

<https://doi.org/10.7236/JIIBC.2023.23.1.163>

JIIBC 2023-1-24

## 상태 정의 및 진단 알고리즘 기반 제조설비 시멘틱 모델링에 대한 연구

### A Study on the Semantic Modeling of Manufacturing Facilities based on Status Definition and Diagnostic Algorithms

곽광진\*, 박정민\*\*

Kwang-Jin Kwak\*, Jeong-Min Park\*\*

**요약** 본 논문은 제조설비의 자율제어와 상태판별 알고리즘을 위한 시멘틱 모델링 기술에 대해 소개한다. 디지털 트윈 기술과 스마트 팩토리의 다양한 ICT 기술의 발전으로 제조업은 새로운 생산 관리모델이 구축되고 있다. 발전된 스마트 제조기술을 바탕으로 상태판별 알고리즘은 자율제어와 공장 내의 설비 문제를 빠르게 파악하고 대처하기 위한 방법론으로 제시되었다. 그러나 기존의 상태판별 알고리즘은 사용자 또는 관리자에게 그리드 맵을 통해 주요 정보를 알려주고, 이에 대처하는 방향으로 제시되었다. 하지만 스마트 제조기술의 고도화와 방향성은 유연 생산과 소비자 니즈에 맞춘 생산등으로 다변화 하고 있다. 이에 따라 본 논문에서는 시멘틱 기반의 Linked List 자료구조를 이용하여 공장을 설계 구축하고 그래프 기반 정보를 통해 사용자 또는 관리자에게 필요한 정보만을 제공하여 관리의 효율성을 높일 수 있는 기술을 소개한다. 이러한 방법론은 유연 생산과 다품종 소량 생산 등에 적합한 구조로 활용될 수 있다.

**Abstract** This paper introduces the semantic modeling technology for autonomous control of manufacturing facilities and status definition algorithm. With the development of digital twin technology and various ICT technologies of the smart factory, a new production management model is being built in the manufacturing industry. Based on the advanced smart manufacturing technology, the status determination algorithm was presented as a methodology to quickly identify and respond to problems with autonomous control and facilities in the factory. But the existing status determination algorithm informs the user or administrator of error information through the grid map and is presented as a model for coping with it. However, the advancement and direction of smart manufacturing technology is diversifying into flexible production and production tailored to consumer needs. Accordingly, in this paper, a technology that can design and build a factory using a semantic-based Linked List data structure and provide only necessary information to users or managers through graph-based information is introduced to improve management efficiency. This methodology can be used as a structure suitable for flexible production and small-volume production of various types.

**Key Words** : Status Definition Algorithm, Semantic Web, SPARQL, Smart Manufacturing

\*정회원, 한국공학대학교 자율제어연구실

\*\*정회원, 한국공학대학교 컴퓨터공학과

접수일자 2022년 10월 31일, 수정완료 2023년 1월 30일

게재확정일자 2023년 2월 3일

Received: 31 October, 2022 / Revised: 30 January, 2023 /

Accepted: 3 February, 2023

\*\*Corresponding Author: jmpark@tukorea.ac.kr

Dept of Computer Engineering, Tech University of Korea, Korea

## I. 서 론

4차 산업 혁명은 빅데이터, 인공지능, IoT, 센서, 로봇 등 다양한 기술을 통해 많은 발전을 이루었다. 특히, 지능형 공장 시스템에서는 빅데이터와 인공지능 등을 바탕으로 비약적인 발전을 이루었다<sup>[1,2]</sup>.

최근 IoT, 빅데이터, 인공지능 기술 등을 바탕으로 디지털 트윈 기술은 많은 발전을 가져왔다. 디지털 트윈을 구성하기 위한 정보의 수집, 처리, 및 예측 등에 맞추어 발전해 왔다. 디지털 트윈으로 현실세계의 제조 환경을 가상공간에 모사하고, 현실세계의 실시간 데이터를 가상 환경에 적용하여 제품의 설계, 생산, 유지 보수 등에 요구되는 다양한 시뮬레이션을 수행하여 효율적인 의사 결정을 지원하는 것이 가능해졌다<sup>[3]</sup>.

디지털 트윈을 구성하기 위해서는 다양한 센서와 IoT와 센서들로부터 데이터를 수집하고 이를 가상세계에 형상화 시키는 과정을 통해 구축한다. 형상화 시킨 가상세계에서 다양한 시뮬레이션을 통해 효율적인 생산 및 관리를 위한 의사결정을 지원한다<sup>[4]</sup>.

지능형 공장에서 자율제어 컴퓨팅은 핵심 목표 과제로서 자동화 시스템에 능동적으로 환경을 인지하고, 상황을 분석하고, 자동적으로 계획을 수립하고 이를 실행에 옮기는 프로세스이다.

지능형 공장의 자율제어 시스템은 모니터링, 분석, 계획, 실행의 4단계로 구성되어진다. 공장의 설비 들의 현재 상태를 실시간으로 감시하고, 각종 데이터를 통해 상황을 분석하고 최적의 해결책을 계획하고 이를 실행한다. 정보를 수집 분석하여 상황을 예측 및 계획 하고 이를 실행에 옮기는 과정이 디지털 트윈과 많이 흡사한 모습을 가지고 있다<sup>[5]</sup>.

그러나 자율제어 컴퓨팅과 디지털 트윈은 현실세계를 정확하게 묘사하기 위해 어려운 부분을 가지고 있다. 객체들에 대한 다양한 상황 및 위치 순서 등을 표현하고 구성하는데 너무 많은 정보가 필요하며 이는 사용자의 요구사항과 맞지 않을 수 있다. 또한, 복합적인 부분의 데이터 통합과 연관성에 대한 정의에 어려움이 있다.

이러한 단점을 해결하기 위해 최근 시멘틱 구조에 대한 연구가 활발하게 이루어지고 있다. 시멘틱 구조의 장점으로는 각 관계의 정의를 링크드 리스트의 관계로 정의하여 복잡한 구조를 추상적으로 설명할 수 있으며, 단순한 구조들의 관계를 확장시켜 복잡한 구조를 완성할 수 있다. 실제 공장의 구조 관계는 매우 복잡한 모양으로 구성되어져 있으므로 복잡한 지능형 공장의 구조를 구축

하거나 복잡한 구조를 설명하기에 적합하며, 부분적인 구조를 설명하기에 매우 유리하다<sup>[6]</sup>.

이러한 시멘틱 구조의 장점은 디지털 트윈 및 자율제어 시스템을 표현하기에 적합한 장점을 가지고 있음으로 최근 연구가 활발하게 이루어진다<sup>[7]</sup>.

본 논문에서는 시멘틱 구조를 이용하여 기존의 상태 정의 및 진단 알고리즘을 개선할 방법에 대하여 논의한다. 본 논문의 구성은 연구의 배경이 된 기술과 필요성을 본장에서 소개하였으며, 2장에서는 시멘틱 기술을 적용하여 개선할 대상과 적용 기술을 설명한다. 3장에서는 본 논문에서 제안하는 구체적인 아키텍처에 대해서 설명하며, 4장에서는 제안하는 알고리즘을 상세히 설명한다. 5장에서는 본 논문의 활용 방안과 개선해야할 방향에 대하여 논의한다.

## II. 관련연구

### 1. 상태 정의 및 진단 알고리즘<sup>[8]</sup>

제조 공정은 여러 설비들의 협업으로 이루어지며, 각 설비 들은 유기적인 관계를 가지고 있다. 따라서 하나의 설비에서 문제가 발생하게 되면 연쇄적으로 다른 설비에도 문제가 발생하게 된다. 만약 하나의 설비에 문제가 생겨서 공정이 중단된 경우 다른 공정에 영향을 미치게 될 때 이에 영향을 받는 공정을 판단하고 해당 공정들도 조치가 필요한 상황이 발생할 수 있다.

실제 공정은 공정간에 유기적인 관계를 가지고 있지만, 관리자의 입장에서 복잡한 관계를 추상적인 개념으로 표현할 수 있다면 문제의 상황을 인식하고 이에 대한 대처 범위를 정하고 판단한다면 문제에 대한 대처가 좀 더 원활하게 이루어 질 수 있을 것이다. 따라서 공정의 상태를 정의하고 공정 상태를 진단함으로써 이러한 공정 내의 상태를 판별하여 효율적인 생산 모델을 제시하였다.

먼저 공정의 상태를 정의하는 과정이 필요하며 이러한 공정을 크게 Ready, Working, Finish, Failure의 4단계로 정의한다.

Ready 상태는 설비가 대기 중인 상태를 나타내며, 공정을 가동하기 위해 부품 및 재료를 공급받으며 대기하고 있거나 작업지시가 내려지지 않은 상태를 나타낸다. 부품 및 재료를 공급 받고 작업지시가 내려지게 되면 Working 상태로 전환된다.

Working 상태는 설비가 공정을 작업을 수행함을 뜻한다. 공정이 완료되면 Finish 상태로 전환된다.

Finish 상태는 설비가 공정을 완료하면 Finish 상태가 되며, Finish 상태에서는 연관되어지는 설비와 공정 의 상태를 확인하고 수행된 작업결과물을 다음 공정으로 보내게 된다.

Failure 상태는 설비에 오류 또는 이상이 발생할 때 Failure 상태로 전환된다. Failure 상태의 원인은 두 가지 경우가 될 수 있다. 설비의 이상과 부품 및 재료의 공급 문제의 경우이다. 설비 자체의 결함을 Internal Failure, 부품 및 재료의 공급 문제를 External Failure 라 정의하면 문제의 원인이 공정내 설비의 결함인지 해당 공정 외적인 문제인지를 쉽게 파악할 수 있다.

공장 내 전체 공정을 모니터링하기 위한 방법으로 테이블을 사용하여 공정내의 문제를 해결하는 방법이 제안되었다.

	1	2	3	4	5	6	7	8	9
1		1							
2			1						
3									1
4									
5									1
6									1
7									1
8							1		
9				1					1

그림 1. 상태 정의 및 진단 알고리즘의 통합 공정 맵<sup>[8]</sup>  
 Fig. 1. Integrated process map.

그림 1에서 보는 바와 같이 공정 수에 따른 정방향렬을 구성하고 부품 별로 이를 마킹하는 방법으로 공정 메트릭스를 구성한다. 부품 1은 공정 1 → 2 → 3 → 9 순으로 진행됨을 보여주며, 부품 2는 공정 8 → 7 → 9 순으로 진행됨을 보여주며, 부품 3, 4는 5, 6번 공정에서 9번 공정으로 이동함을 보여주고 있다. 설비상태를 항등행렬에 추가하면 Internal Failure와 External Failure를 모두 표현할 수 있다. 오류 또는 문제가 발생되어 Failure가 발생하게 되면 1을 -1로 바꾸어 표현함으로써 이에 대한 표현이 가능하며, Working 상태가 아닌 상황에 대하여 0을 표현함으로써 Ready 상태임을 판별할 수

있다.

## 2. Semantic Web

기존의 웹은 정보를 제공하는 것을 목적으로 가지고 있었으며, 이는 하나의 문서화된 웹 구조였다. 그러나 시멘틱 웹은 기존의 웹을 확장하여 의미론적 웹을 구성하였다.

시멘틱 웹의 특징으로는 웹에서 제공되는 수많은 정보와 자원 간의 관계 의미 정보를 기계나 컴퓨터가 이해하고 처리할 수 있는 형태로 표현한다. 이로 인해 단순한 정보를 프레임워크에 전달할 수 있어 산업, 소셜네트워크, 의료, 재해, 국가정보시스템 등 다양한 분야에서 상호운용성을 확보하기 위해 활용되고 있다<sup>9,10</sup>.

시멘틱 기술의 특징으로는 웹 상의 리소스는 IRI(International Resource Identifier)를 통해 식별되고, RDF(Resource Description Framework)와 RDF Schema라는 데이터 모델을 통해 정의된다[W3C 스펙 <https://www.w3.org/RDF/>]. 또한 OWL(Web Ontology Language)과 같은 변형된 형태의 모델로 정의되기도 한다<sup>11</sup>.

이러한 언어들의 특징으로는 주어(Subject), 술어(Predicate), 목적어(Object)의 3단 구조를 이루고 있으며, 이를 통해 객체와 객체간의 관계를 정의하고 있다. 또한 구조 자체가 Linked List 형태로 이루어져 그래프나 트리 등의 복잡한 구조에 적용이 가능하다.

또한, 이러한 언어를 처리하기 위한 엔진으로 SPARQL (Semantic Protocol and RDF Query Language)라는 엔진을 사용하여 데이터베이스의 SQL과 비슷하게 데이터를 검색할 수 있다<sup>12</sup>.

상태 판별 테이블은 직관적인 가시성을 제공하지만 몇 가지 단점을 가지고 있다. 공정의 수가 늘어날수록 테이블의 크기가 커지게 됨으로 가시성이 낮아질 수 있다는 단점이 있으며, 테이블을 저장하거나 데이터를 메모리에 상주 시킬 때, 불필요한 저장공간이 더 필요하게 된다. 공정 내에서 처리하는 부품 및 재료가 복잡하게 될수록 이에 대한 정확한 처리가 어려워질 수 있다.

## III. 시멘틱 상태 판별 알고리즘

다음 그림 2는 시멘틱 상태 판별 알고리즘을 구성하는 과정을 설명한다.

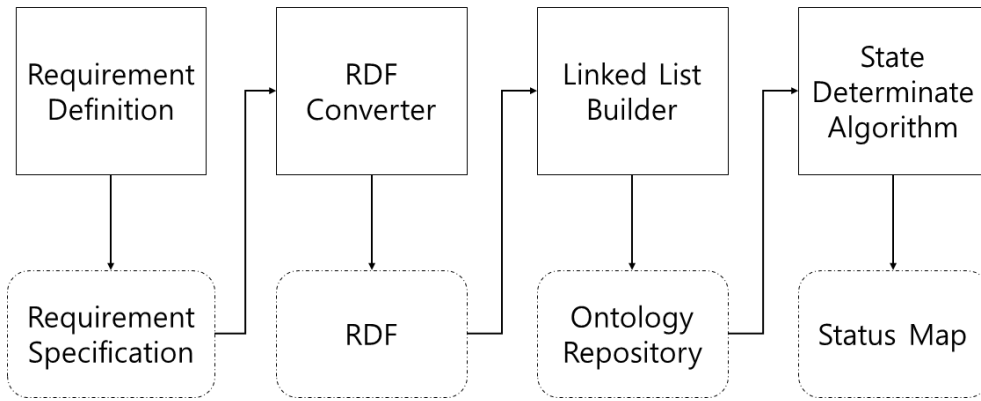


그림 2. 시멘틱 상태 판별 알고리즘 프로세스  
Fig. 2. Process of Semantic based Status Definition and Diagnostic Algorithm.

상태 판별 알고리즘을 시멘틱화 하기 위해서는 공정간의 명제를 명확하게 하는 것이 선행되어야 한다. 공정에서 어떠한 생산물을 만들기 위해서 필요한 부품과 재료를 명시하여야 하며, 다음 공정과 부품과 재료의 투입 경로가 명시되어야 한다. 이를 통한 산출 결과물인 요구사항 명세서를 토대로 RDF를 변환 후 객체와 리소스들을 Linked List 형태로 변환한다. 결과적으로 그래프 형태의 구조로 변환된 온톨로지 저장소가 구성되어지게 된다. 이를 상태판별 알고리즘을 통해 재구성한 상태맵을 보여주게 된다. 상태맵은 기존의 테이블 형태가 아닌 그래프 형태의 시각화로 보여주게 된다.

### 1. 요구사항 정의

요구사항 정의는 공장 내에서 수행되는 작업에 대하여 수집 및 정의되어야 할 내용을 기술하는 작업이다.

예를 들어, “공정 1은 모터의 캡을 만들기 위해 재료를 제공하는 공급기이고, 공정 2는 캡의 내경을 갈아내는 작업을 한다.” 라는 명제가 있다면 다음과 그림 3과 같은 3단 구조 형태의 명세서를 만들어 낼 수 있다.

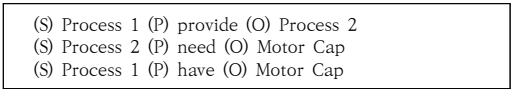


그림 3. 3단 구조의 명세서  
Fig. 3. Specification of 3-tier structure.

### 2. RDF 변환

그림 3에서 보는 바와 같이 3단 구조의 명세서를 작성한 후 이를 RDF 로 변환하면 다음 그림 4와 같이 표현된다.

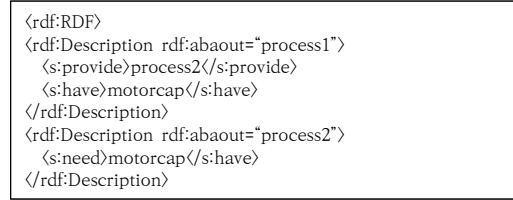


그림 4. 명세서를 RDF로 컨버전  
Fig. 4. Convert specification to RDF.

### 3. Ontology 명료화

그림 4의 rdf와 같은 xml 형태의 데이터들이 축적이 되면 축적된 데이터들을 효율적으로 관리하기 위하여 체계화, 정형화 시키기 위한 과정이 필요하다.

각 개념들간의 상호관계를 설정하고 개념을 명확하고 명시적으로 규정하기 위한 작업을 온톨로지라고 한다.

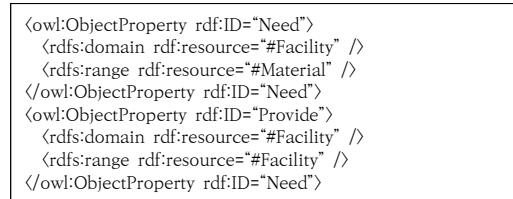


그림 5. RDF를 OWL로 명료화  
Fig. 5. Refining RDF to OWL.

그림 5에서 보는 바와 같이 Need와 Provide Property가 각기 주어(Subject)로 Facility의 범주안에 있으며, 목적어(Object)로는 Material과 Facility의 범주안에 있다는 것을 명시적으로 규칙을 둔 것을 표현한다.

## IV. 시멘틱 기반 상태 판별 알고리즘

### 1. 그래프 제작 알고리즘

시멘틱 기반 상태 판별 알고리즘은 기존의 테이블 방식으로 그려진 것과 다소 다른 방식으로 표현되게 된다.

기존의 상태 판별 알고리즘은 정방 행렬을 생성하여 정방 행렬 내에서 공정의 순서와 설비의 상태를 판별하였으나, 시멘틱 기반 상태 판별 알고리즘은 다소 복잡한 그래프 방식으로 그려지게 된다.

먼저 그래프의 가장 끝에 위치할 객체 노드들을 선별하여야 하는데 이 조건에 충족되는 객체 노드들은 need를 가지고 있지 않아야 한다.

이를 관계 대수식과 알고리즘을 표현하면 다음과 같다.

$$\pi_{\text{facility}} \sigma_{0 < \text{count}(\text{need}) \text{process}} \quad (1)$$

---

#### Algorithm 1 Make EdgeNodes

---

```

Input : Nothing
for all facilitylist do
  if facility.predicate not exist "need" then
    Draw facility
  end if
end for
    
```

---

#### 그림 6. 엣지 노드를 찾는 알고리즘

Fig. 6. Algorithm to find edge nodes.

수식 (1)과 그림 6에서 보는 바와 같이 predicate로 need를 가지지 않은 객체 노드들을 추려내고 이를 그려질 부분의 가장자리에 위치시켜 놓는다.

그림 6에서 보는 바와 같이 객체 노드들을 외곽자리에 배치한 후 배치된 노드들에서 need 속성을 가진 process들을 찾아내어 연결하는 작업을 반복한다. 그림 7은 엣지 노드들에서 시작하여 need를 포함하는 노드들을 재귀적으로 호출하여 Linked List들을 연결한 알고리즘이다.

---

#### Algorithm 2 LinkedEdgeNodes

---

```

Input : EdgeNodeList
for all facilitylist do
  if facility.predicate.need == EdgeNode then
    Draw facility
    Draw Arrow(EdgeNode, facility)
    EdgeNode=facility
    LinkedEdgeNodes
  end if
end for
    
```

---

그림 7. 엣지 노드에서 Linked List Node들을 연결하는 알고리즘  
 Fig. 7. Algorithm that connects Linked List Nodes at Edge Nodes

그림 7과 같이 재귀적으로 엣지 노드에서 추가적으로 그림을 Linked List들을 추가해 나가면 다음 그림 8과 같은 순서로 전체 맵을 그래프로 표현되어진다.

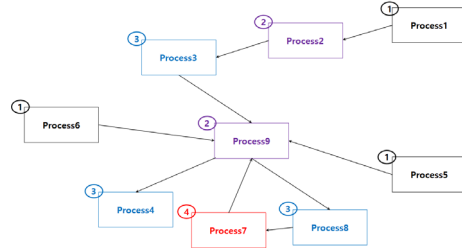


그림 8. 그래프 제작 알고리즘에 따라 그려진 그래프  
 Fig. 8. Graph drawn according to the graph making algorithm

그림 8의 위와 같이 각 노드들을 맵의 가장자리에 배치한 후 파생되는 노드들을 재귀적으로 연결하면 전체적인 그래프 맵을 만들 수 있다.

알고리즘과 그림으로 복잡하게 설명하였지만 실제 시멘틱 모델로 구현하면 간단한 SPARQL로 쉽게 구현이 가능하다.

```

select DISTINCT ?facility
where {
  ?o facility
  FILTER NOT EXISTS { ?p "need" }
}
    
```

그림 9. Sparql을 이용하여 Object Property로 need를 가지고 있지 않은 퍼실리티 검색

Fig. 9. Using Sparql to search for facilities that do not have a need with Object Property

그림에서 보는 바와 같이 need가 술어부에 없는 rdf 중 목적어로 facility를 가지고 있는 중복이 없는 facility를 찾는다.

### 2. 상태판별 알고리즘

제작된 그래프를 바탕으로 Internal Failure로 인해 발생한 External Failure의 범위를 쉽게 파악할 수 있다.

자체 설비의 오류로 생성된 Internal Failure를 기점으로 다른 공정이 멈추게 되었을 때 이는 연계된 설비에서 오류 발생임으로 그래프로 나타내어진 인접한 링크드 리스트에서 쉽게 확인이 가능하다.

기존 상태판별 알고리즘은 전체 그리드 맵을 보여줌으로 전반적인 상태를 표현하였다면 본 논문에서 설명하는 상태판별 알고리즘은 문제가 생긴 공정만을 쉽게 표시할 수 있다.

다음 그림 10은 상태판별 알고리즘의 생성과정을 보여준다.

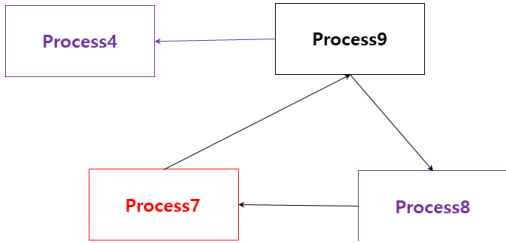


그림 10. Process7이 Internal Failure가 되면 영향을 받는 Process4, 8은 External Failure가 됨  
Fig. 10. When Process7 becomes an Internal Failure, the affected Process4, 8 becomes an External Failure.

그림 10에서 보는 바와 같이 설비 7에서 Internal Failure가 발생하게 되면 설비 7에게만 부자재를 주어야 하는 설비 8은 External Failure 상태가 된다. 그러나 다른 설비와도 연계가 많은 설비 9는 External Failure 상태일 수도 있고 아닐 수도 있다. 단일 제품을 생산하기 위해 전체 공정이 움직인다면 External Failure가 되지만 잉여 생산물을 재고로 한다고 하면 다른 설비의 생산과 연계해서 움직이고 있으므로 Failure가 되지 않는다. 그러나 그러나 최종 결과로 제품이 나오는 설비 4는 External Failure가 될 수 있다.

따라서 그림 11과 같은 전체 설비 오류 맵을 볼 수 있다.

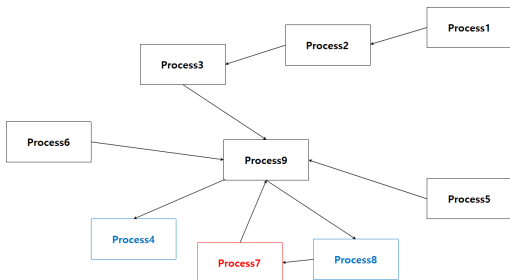


그림 11. Process7이 Internal Failure가 되었을때 전체 공정 맵  
Fig. 11. The entire process map when Process7 becomes Internal Failure.

기존의 상태판별 알고리즘은 단일 제품의 매뉴얼 순서대로 그림을 보여줬다면 본 논문에서 제안하는 상태판별 알고리즘은 그림 11에서 보는 바와 같이 연관된 설비만이 External Failure에 대하여 고찰해 볼 필요성이 있다. 설비의 기능이 다른 설비로 인해 완전히 멈추게 되었다면 External Failure이지만 지속적인 생산을 할 수 있다면 정상적으로 동작을 하고 있다. 또한 그림 11과 같이 문제가 발생할 시 전체 공정을 관리자에게 보여주는 것보다 문제가 되고 있는 공정만을 보여주는 것이 관리자에게 직관성을 제공할 것이다.

#### IV. 실험 및 성능 평가

MTConnect 기반의 시뮬레이터를 통해 성능을 평가하였다. 평가 항목은 기존 히트맵 상태진단 기반의 알고리즘과 본 연구에서 제안한 Sementic Web 방식의 데이터 생성과 메모리 점유에 대해 비교하였다.

그림 12에서 보는 바와 같이 초기 저장에 소모되는 저장소는 Sementic 상태진단이 미세하게 높다. 그러나 Raw Data가 생성됨에 따라 Sementic 상태진단 기법이 약 25% 증가를 낮음을 알 수 있다.

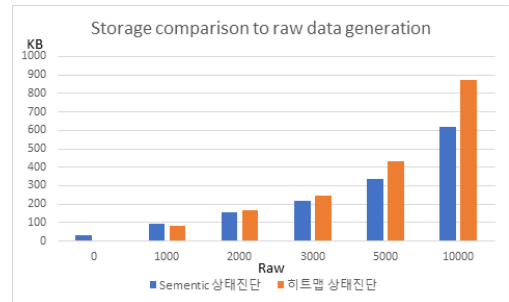


그림 12. Raw Data 증가에 따른 저장소 비교  
Fig. 12. Storage comparison to raw data generation

그림 13에서 보는 바와 같이 시뮬레이터 구동에서 작동하는 과정까지 Sementic 상태진단이 오히려 비효율적임을 보였다. 그러나 본 논문에서 제안하는 알고리즘은 레거시 장비 또는 임베디드 장비가 아닌 상위단에 PC에서 구동됨으로 큰 문제가 되지 않을 것이라 생각되며, 효율적인 최적화를 위해 향후 과제로 생각하여야 할 문제이다. 또한, MTConnect의 메모리 저장구조가 큐형태의 자료구조를 가지고 있으며, 휘발성을 기본으로 하여 일정 데이터 생성 이후에는 변동성 없이 구동되는 것을 확인하였다.

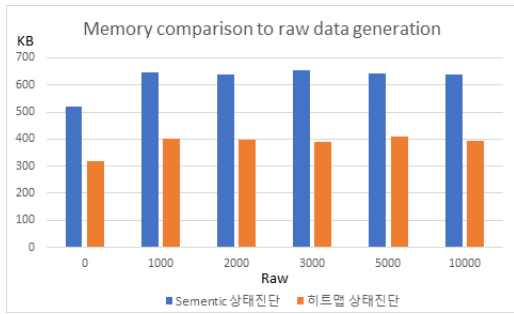


그림 13. Raw Data 증가에 따른 메모리 비교  
 Fig. 13. Memory comparison to raw data generation

## V. 결 론

본 논문에서는 기존에 연구하였던 상태판별 알고리즘을 개선하기 위하여 그래프 모델을 이용한 새로운 방법론을 제시하였다.

기존의 상태판별 알고리즘은 크게 두 가지 문제점을 가지고 있다. 첫 번째로 기존의 상태판별 알고리즘의 설비는 One-Hot 인코딩과 유사한 맵을 나타내는 방식이며, 즉, 영향을 주는 장비를 표시하기 위해 전체 설비수의 제곱의 저장공간을 필요로 하다. 두 번째로 전체 공정을 표현하고 있음으로 사용자에게 필요한 부분 외적인 부분 까지 보여주므로 가독성을 저해한다.

그러나 본 논문에서 제안하는 시멘틱 기반의 그래프 방식은 저장공간의 문제를 Key-Value 방식으로 개선하기에 적합하다. 또한 문제가 일어난 부분에 집중하여 설비 관계를 파악할 수 있는 유용한 방법론을 제시하였다.

또한, 이러한 구조는 생산성 향상과 다양한 모델을 생산하여야 하는 유연 생산 구조에서 활용도가 높을 것으로 전망된다.

그러나, 우리나라의 현재 스마트 팩토리 고도화는 높은 수준에 이르지 못하였으며, 시멘틱 모델의 연구는 도입되지 못하고 있다. 연구가 활성화 되고 이에 따른 기술력이 높아진다면 본 논문에서 제안한 그래프 기반의 시멘틱 모델 상태판별 기술은 효용성을 입증될 것으로 기대된다.

## References

[1] Jae-Wan. Yang, Young-Doo Lee, & In-Soo. Koo,

"Timely Sensor Fault Detection Scheme based on Deep," The Journal of The Institute of Internet, Broadcasting and Communication, Vol. 20, No. 1, pp. 163-169, 2020.

DOI: <https://doi.org/10.7236/IIBC.2020.20.1.163>

[2] In-Su, Lee, "A Study on Geospatial Information Role in Digital Twin," Journal of the Korea Academia-Industrial cooperation Society(JKAIS), Vol. 22, No. 3, pp. 268-278, 2021.

DOI: <https://doi.org/10.5762/KAIS.2021.22.3.268>

[3] Padovano, A., Longo, F., Nicoletti, L., & Mirabelli, G., "A Digital Twin based Service Oriented Application for a 4.0 Knowledge Navigation in the Smart Factory." IFAC-PapersOnLine, Vol.51, No.11, pp. 631-636. 2018.

DOI: <https://doi.org/10.1016/j.ifacol.2018.08.389>

[4] Kunath, M., & Winkler, H., "Integrating the Digital Twin of the Manufacturing System into a Decision Support System for Improving the Order Management Process." Procedia Cirp, No. 72. pp. 225-23, 2018.

DOI: <https://doi.org/10.1016/j.procir.2018.03.192>

[5] Ganek, A. G., & Corbi, T. A., "The Dawning of the Autonomic Computing Era." IBM systems Journal, Vol. 42, No. 1, pp. 5-18, 2003.

DOI: <https://doi.org/10.1147/sj.421.0005>

[6] Li, X., He, B., Wang, Z., Zhou, Y., Li, G., & Jiang, R. "Semantic-Enhanced Digital Twin System for Robot-Environment Interaction Monitoring." IEEE Transactions on Instrumentation and Measurement, Vol. 70, pp. 1-13, 2021.

DOI: <https://doi.org/10.1109/TIM.2021.3066542>

[7] Sung-Hyun, Nam, Yong-Ju Lee, "Developing a Semantic-based IoT Mashup Platform," The Journal of KIIT, Vol. 18, No. 6, pp. 117-127, 2020.

DOI : <https://doi.org/10.14801/jkiit.2020.18.6.117>

[8] Dong-Bum. Ko, & Jeong-Min. Park, "A Study on Status Definition and Diagnostic Algorithm for Autonomic Control of Manufacturing Facilities," The Journal of The Institute of Internet, Broadcasting and Communication, Vol. 20, No. 2, pp. 227-234, 2020.

DOI: <https://doi.org/10.7236/IIBC.2020.20.2.227>

[9] Kwang-Jin. Kwak, & Jeong-Min. Park, "A Study on Semantic based Autonomous Computing Technology for Highly Reliable Smart Factory in Industry 4.0.," Applied Sciences, Vol. 11, No. 21, 10121, 2021.

DOI: <https://doi.org/10.3390/app112110121>

[10] Kyong-Hyun. Park, Yong-Jae. Lim, Dae-Sub. Yoon, & Yang-Koo. Lee, "Unified Information Data Model for Semantic Interoperability of Digital Twins," Journal of Digital Contents Society, Vol. 22, No. 5, pp. 823-829, 2021.

DOI: <https://doi.org/10.9728/dcs.2021.22.5.823>

[11] OWL, <https://www.w3.org/OWL>

[12] SPARQL, [https://www.w3.org/2009/sparql/wiki/Main\\_Page](https://www.w3.org/2009/sparql/wiki/Main_Page)



## 저 자 소 개

### 곽 광 진(정회원)



- 2021년 ~ 현재 : 한국공학대학교 자율 제어연구실 박사 후 과정
- 2021년 : 한국공학대학교 스마트팩토리융합과 박사
- 2016년 : 건국대학교 컴퓨터공학과 석사
- 2008년 : 건국대학교 컴퓨터공학과 학사
- 주관심분야 : CPS, Autonomic Computing, IoT, Smart Factory

### 박 정 민(정회원)



- 2014년 ~ 현재 : 한국공학대학교 컴퓨터공학과 부교수
- 2012년 ~ 2014년 : 한국전자통신연구원 선임연구원
- 2008년 ~ 2011년 : 동양미래대 조교수
- 2009년 : 성균관대학교 컴퓨터공학과 박사
- 주관심분야 : CPS, Autonomic Computing, Software Engineering

※ 이 논문은 2022년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 기본연구지원사업임 (NRF-2021R1F1A1063634).  
※ 본 연구는 과학기술정보통신부 및 정보통신기술진흥센터의 대학ICT연구센터지원사업의 연구결과로 수행되었음 (IITP-2023-2018-0-01426).