의 종류에 관계없이 응용 프로그램이 서로 통신할 수 있는 공통 플랫폼을 제공한다. MOM은 일반적으로 메시지 대기열을 사용하여 응용 프로그램 간의 메시지를 저장하고 전송한다. 메시지를 발생하고 전달하는 부분을 Producer라고 하며, 메시지를 받아서 소비하는 부분을 Consumer라고 한다. 메시지 큐는 처리 대기중인 메시지의 임시 저장소이며 Producer와 Consumer의 메시지 전달 역할을 한다.

메시지 큐에서 데이터를 운반하는 방식에 따라 메시지 브로커(Message Broker)와 이벤트 브로커(Event Broker)로 나눌 수 있다. 메시지 브로커는 생산된 메시지를 메시지 큐에 저장하고, 저장된 메시지를 Consumer가 가져갈 수 있도록 관리한다. 메시지는 Consumer가 메시지를 가져간 후 짧은 시간동안만 메시지 큐에 유지된 후 삭제된다. 응용 프로그램으로는 RabbitMQ, ActiveMQ, Redis등이 있다. 이벤트 브로커는 메시지 브로커와 다르게 Consumer가 메시지를 소비한 이후에도 메시지가 삭제되지 않으며 필요할 때 다시 소비할 수 있다. 응용 프로그램으로는 Kafka등이 있다. 메시지 큐의 장점으로 비동기 처리, 낮은 결합도, 확장성등을 들 수 있다.

본 논문에서는 Kafka를 선정하여 사용하였다. Kafka는 Apache 재단에서 관리하는 고성능 데이터 파이프라인, 스트리밍 분석, 데이터 통합 및 미션 크리티컬 어플리케이션을 위한 오픈 소스 분산 이벤트 스트리밍 플랫폼이다. Kafka는 Broker, Topic, Partition, Producer, Consumer등으로 구성되어 있다. Broker는 Kafka 클러스터에서 가장 중요하며, 메시지를 저장하고 클러스터 전체에 데이터를 균일하게 분산하는 역할을 수행한다. Topic은 메시지 저장되는 논리적 단위이다. Partition은 토픽을 구성하는 단위로 파티션을 통해 데이터가 분산 저장되고 병렬처리 된다. Producer는 메시지를 Kafka로 전달하고 Consumer는 Topic을 읽어오는 역할을 한다. Kafka의 장점으로 고성능, 고가용성, 디스크 기반의 메시지 영속성을 들 수 있다.

3.4 Open API

외부 개발자들이 특정 서비스나 어플리케이션의 기능을 사용할 수 있도록 공개한 어플리케이션 프로그래밍 인터페이스(Application Programming Interface)이다[위키백과]. 이를 통해 개발자들은 특정 플랫폼의 데이터나 기능에 접근하여 개인화된 어플리케이션을 구현하거나 여러 기능을 하나로 통합할 수 있다. 일반적으로 프로토콜로 HTTP를 사용하며, 요청과 응답은 JSON이나 XML형식을 이용한다. XML은 W3C에서 개발된 다목적 마크업 언어이다[위키백과]. 주로 인터넷에 연결된 시스템끼리 데이터를 송수신할 수 있게 하여 HTML(HyperText Markup Language)의 한계를 극복할 목적으로 만들어졌다. JSON은 키-값 쌍으로 이루어진 데이터 객체를 전달하기 위해 인간이

읽을 수 있는 텍스트를 사용하는 개방형 표준 형식이다[위키백과]. 웹등에서 비동기 통신을 위해 폭넓게 사용하고 있으며 최근 현장에서는 가볍고 처리하기 쉬우며 bandwidth를 적게 사용하는 JSON을 XML보다 더 많이 사용하는 추세이다.

데이터베이스에 저장된 센싱 정보와 공간 정보등을 공개하기 위해 JSON을 이용하였다. REST 원리를 이용하여 데이터를 관리할 수 있도록 하였으며 WAS(Web Application Server)를 통해 웹으로 서비스된다. REST원리를 따르는 시스템은 종종 RESTful이란 용어로 지칭된다. REST원리는 인터페이스 일관성, 무상태, 캐시 처리 가능, 계층화등의 제한조건이 존재한다. URI(Uniform Resource Identifier)를 통해 리소스를 식별할 수 있어야 하며, HTTP method를 이용해 메시지 처리 방식을 전송한다. 등록/수정의 경우 HTTP body에 JSON형식의 메시지를 포함해 전송한다. 조회/삭제의 경우 URI를 통해 리소스를 식별할 수 있도록 하며 계층적으로 접근할 수 있도록 한다.

4. 지하공동구 화재탐지 시스템 구현

4.1 데이터베이스 구축

본 논문에서는 데이터베이스로 오픈소스 ORDBMS(Object-Relational Database Management System)인 PostgreSQL을 선정하였다. MySQL에 비해 SQL표준을 더 잘 지원하고 쿼리가 복잡할수록 성능이 더 잘나오는 편이다. 특히 확장기능인 PostGIS를 통해 GIS(Geospatial Information Service)와의 협업도 잘 이루어지는 편이다. 대용량 데이터를 다루기 위한 적합한 데이터베이스라고 할 수 있다. 공공기관 데이터베이스 표준화 지침[14]을 준용하여 테이블을 설계한다. 표준 단어·용어를 선정하여 사용하며 도메인에 맞는 데이터 타입을 선택한다. 각 테이블은 감사(Audit) 컬럼으로 등록자아이디, 등록일시, 수정자아이디, 수정일시 컬럼을 반드시 포함한다. 등록자아이디, 등록일시 컬럼은 필수 컬럼으로 지정하여 널값이 저장될 수 없도록 한다. 데이터가 데이터베이스에 등록될 때는 반드시 등록자아이디, 등록일시 컬럼에 값이 존재해야 한다. 테이블은 저장하는 데이터 성격에 따라 크게 3가지로 분류한다. 첫 번째는 시스템 운영을 위한 메타정보를 저장하는 테이블이고 두 번째는 실시간으로 수집되는 데이터를 저장하는 테이블이며 세 번째는 3차원 공간정보를 저장하는 테이블이다. PostGIS의 확장 기능을 통해 각종 GIS 함수 기능을 사용할 수 있다. 아래는 데이터베이스 테이블에 저장된 센싱 데이터 일부를 표시하고 있다.

4.2 가상 데이터 생성기(에뮬레이터)

실제 화재 환경을 구축하기에는 많은 어려움이 있기에 가상의 데이터를 생성해 재난 상황을 가정해 데이터를 발생시키는 역할을 한다. 데이터 생성 방법은 크게 2가지가 있다. 첫 번째 방법은 가상데이터를 파일이나 데이터베이스에 미리 저장해 놓은 후 필요시 읽어서 발생시키는 것이고 또 다른 하나

는 실시간으로 화재 데이터를 자동 생성하는 것이다. 각 데이터의 범위를 미리 지정한 후 임의의 데이터를 생성하는 방식이다. 화재시 생성되는 데이터는 시계열 성격을 띄고 있기 때문에 서서히 변경되는 특징을 가지고 있다. 임의로 생성되는 데이터의 값은 이점을 고려하여야 하며 갑자기 값이 오르락 내리락 하면 좋은 데이터가 될 수 없다. 가상 데이터 생성기는 각 데이터 종류에 맞게 주기적으로 데이터를 자동 생성하며 센서의 종류만큼 동시 다발적으로 데이터를 전송한다. 화재 발생을 가정할 때는 온도, 산소(O₂), 일산화탄소(CO), 이산화탄소(CO₂) 데이터를 발생시킨다. 산소는 시간이 흐를 수록 감소하는 형태로, 나머지는 증가하는 형태로 가상 데이터를 생산한다.

스레드를 이용하여 동시에 여러개의 센서에서 데이터를 전송하는 상황을 가정하였으며 센서별 데이터의 초기값은 정상 범위의 90%~100%에서 시작할 수 있도록 설정하였다. 각 값은 랜덤하게 동적으로 생성하며 앞에서 설명한 것처럼 화재 데이터는 시계열성을 띄고 있기에 시간이 지날 수록 초기값보다 값이 조금씩 더 점진적으로 커지도록 처리한다. 단, 화재 발생시 산소의 값은 줄어들어 시간이 지날수록 감소하도록 처리한다.

4.3 화재탐지

지하공동구에서 발생할 수 있는 재난은 화재, 비인가침입, 지진, 침수등이 있으며 여기서는 화재를 대상으로 진행한다. 재난은 크게 정상, 이상징후, 위험 단계로 구분하며 각 센서의 데이터를 융합하여 재난을 탐지하고 그 단계를 판단한다. 각 단계별 가중치 값을 다르게 설정하여 계산하며 최종적으로 모든 결과 값을 더한 후 총합으로 위험도를 판단한다. 예를 들면, 데이터의 값이 정상범위 일때는 0, 이상징후 범위 일때는 1, 위험 범위 일때는 2의 값을 가중하는 방식이다. 각 데이터별로 단계를 구분하는 값의 범위가 존재하며 그 값은 Table 1과 같다.

가상 데이터 생성기를 이용하여 데이터를 발생시키고 화재 탐지 프레임워크를 통해 이벤트를 수신, 연관 데이터 조회하여 재난 단계를 판단한다.

이벤트를 수신한 후 재난을 탐지하는 기능의 Consumer는 응용 프로그램으로 구현한다. 이 프로그램은 메모리 상주형으로 항상 메모리에서 실행되며, Kafka로부터 이벤트를 수신하기 위해 Kafka에 연결되어야 하며 무한 루프를 돌면서 이벤트 여부를 감지 한다. 오류등에 의해 Consumer 프로그램이 다운 상태가 되어도 Kafka에 저장된 메시지는 삭제되지 않

며, Consumer 프로그램이 다시 동작하게 되면 그 동안 처리하지 못했던 메시지를 받아 정상적인 처리를 하게 된다. 많은 프로그래밍 언어에서 Kafka를 사용할 수 있도록 라이브러리를 제공하고 있으며 본 실험에서는 Python을 이용하여 구현한다. 연관 데이터 위험 판단표는 자주 수정되는 정보가 아니기에 성능을 높이고자 cache 기능을 이용하여 한번 데이터 조회한 후 일정시간 동안 메모리에 저장해 두어 필요시 disk I/O없이 바로 사용할 수 있도록 하였다. 프로그램은 Kafka의 이벤트를 처리하기 위해 루프를 돌고 있으며 이벤트 수신시 바로 처리하고 다시 다음 이벤트를 수신하는 상태로 대기한다. Kafka는 비동기 처리가 가능하기에 Consumer가 다른 로직을 처리하고 있는 순간에 이벤트가 발생하더라도 Consumer의 상태와 관계없이 이벤트는 처리를 기다리지 않고 바로 Kafka 내부에 저장되며 Consumer가 다시 이벤트를 처리할 수 있는 상태가 되면 저장된 이벤트는 Consumer에 의해 처리된다. 해서, 신뢰성있고 비동기적으로 고속 처리가 가능하게 된다.

주요 프로그램 로직은 Fig. 3과 같으며 데이터 수신, 저장, 파싱, 탐지/판단, 전파의 프로그램 흐름을 반복하게 된다. 추가적으로 재난으로 탐지 및 판단된 결과는 데이터베이스에 저장되며 외부 시스템에 제공하기 위해 JSON형태의 데이터가 제공되는 Open API가 사용된다. 주요 로직으로는 데이터별 임계값 계산이 있다. JSON데이터를 파싱하여 온도 데이터를 읽어들인다. 수신된 온도 데이터를 화재 단계를 판단하는 값과 비교하여 측정된 값이 어느 범위에 있는지 식별한 후 가중치 값을 설정한다. 정상범위이면 0, 이상징후 범위이면 1, 위험범위이면 2를 설정하는 식이다. 같은 방법으로 산소, 일산화탄소, 이산화탄소도 계산한다. 모든 계산이 완료된 후 설정된 숫자를 모두 더해 합계를 구한다. 최종적으로 구한 숫자를 통해 화재 재난의 단계를 파악한다.

4.4 Open API

화재탐지 결과 및 부가적인 정보를 외부 시스템과 신속하고 정확하게 공유하기 위해 Open API를 활용한다. 외부시스템은 제안한 시스템의 내부를 알지 못해도 충분한 정보를 조회할 수 있다. REST(Representational State Transfer) 아키텍처 스타일을 따르는 API를 RESTful이라 하며 Http Method를 통해 수행해야 하는 작업을 서버에 전달한다. 대부분의 작업은 데이터를 조회하는 일로 GET Method를 사용한다. 데이터

Table 1. Threshold by Stages

	Unit	Normal	Warning	Danger
Temperature	℃	~50	50~70	70~
O ₂	%	20~	14.5~20	~14.5
CO	ppm	~9	9~34	34~
CO ₂	ppm	~1000	1000~2000	2000~

```

while(true) {
    // MQ로부터 메시지(데이터) 수신
    string message = readEventFromMQ()

    // 데이터를 DB에 저장
    saveToDb(message)

    // json 파싱
    object json = Json.parse(message)

    // 데이터별 임계값 계산 후 임계값 계산
    int 전체합계 = sum(calcLevel(온도값, 습도값, 산소값, 일산화탄소값, 이산화탄소값))

    // 화재 탐지시 재난 정보 전파
    if(isDisaster(전체합계))
        propagationDisaster()
}
    
```

Fig. 3. Key Logic for Fire Disaster Detection

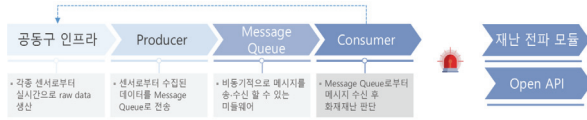


Fig. 4. Data Flow

등록시에는 POST, 수정시에는 PUT, 삭제시에는 DELETE Method를 사용하도록 한다. 승인된 시스템만 API를 호출할 수 있도록 하기 위해 Service Key를 이용한다. 승인된 시스템에 할당되는 임의의 긴 문자열로 API 요청시에는 이 값을 항상 포함하여 전달해야 한다. 리소스는 디렉터리 구조로 식별하도록 한다. 예를 들어, 이벤트 목록을 조회할 때는 GET /api/v1/eventes uri를 이용한다. 이 이벤트들중 1번 이벤트에 대한 정보를 조회할 때는 GET /api/v1/events/1 이라고 요청한다. 응답 Http Header에는 Content-Type : application/json을 추가하여 응답되는 데이터 형식이 JSON형식임을 알린다. 응답되는 데이터에는 이벤트 상세 정보, 공간 정보등이 포함된다. 이벤트 상세 정보에는 이벤트 아이디, 이벤트 종류, 발생일시, 센서정보, 센싱된 값등이 있으며 공간 정보에는 이벤트를 감지한 센서의 종류, 좌표 정보등이 있다. 지하공동구 내부를 디지털 트윈으로 3차원 모델링을 하였기에 공간 정보가 존재하며 좌표 정보에 의해 이벤트가 발생한 위치를 정확하게 알 수 있다. 디지털 트윈 기능을 활용한 UI 시스템에서 위치 정보를 활용할 수 있도록 하기 위해 공간 정보를 응답한다.

전체적인 데이터 흐름은 아래 그림과 같다. 공동구 인프라 단에 설치된 각종 센서에서의 데이터는 Producer로 전송되며 Producer는 데이터 수집/가공한 후 Message Queue로 전송하게 된다. Consumer는 Message Queue로부터 비동기적으로 데이터를 수신한 후 파싱/분석등의 과정을 거친 후 화재를 판단하게 된다. 화재가 아닌 평상시 상태이면 Message Queue로부터 데이터 수신하는 작업을 반복하게 되며 화재라 판단되는 경우 재난 전파 모듈로 데이터를 전송하게 된다. 재난 전파 모듈은 수신된 데이터와 필요한 다른 데이터들을 조합하여 재난 메시지를 생성하게 되고 다른 시스템에 전파하기 위해 Message Queue로 데이터를 전송하게 된다.

5. 결 론

본 논문은 지하공동구에서 실시간으로 생산되는 센싱 데이터를 수집하여 신뢰성있게 화재를 탐지할 수 있는 프레임워크 및 처리 절차(Pipeline), 그리고 신속하게 정보를 공유하기 위한 Open API 기술을 활용하여 시스템을 구현하였다. 수신되는 모든 센싱 데이터는 데이터베이스에 저장하며 평소 데이터와 다른 특이값 발생시 메시지 큐로 데이터를 전송하여 이벤트를 발생시켰다. Consumer는 JSON형태로 구성된 이벤트 메시지를 수신한 후 해당 데이터를 파싱하여 이벤트가 발생한 데이터 종류를 판단하였다. 데이터 종류가 특정되면 관련된 연관

데이터를 조회한 후 종합적으로 판단하여 재난 여부와 위험 정도를 판단하였다. 실제로 지하공동구상에 재난을 발생시킬 수 없기에 가상으로 재난 데이터를 발생시키는 프로그램을 구현하였고 이를 활용하여 실험을 진행하였다. PostgreSQL을 RDBMS로 선정하였으며 시스템을 운용하기 위한 메타데이터와 실시간으로 센싱되는 데이터를 저장하였다.

본 논문에서 제시한 모듈과 시스템을 실제 시스템에 적용하였으며 생존성과 유연성, 가변성, 성능향상의 결과를 확인하였다. 가상 데이터 생성기를 활용해 다양한 시나리오의 데이터를 생성하여 화재 재난을 테스트할 수 있었으며 Message Queue를 활용한 데이터 전파는 비동기적인 데이터 생산/소비의 특징을 활용해 신속하고 정확하게 목적지에 데이터가 도착하였다. 또한, Kafka의 특징인 데이터 저장 기능은 데이터 유실 방지에 큰 도움이 되어 시스템 및 데이터의 안정성을 향상시켰다. 사용자단의 가시화 모듈이 재기동된 후 Kafka에 소비되지 않은 데이터가 존재하면 Kafka는 해당 데이터를 가시화 모듈로 전송했으며 사용자는 해당 데이터 및 내용을 확인할 수 있었다.

실제 데이터와 가상 데이터 생성기를 통한 데이터의 차이는 구분할 필요가 있다. 실제 운영 환경에서는 항상 좋은 데이터만 전송되지 않을 가능성이 높다. 센싱 데이터의 경우 하드웨어의 오류나 네트워크의 장애등으로 인해 이상치의 데이터가 송·수신되거나 결측치가 발생할 수 있으나 가상 데이터의 경우 이상치 및 결측치는 가정하지 않고 데이터를 생산하였다. 가상 데이터 생성기가 현실세계의 화재 데이터를 모사하였으나 100% 완전 일치하지 않을 수 있음을 인지해야 한다.

결과적으로 가상 데이터 생성기를 구현하여 실제 재난 상황이 아니어도 시스템을 테스트할 수 있는 방법을 제시하였고 이기간 비동기적으로 데이터를 생산/소비할 수 있도록 하기 위해 Kafka를 도입하여 실시간성 데이터 처리 및 흐름 환경을 구축하였으며 Open API를 활용하여 외부 시스템에서도 수집된 데이터를 조회 및 관리할 수 있도록 하였다.

화재의 경우 심각한 피해를 초래할 수 있기에 조기에 식별하여 조치하는 것은 매우 중요하다. 전체 아키텍처는 각종 센서로부터 데이터를 수신하는 모듈과, 비동기적으로 메시지를 생산/소비할 수 있는 메시지 큐, 그리고 메시지를 수신하여 재난을 판단하고 전파하는 이벤트 처리 모듈로 구성하였다. 지하공동구내에 설치된 각종 센서의 데이터는 일정 시간 주기로 수집되어 데이터베이스에 저장된다. 임계값 이상의 데이터가 수신되는 경우 이벤트로 판단하여 메시지 큐로 정보를 전달한다. 이벤트 처리 모듈은 메시지를 수신한 후 데이터베이스에 저장된 연관 데이터를 조회하여 종합적으로 계산한 후 재난 여부 및 단계를 판단하였다.

실제 재난을 발생시킬 수 없기에 가상 데이터를 생성하여 사용했고 지하공동구는 출입 제한 구역으로 네트워크가 외부와 단절되어 있기에 제한된 환경에서 실제 운영 서버에 접속하여 실험을 진행하였다. 또한, 2가지 이상의 재난이 함께 발

생하는 복합 재난 상황의 실험이 이루어지지 못했다. 추 후에는 이점을 감안한 연구가 진행되어야 한다.

References

[1] M. S. Lee, W. S. Jung, and E. S. Kim, "A Study on the Disaster Safety Management Method of Underground Lifelines based on Digital Twin Technology", *Communications of the Korean Institute of Information Scientists and Engineers*, Vol.39, No.2, pp.16-24, 2021.

[2] Ministry of the Interior and Safety [Internet], <https://www.safekorea.go.kr>

[3] D. S. Han and K. T. Park, "Evaluation of Road Facilities Performance Improvement Projects under the Basic Act on Sustainable Infrastructure Management", *Korean Society of Civil Engineers*, Vol.2021, No.10, pp.130-131, 2021.

[4] B. J. Lee, C. W. Park, M. S. Lee, and W. S. Jung, "Development of a Acoustic Acquisition Prototype device and System Modules for Fire Detection in the Underground Utility Tunnel", *The Journal of The Institute of Internet, Broadcasting and Communication*, Vol.22, No.5, pp.7-15, 2022.

[5] J. E. Kim and W. S. Jung, "A Study on a Common Alerting Protocol Profile for Disaster Management of the Underground Utility Tunnel", *The Journal of Korean Institute of Communications and Information Science*, Vol.47, No.11, 2022.

[6] M. W. Jeong and D. B. Shin, "Developing a 3D Spatial Information Data Model for Ensuring Interoperability in Underground Utility Tunnels", *Journal of the Korea Contents Association*, Vol.23, No.12, pp.37-45, 2023.

[7] J. I. Lee, "A Study on the Fire Prevention Activities and Suppression Measures of Utility-Pipe Conduit", *Journal of the Korean Society of Hazard Mitigation*, Vol.10, No.4, pp.63-68, 2010.

[8] S. H. Choi, J. Y. Son, M. S. Kim, H. W. Yoon, S. H. Ryu, and S. H. Yoon, "Research on Making a Disaster Situation Management Intelligent Based on User Demand", *Korean Journal of Remote Sensing*, Vol.39, No.39, pp.811-825, 2023.

[9] J. Y. Kim, J. H. Park, B. H. Cho, and S. S. Lee, "A Comparative Analysis of Disaster Vulnerability Factors between Declining Areas and Unban Areas", *Journal of Digital Contents Society*, Vol.21, No.11, pp.2021-2032, 2020.

[10] S. S. Kim and D. Y. Shin, "Current Status of Technology Development and Policy Recommendations of Disaster

Robot for Inaccessible Disaster Site", *Journal of the Korea Academia-Industrial Cooperation Society*, Vol.22, No.11, pp.270-276, 2021.

[11] J. W. Park, J. H. Jeon, and G. M. Park, "Design of Data Generating for Fast Searching and Customized Service for Underground Utility Facilities", *Journal of Broadcast Engineering*, Vol.26, No.4, pp.390-397, 2021.

[12] J. I. Song, Y. C. Kim, J. S. Lee, and H. T. Kim, "Establishment of An Early Warning System for Disaster Response in Disaster-Prone Areas", *Journal of the Korea Institute for Structural Maintenance and Inspection*, Vol.26, No.2, pp.199, 2022.

[13] J. S. Park, H. K. Kim, T. H. Jung, E. S. Shin, and S. H. Bae, "Malicious comment classification on a microservice architecture using message queue", in *Proceedings of Korean Institute of Information Scientists and Engineers*, pp.1876-1878, 2021.

[14] Ministry of the Interior and Safety [Internet], https://www.mois.go.kr/frt/bbs/type001/commonSelectBoardArticle.do?bbsId=BBSMSTR_000000000016&nttId=99662

[15] U. M. Kim et al., "Utilizing Message Queue in Disaster Recovery Process: Efficient Management of Cloud Services", in *Proceedings of The Korea Contents Association*, pp.611-612, 2024.



현성길

<https://orcid.org/0009-0001-6054-9180>
 e-mail : gravity@vaiv.kr
 2001년 충북대학교 정보통신학과(학사)
 2020년~현 재 바이브컴퍼니
 수석 연구원
 2022년~현 재 홍익대학교
 산업융합협동과정 석사과정

관심분야 : Smart City, IoT



유동영

<https://orcid.org/0000-0002-8231-5203>
 e-mail : ydy@hongik.ac.kr
 2011년 고려대학교 컴퓨터학과(박사)
 2014년~2018년 숭실대학교
 정보과학대학원 겸임교수
 2000년~2019년 한국인터넷진흥원 단장

2021년~현 재 홍익대학교 소프트웨어융합과 부교수
 관심분야 : 정보보호, 융합보안, IoT, 블록체인 등