



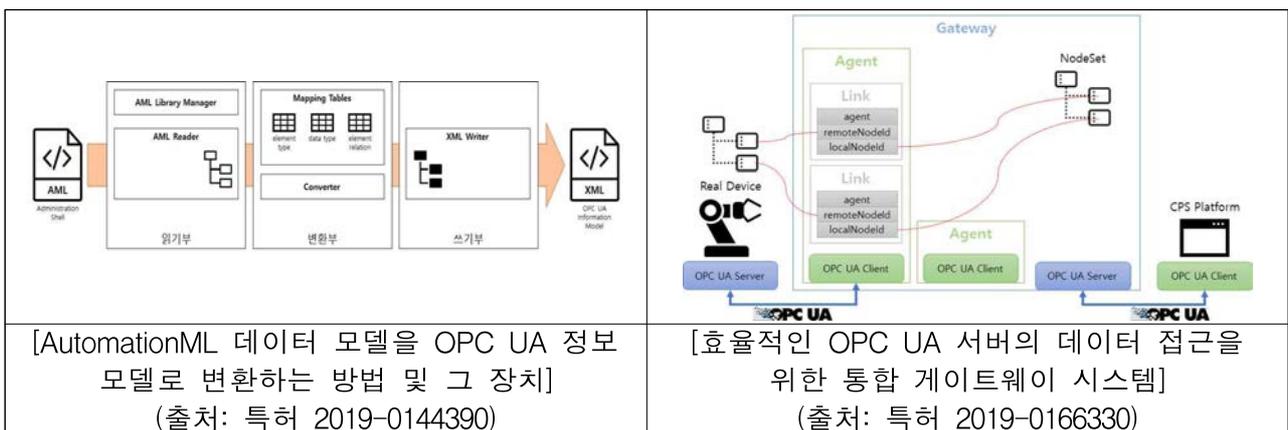
■ 기술명 : 국제 표준 기반 CPS(Cyber Physical System) 구현 기술 (Utilizing CPS Technologies based on the industry standards)

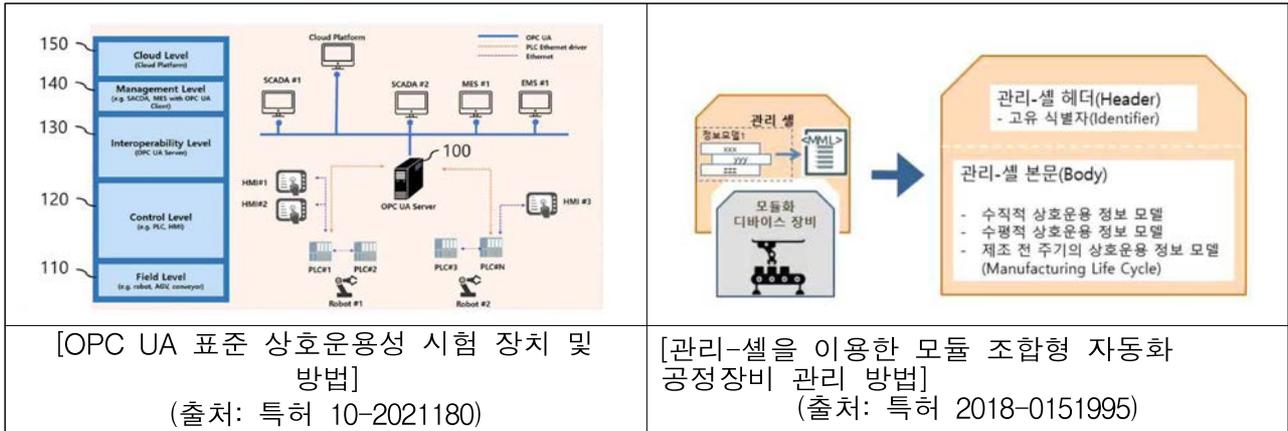
산업기술분류	정보통신 / 소프트웨어 / SW솔루션
Key-word(국문)	CPS, 산업 표준, 관리-셸, OPC UA, AutomationML
Key-word(영문)	CPS, Industry Standard, Administration Shell, OPC UA, AutomationML

■ 기술의 개요

- (배경) 현재의 생산 환경에서는 공장별로 더 나아가 공정 라인 별로 서로 다른 시스템을 사용하고 있으며, 시스템 별로 상이한 데이터 포맷과 데이터 전송 프로토콜을 채택하여 사용하고 있어 전체 시스템의 유연한 변경에 큰 장애 요소. 따라서 최근 다양한 생산 시스템 간의 상호호환성을 높이고 연결을 가능하도록 하는 국제 표준 기술의 적극적인 도입 및 활용이 필요
- (개요) 전자 제조공장의 모듈형 장비, 자산간 데이터와 서비스, 인터페이스의 수직·수평적 상호운용성을 보장할 수 있는 국제 표준기반의 관리-셸 참조 모델, 생산 시스템의 각종 정보를 기술하는 AutomationML 데이터모델을 OPC UA 정보 모델로 자동 변환하는 시스템, OPC UA 통합 게이트웨이 포함

< 기술 개요도 >





■ 기술의 구현수준(TRL) → 4단계로 수정



■ 기술의 장점(경쟁기술과의 차별성)

- 관리-셀 표준 기반의 참조 모델
 - 다양한 제조 공정과 관련된 자동화 장비들의 구성 요소와 이들의 수명주기 안에서 연속적으로 저장되어야 하는 중요한 산업 데이터를 위해 규격화 방안 제안
- AutomationML 데이터 모델 기반의 OPC UA 정보 모델 변환 시스템
 - 수동의 AutomationML 데이터 모델의 분석 및 OPC UA 정보 모델 변환에 대한 로직화 지원
 - AutomationML 스키마에 기반하여 데이터 모델 자료구조를 생성함으로써, AutomationML 표준의 수정 및 버전 업그레이드 용이
- 효율적인 OPC UA 서버의 데이터 접근을 위한 통합 게이트웨이 시스템
 - 많은 수의 제조 디바이스 및 장치가 동시에 클라이언트 어플리케이션과 연결되어도 안정적인 서비스 제공
 - 클라이언트의 생성, 연결, 연결 끊기 등의 라이프 사이클관리 기능 제공



■ 활용범위 및 응용분야

	<table border="1"> <thead> <tr> <th>Structure</th> <th>Simulation variable</th> <th>Simulation type</th> <th>Simulation value</th> <th>Latest value</th> <th>Server variable</th> <th>Server type</th> </tr> </thead> <tbody> <tr> <td></td> <td>VALUE</td> <td>Real</td> <td>-35.169813819229667</td> <td>-35.169813819229667</td> <td>Joint-data-source_1</td> <td>Double</td> </tr> <tr> <td></td> <td>VALUE</td> <td>Real</td> <td>3.127698865609545</td> <td>3.127698865609545</td> <td>Joint-data-source_2</td> <td>Double</td> </tr> <tr> <td></td> <td>VALUE</td> <td>Real</td> <td>73.03036202217767</td> <td>73.03036202217767</td> <td>Joint-data-source_3</td> <td>Double</td> </tr> <tr> <td></td> <td>VALUE</td> <td>Real</td> <td>-51.591208912453425</td> <td>-51.591208912453425</td> <td>Joint-data-source_4</td> <td>Double</td> </tr> <tr> <td></td> <td>VALUE</td> <td>Real</td> <td>-34.739440007266062</td> <td>-34.739440007266062</td> <td>Joint-data-source_5</td> <td>Double</td> </tr> <tr> <td></td> <td>VALUE</td> <td>Real</td> <td>57.00236332156512</td> <td>57.00236332156512</td> <td>Joint-data-source_6</td> <td>Double</td> </tr> </tbody> </table>	Structure	Simulation variable	Simulation type	Simulation value	Latest value	Server variable	Server type		VALUE	Real	-35.169813819229667	-35.169813819229667	Joint-data-source_1	Double		VALUE	Real	3.127698865609545	3.127698865609545	Joint-data-source_2	Double		VALUE	Real	73.03036202217767	73.03036202217767	Joint-data-source_3	Double		VALUE	Real	-51.591208912453425	-51.591208912453425	Joint-data-source_4	Double		VALUE	Real	-34.739440007266062	-34.739440007266062	Joint-data-source_5	Double		VALUE	Real	57.00236332156512	57.00236332156512	Joint-data-source_6	Double
Structure	Simulation variable	Simulation type	Simulation value	Latest value	Server variable	Server type																																												
	VALUE	Real	-35.169813819229667	-35.169813819229667	Joint-data-source_1	Double																																												
	VALUE	Real	3.127698865609545	3.127698865609545	Joint-data-source_2	Double																																												
	VALUE	Real	73.03036202217767	73.03036202217767	Joint-data-source_3	Double																																												
	VALUE	Real	-51.591208912453425	-51.591208912453425	Joint-data-source_4	Double																																												
	VALUE	Real	-34.739440007266062	-34.739440007266062	Joint-data-source_5	Double																																												
	VALUE	Real	57.00236332156512	57.00236332156512	Joint-data-source_6	Double																																												
<p>[산업로봇 시스템용 자산 관리 셀 구현]</p>	<p>[CPS 가상화 시스템과의 동기화 연동]</p>																																																	

○ 스마트 공장을 위한 실시간 공정 분석 및 관리 자동화 응용 분야

■ 지식재산권 현황

구분	발명의 명칭	출원번호 (출원일)	등록번호 (등록일)
특허	AutomationML 데이터 모델을 OPC UA 정보 모델로 변환하는 방법 및 그 장치	2019-0144390 (2019.11.12)	진행중
특허	효율적인 OPC UA 서버의 데이터 접근을 위한 통합 게이트웨이 시스템	2019-0166330 (2019.12.13)	진행중
특허	OPC UA 표준 상호운용성 시험 장치 및 방법	2017-0088892 (2017.07.13)	10-2021180 (2019.09.05)
특허	관리-셀을 이용한 모듈 조합형 자동화 공정장비 관리 방법	2018-0151995 (2018.11.30)	진행중



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2021년08월31일
(11) 등록번호 10-2295100
(24) 등록일자 2021년08월24일

(51) 국제특허분류(Int. Cl.)
G06F 16/84 (2019.01) G05B 19/042 (2006.01)
(52) CPC특허분류
G06F 16/84 (2019.01)
G05B 19/042 (2013.01)
(21) 출원번호 10-2019-0144390
(22) 출원일자 2019년11월12일
심사청구일자 2019년11월12일
(65) 공개번호 10-2021-0057533
(43) 공개일자 2021년05월21일
(56) 선행기술조사문헌
Robert HenBen 외 1명, 'Interoperability
between OPC UA and AutomationML', Procedia
CIRP Volume 25, 2014년, pp. 297-304. 1부.*
(뒷면에 계속)

(73) 특허권자
한국전자기술연구원
경기도 성남시 분당구 새나리로 25 (야탑동)
(72) 발명자
정지은
경기도 성남시 분당구 동판교로 153 붓들마을8단
지아파트 807동 603호
이재현
서울특별시 송파구 삼전로8길 4-17 세원빌리지
302호
(74) 대리인
정종욱, 진천웅

전체 청구항 수 : 총 10 항

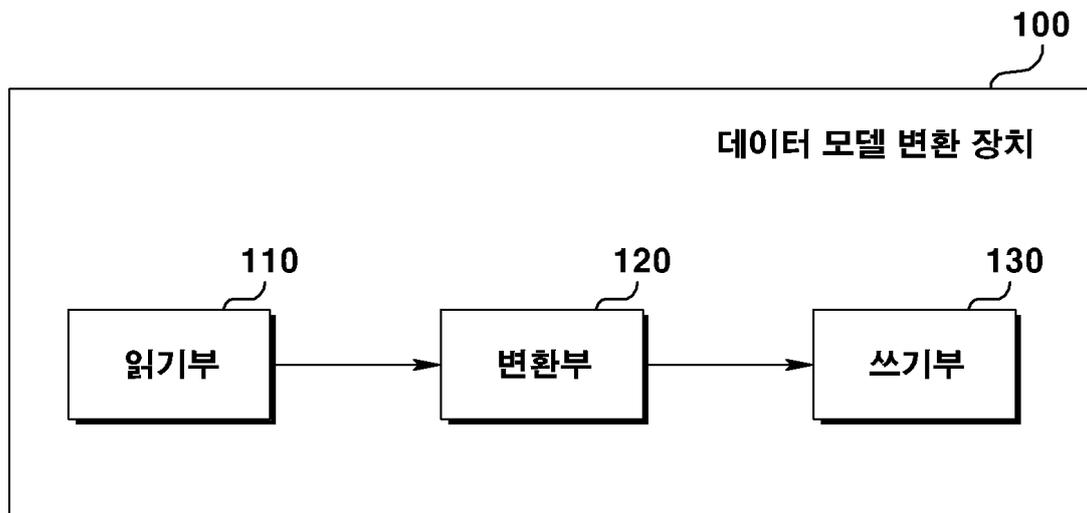
심사관 : 김경완

(54) 발명의 명칭 AutomationML 데이터 모델을 OPC UA 정보 모델로 변환하는 방법 및 그 장치

(57) 요약

일 실시예에 따른 데이터 모델 변환 방법은 AutomationML 모델에 따라 작성된 AutomationML 파일을 수신하고, 상기 수신한 AutomationML 파일로부터 AutomationML 엘리먼트와 상기 AutomationML 엘리먼트에 대한 데이터를 파싱하여 자료 구조를 생성하는 단계; 상기 생성된 자료 구조를 기초로, AutomationML 엘리먼트를 OPC UA 노드로 맵핑하는 맵핑 규칙을 이용하여 상기 AutomationML 엘리먼트를 OPC UA 노드로 변환하는 단계; 상기 변환된 OPC UA 노드를 이용하여 OPC UA 모델에 따른 OPC UA 파일을 생성하는 단계를 포함한다.

대표도 - 도1



(56) 선행기술조사문헌

US20180224821 A1

KR1020110081945 A

KR1020050114947 A

KR1020110026973 A

*는 심사관에 의하여 인용된 문헌

이 발명을 지원한 국가연구개발사업

과제고유번호 20000825

부처명 산업통상자원부

과제관리(전문)기관명 한국산업기술평가관리원

연구사업명 전자부품산업핵심기술개발사업-주력산업IT융합

연구과제명 전자디바이스 모듈화 장비 가상화를 위한 CPS 연동 및 검증기술 개발

기 여 율 1/1

과제수행기관명 전자부품연구원

연구기간 2018.04.01 ~ 2020.12.31

명세서

청구범위

청구항 1

데이터 모델 변환 장치가 AutomationML 모델에 따라 작성된 AutomationML 파일을 수신하는 단계;
 상기 데이터 모델 변환 장치가 상기 수신한 AutomationML 파일의 스키마 버전을 추출하는 단계;
 상기 데이터 모델 변환 장치가 외부로부터 상기 추출된 스키마 버전에 대응하는 스키마 파일을 수신하는 단계;
 및
 상기 데이터 모델 변환 장치가 상기 수신한 스키마 파일을 이용하여, 상기 수신한 AutomationML 파일로부터 AutomationML 엘리먼트와 상기 AutomationML 엘리먼트에 대한 데이터를 파싱하여 자료 구조를 생성하는 단계;
 상기 데이터 모델 변환 장치가 상기 생성된 자료 구조를 기초로, AutomationML 엘리먼트를 OPC UA 노드로 맵핑하는 맵핑 규칙을 이용하여 상기 AutomationML 엘리먼트를 OPC UA 노드로 변환하는 단계; 및
 상기 데이터 모델 변환 장치가 상기 변환된 OPC UA 노드를 이용하여 OPC UA 모델에 따른 OPC UA 파일을 생성하는 단계를 포함하는 데이터 모델 변환 방법.

청구항 2

삭제

청구항 3

제1항에 있어서,
 상기 AutomationML 엘리먼트를 OPC UA 노드로 변환하는 단계는,
 상기 AutomationML 파일의 엘리먼트 트리 구조에 대응되는 OPC UA 노드 트리를 생성하는 단계를 포함하는 데이터 모델 변환 방법.

청구항 4

제3항에 있어서,
 상기 OPC UA 파일을 생성하는 단계는,
 상기 생성된 OPC UA 노드 트리 중 루트 노드를 기준으로 OPC UA 모델에 따른 자료 구조를 생성하는 단계; 및
 상기 생성된 OPC UA 모델에 따른 자료 구조를 이용하여 XML 기반의 OPC UA 파일을 생성하는 단계를 포함하는 데이터 모델 변환 방법.

청구항 5

제1항에 있어서,
 상기 AutomationML 엘리먼트를 OPC UA 노드로 변환하는 단계는,
 상기 AutomationML 엘리먼트를 UAObject, UAObjectType, UAVariable, 또는 기타 노드로 맵핑하는 단계; 및
 상기 맵핑된 AutomationML 엘리먼트의 관계에 따라 OPC UA 노드 트리를 생성하는 단계를 포함하는 데이터 모델 변환 방법.

청구항 6

제5항에 있어서,
 상기 기타 노드는,

해당 엘리먼트는 AutomationML 엘리먼트를 참조하는 노드이고, 상기 AutomationML 엘리먼트에 맵핑되는 UA 노드 타입이 존재하지 않는 경우 맵핑되는 노드인 것을 특징으로 하는 데이터 모델 변환 방법.

청구항 7

하나 이상의 메모리 및 하나 이상의 프로세서를 포함하고,

상기 프로세서는,

AutomationML 모델에 따라 작성된 AutomationML 파일을 수신하고, 상기 수신한 AutomationML 파일로부터 AutomationML 엘리먼트와 상기 AutomationML 엘리먼트에 대한 데이터를 파싱하여 자료 구조를 생성하는 읽기부;

상기 생성된 자료 구조를 기초로, AutomationML 엘리먼트를 OPC UA 노드로 맵핑하는 맵핑 규칙을 이용하여 상기 AutomationML 엘리먼트를 OPC UA 노드로 변환하는 변환부; 및

상기 변환된 OPC UA 노드를 이용하여 OPC UA 모델에 따른 OPC UA 파일을 생성하는 쓰기부를 포함하고,

상기 읽기부는,

상기 수신한 AutomationML 파일의 스키마 버전을 추출하고,

통신부를 통해 외부로부터 상기 추출된 스키마 버전에 대응하는 스키마 파일을 수신하고,

상기 통신부를 통해 수신한 스키마 파일을 이용하여 상기 수신한 AutomationML 파일로부터 상기 AutomationML 엘리먼트와 상기 AutomationML 엘리먼트에 대한 데이터를 파싱하는 것을 특징으로 하는 데이터 모델 변환 장치.

청구항 8

삭제

청구항 9

제7항에 있어서,

상기 변환부는,

상기 AutomationML 파일의 엘리먼트 트리 구조에 대응되는 OPC UA 노드 트리를 생성하는 것을 특징으로 하는 데이터 모델 변환 장치.

청구항 10

제9항에 있어서,

상기 변환부는,

상기 생성된 OPC UA 노드 트리 중 루트 노드를 기준으로 OPC UA 모델에 따른 자료 구조를 생성하고,

상기 쓰기부는,

상기 생성된 OPC UA 모델에 따른 자료 구조를 이용하여 XML 기반의 OPC UA 파일을 생성하는 것을 특징으로 하는 데이터 모델 변환 장치.

청구항 11

제7항에 있어서,

상기 변환부는,

상기 AutomationML 엘리먼트를 UAObject, UAObjectType, UAVariable, 또는 기타 노드로 맵핑하고,

상기 맵핑된 AutomationML 엘리먼트의 관계에 따라 OPC UA 노드 트리를 생성하는 것을 특징으로 하는 데이터 모델 변환 장치.

청구항 12

제11항에 있어서,

상기 기타 노드는,

해당 엘리먼트는 AutomationML 엘리먼트를 참조하는 노드이고, 상기 AutomationML 엘리먼트에 맵핑되는 UA 노드 타입이 존재하지 않는 경우 맵핑되는 노드인 것을 특징으로 하는 데이터 모델 변환 장치.

발명의 설명

기술 분야

[0001] 본 발명은 데이터 모델 변환 방법에 관한 것으로, 보다 구체적으로 AutomationML 데이터 모델을 OPC UA 정보 모델로 변환하는 데이터 모델 변환 방법 및 데이터 모델 변환 장치에 관한 발명이다.

배경 기술

[0002] 미래 스마트 공장의 가장 중요한 특징으로 생산 시스템과 해당 IT 시스템에 대한 적응 용이성(Adaptability)을 들 수 있다. 즉, 스마트 공장은 지속적으로 생산 공정을 최적화하고 생산 설비를 변경할 수 있으며, 다양한 제품으로 변형 생산이 가능한 능력을 갖춰야 한다. 하지만 위와 같은 목표를 달성하기 위해서는 생산 기계와 설비 뿐만 아니라 관련 소프트웨어의 지속적인 재설정(Reconfiguration)을 요구한다. 하지만 현재의 생산 환경에서는 공장별로 더 나아가서는 공정 라인 별로 서로 다른 시스템을 사용하고 있으며, 시스템 별로 상이한 데이터 포맷과 데이터 전송 프로토콜을 채택하여 사용하고 있어 전체 시스템의 유연한 변경에 큰 장애가 되고 있다. 따라서 최근 다양한 생산 시스템 간의 상호호환성을 높이고 연결을 가능하도록 하는 AutomationML이나 OPC UA 표준과 같은 기술들이 제안되고 있으며, 더 나아가 두 표준간의 통합이 크게 주목 받고 있다.

[0003] AutomationML은 생산 시스템 엔지니어링 단계에서 데이터를 일관되게 교환함으로써 정보전달의 효율성을 높이고 상호호환성을 향상시키기 위해 제안된 표준(IEC 62714)이다. 생산 시스템의 엔지니어링 단계는 크게 공장 설계, 시공, 커미셔닝, 유지관리로 구성되며, 각 단계별로 사용하는 도구(예를 들어, solidworks, excel, AutoCAD 등)와 데이터 포맷이 매우 다양하다. 하지만 그동안 여러 엔지니어링 단계 간에 일관된 데이터 교환을 지원하는 공통 데이터 모델은 없었다. AutomationML은 이와 같이 여러 엔지니어링 단계에서 발생하는 이기종 형태의 데이터를 일관되게 교환하고자 고안된 개방형 데이터 표준 포맷이며, XML기반으로 데이터의 구문과 의미를 정의한다.

[0004] OPC UA는 서로 다른 산업용 네트워크 혹은 다른 공급업체의 장비 간의 수평적인 정보전달을 가능하게 하고, 필드 레벨부터 엔터프라이즈 레벨까지의 수직적인 구조에서의 정보전달을 통해 공장 내·외부의 다양한 사물 및 서비스와 상호호환을 가능하도록 하는 통신 기술 표준(IEC 62541)이다.

[0005] 그리고 최근 두 표준을 통합하여 활용하는 사례가 급격히 증가함에 따라 OPC Foundation과 AutomationML e.V 그룹에서는 조인트 워킹 그룹을 만들어 AutomationML 데이터 모델을 OPC UA 정보 모델로 표현하는 방법을 정의한 문서인 “OPC Unified Architecture Information Model for AutomationML” 을 발표하였다. 이는 AutomationML 과 OPC UA가 정의하는 정보 모델의 유사성을 기반으로 두 표준 데이터 모델 간의 상호 호환을 가능하게 하는 기준이 된다.

[0006] 다만, 여기서 해결해야 할 문제는 OPC UA 서버를 구성하기 위해 AutomationML 데이터 모델을 OPC UA 정보 모델로 변환해야 한다는 것이다. 기존에는 “OPC Unified Architecture Information Model for AutomationML” 표준 문서에서 제시한 변환 방법론에 따라 수동으로 AutomationML의 데이터 모델을 분석하고 수동으로 정보 모델로 변환했기 때문에, 변환 과정에서의 오류가 발생하기 쉽고, 정보의 양이 많아질수록 변환 작업에 많은 시간과 노력이 필요했다. 더불어 지속적인 버전 업그레이드를 통해 AutomationML이 기술하는 데이터의 종류가 증가하면서 수동 변환 방식은 한계에 다다랐다.

발명의 내용

해결하려는 과제

[0007] 본 발명이 해결하고자 하는 기술적 과제는, 생산 시스템의 각종 정보를 기술하는 AutomationML 데이터 모델을 OPC UA 정보 모델로 변환하는 데이터 모델 변환 방법 및 데이터 모델 변환 장치를 제공하는 것이다.

과제의 해결 수단

- [0008] 상기 기술적 과제를 해결하기 위하여, 본 발명의 일 실시예에 따른 데이터 모델 변환 방법은 AutomationML 모델에 따라 작성된 AutomationML 파일을 수신하고, 상기 수신한 AutomationML 파일로부터 AutomationML 엘리먼트와 상기 AutomationML 엘리먼트에 대한 데이터를 파싱하여 자료 구조를 생성하는 단계; 상기 생성된 자료 구조를 기초로, AutomationML 엘리먼트를 OPC UA 노드로 맵핑하는 맵핑 규칙을 이용하여 상기 AutomationML 엘리먼트를 OPC UA 노드로 변환하는 단계; 상기 변환된 OPC UA 노드를 이용하여 OPC UA 모델에 따른 OPC UA 파일을 생성하는 단계를 포함한다.
- [0009] 또한, 상기 수신한 AutomationML 파일로부터 AutomationML 엘리먼트와 상기 AutomationML 엘리먼트에 대한 데이터를 파싱하여 자료 구조를 생성하는 단계는, 상기 수신한 AutomationML 파일의 스키마 버전을 독출하는 단계; 상기 독출된 스키마 버전에 대응하는 스키마 파일을 저장부에서 독출하거나, 외부로부터 상기 독출된 스키마 버전에 대응하는 스키마 파일을 수신하는 단계; 및 상기 저장부에서 독출된 스키마 파일 또는 상기 수신한 스키마 파일을 이용하여 상기 수신한 AutomationML 파일로부터 상기 AutomationML 엘리먼트와 상기 AutomationML 엘리먼트에 대한 데이터를 파싱하는 단계를 포함할 수 있다.
- [0010] 또한, 상기 AutomationML 엘리먼트를 OPC UA 노드로 변환하는 단계는, 상기 AutomationML 파일의 엘리먼트 트리 구조에 대응되는 OPC UA 노드 트리를 생성하는 단계를 포함할 수 있다.
- [0011] 또한, 상기 OPC UA 파일을 생성하는 단계는, 상기 생성된 OPC UA 노드 트리 중 루트 노드를 기준으로 OPC UA 모델에 따른 자료 구조를 생성하는 단계; 및 상기 생성된 OPC UA 모델에 따른 자료 구조를 이용하여 XML 기반의 OPC UA 파일을 생성하는 단계를 포함할 수 있다.
- [0012] 또한, 상기 AutomationML 엘리먼트를 OPC UA 노드로 변환하는 단계는, 상기 AutomationML 엘리먼트를 UAObject, UAObjectType, UAVariable, 또는 기타 노드로 맵핑하는 단계; 및 상기 맵핑된 AutomationML 엘리먼트의 관계에 따라 OPC UA 노드 트리를 생성하는 단계를 포함할 수 있다.
- [0013] 또한, 상기 기타 노드는, 해당 엘리먼트는 AutomationML 엘리먼트를 참조하는 노드이고, 상기 AutomationML 엘리먼트에 맵핑되는 UA 노드 타입이 존재하지 않는 경우 맵핑되는 노드일 수 있다.
- [0014] 상기 기술적 과제를 해결하기 위하여, 본 발명의 일 실시예에 따른 데이터 모델 변환 장치는 하나 이상의 메모리 및 하나 이상의 프로세서를 포함하고, 상기 프로세서는, AutomationML 모델에 따라 작성된 AutomationML 파일을 수신하고, 상기 수신한 AutomationML 파일로부터 AutomationML 엘리먼트와 상기 AutomationML 엘리먼트에 대한 데이터를 파싱하여 자료 구조를 생성하는 읽기부; 상기 생성된 자료 구조를 기초로, AutomationML 엘리먼트를 OPC UA 노드로 맵핑하는 맵핑 규칙을 이용하여 상기 AutomationML 엘리먼트를 OPC UA 노드로 변환하는 변환부; 및 상기 변환된 OPC UA 노드를 이용하여 OPC UA 모델에 따른 OPC UA 파일을 생성하는 쓰기부를 포함한다.
- [0015] 또한, 상기 읽기부는, 상기 수신한 AutomationML 파일의 스키마 버전을 독출하고, 상기 독출된 스키마 버전에 대응하는 스키마 파일을 저장부에서 독출하거나, 통신부를 통해 외부로부터 상기 독출된 스키마 버전에 대응하는 스키마 파일을 수신하고, 상기 저장부에서 독출된 스키마 파일 또는 상기 통신부를 통해 수신한 스키마 파일을 이용하여 상기 수신한 AutomationML 파일로부터 상기 AutomationML 엘리먼트와 상기 AutomationML 엘리먼트에 대한 데이터를 파싱할 수 있다.
- [0016] 또한, 상기 변환부는, 상기 AutomationML 파일의 엘리먼트 트리 구조에 대응되는 OPC UA 노드 트리를 생성할 수 있다.
- [0017] 또한, 상기 변환부는, 상기 생성된 OPC UA 노드 트리 중 루트 노드를 기준으로 OPC UA 모델에 따른 자료 구조를 생성하고, 상기 생성된 OPC UA 모델에 따른 자료 구조를 이용하여 XML 기반의 OPC UA 파일을 생성할 수 있다.
- [0018] 또한, 상기 변환부는, 상기 AutomationML 엘리먼트를 UAObject, UAObjectType, UAVariable, 또는 기타 노드로 맵핑하고, 상기 맵핑된 AutomationML 엘리먼트의 관계에 따라 OPC UA 노드 트리를 생성할 수 있다.
- [0019] 또한, 상기 기타 노드는, 해당 엘리먼트는 AutomationML 엘리먼트를 참조하는 노드이고, 상기 AutomationML 엘리먼트에 맵핑되는 UA 노드 타입이 존재하지 않는 경우 맵핑되는 노드일 수 있다.

발명의 효과

- [0020] 본 발명의 실시예들에 따르면, AutomationML 데이터 모델을 정보의 유실이나 오류없이 OPC UA 정보 모델로 자동

변환할 수 있다.

- [0021] 또한, 변환된 OPC UA 정보 모델을 사용하면 대부분의 OPC UA SDK에서 제공하는 도구나 방법을 통해 쉽게 OPC UA 서버를 생성할 수 있으며, 생성된 OPC UA 서버를 통해 생산 시스템들과 연결하고 데이터를 전달할 수 있다.
- [0022] 또한, AutomationML 스키마를 변환하여 데이터 모델 자료구조를 생성함으로써, AutomationML 표준의 수정이나 버전 업그레이드에 쉽게 대응할 수 있다.
- [0023] 또한, 생산 관련 OT 및 IT 시스템이 사용하는 기기종의 데이터 포맷과 프로토콜을 OPC UA 표준 기반으로 통합하고 상호운용성을 향상시킬 수 있다.

도면의 간단한 설명

- [0024] 도 1은 본 발명의 일 실시예에 따른 데이터 모델 변환 장치의 블록도이다.
- 도 2는 본 발명의 다른 실시예에 따른 데이터 모델 변환 장치의 블록도이다.
- 도 3 내지 도 7은 본 발명의 실시예에 따른 데이터 모델 변환 장치에서 데이터 모델을 변환하는 과정을 설명하기 위한 도면이다.
- 도 8은 본 발명의 일 실시예에 따른 데이터 모델 변환 방법의 흐름도이다.
- 도 9 내지 도 12는 본 발명의 다른 실시예에 따른 데이터 모델 변환 방법의 흐름도이다.

발명을 실시하기 위한 구체적인 내용

- [0025] 이하, 첨부된 도면을 참조하여 본 발명의 바람직한 실시예를 상세히 설명한다.
- [0026] 다만, 본 발명의 기술 사상은 설명되는 일부 실시 예에 한정되는 것이 아니라 서로 다른 다양한 형태로 구현될 수 있고, 본 발명의 기술 사상 범위 내에서라면, 실시 예들간 그 구성 요소들 중 하나 이상을 선택적으로 결합 또는 치환하여 사용할 수 있다.
- [0027] 또한, 본 발명의 실시예에서 사용되는 용어(기술 및 과학적 용어를 포함)는, 명백하게 특별히 정의되어 기술되지 않는 한, 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자에게 일반적으로 이해될 수 있는 의미로 해석될 수 있으며, 사전에 정의된 용어와 같이 일반적으로 사용되는 용어들은 관련 기술의 문맥상의 의미를 고려하여 그 의미를 해석할 수 있을 것이다.
- [0028] 또한, 본 발명의 실시예에서 사용된 용어는 실시예들을 설명하기 위한 것이며 본 발명을 제한하고자 하는 것은 아니다.
- [0029] 본 명세서에서, 단수형은 문구에서 특별히 언급하지 않는 한 복수형도 포함할 수 있고, "A 및(와) B, C 중 적어도 하나(또는 한 개 이상)"로 기재되는 경우 A, B, C로 조합할 수 있는 모든 조합 중 하나 이상을 포함할 수 있다.
- [0030] 또한, 본 발명의 실시 예의 구성 요소를 설명하는데 있어서, 제1, 제2, A, B, (a), (b) 등의 용어를 사용할 수 있다. 이러한 용어는 그 구성 요소를 다른 구성 요소와 구별하기 위한 것일 뿐, 그 용어에 의해 해당 구성요소의 본질이나 차례 또는 순서 등으로 한정되지 않는다.
- [0031] 그리고, 어떤 구성 요소가 다른 구성 요소에 '연결', '결합', 또는 '접속'된다고 기재된 경우, 그 구성 요소는 그 다른 구성 요소에 직접적으로 '연결', '결합', 또는 '접속'되는 경우뿐만 아니라, 그 구성 요소와 그 다른 구성 요소 사이에 있는 또 다른 구성 요소로 인해 '연결', '결합', 또는 '접속'되는 경우도 포함할 수 있다.
- [0032] 또한, 각 구성 요소의 "상(위)" 또는 "하(아래)"에 형성 또는 배치되는 것으로 기재되는 경우, "상(위)" 또는 "하(아래)"는 두 개의 구성 요소들이 서로 직접 접촉되는 경우뿐만 아니라, 하나 이상의 또 다른 구성 요소가 두 개의 구성 요소들 사이에 형성 또는 배치되는 경우도 포함한다. 또한, "상(위)" 또는 "하(아래)"로 표현되는 경우 하나의 구성 요소를 기준으로 위쪽 방향뿐만 아니라 아래쪽 방향의 의미도 포함될 수 있다.
- [0034] 도 1은 본 발명의 일 실시예에 따른 데이터 모델 변환 장치의 블록도이다. 본 발명의 일 실시예에 따른 데이터 모델 변환 장치(100)는 하나 이상의 메모리 및 하나 이상의 프로세서를 포함하고, 상기 프로세서는 읽기부(110), 변환부(120), 및 쓰기부(130)로 구성된다. 또한, 도 2와 같이, 저장부(140) 또는 통신부(150)를 더 포함할 수 있다. 읽기부(110), 변환부(120), 및 쓰기부(130)는 하나의 프로세서에 포함되거나 하나 이상이 다른 프

로세서에 포함될 수 있다.

- [0035] 읽기부(110)는 AutomationML(AML) 모델에 따라 작성된 AutomationML 파일을 수신하고, 상기 수신한 AutomationML 파일로부터 AutomationML 엘리먼트와 상기 AutomationML 엘리먼트에 대한 데이터를 파싱하여 자료 구조를 생성한다.
- [0036] 보다 구체적으로, 읽기부(Reader, 110)는 AutomationML 모델에 따라 작성된 AutomationML 파일을 수신한다. AutomationML 파일을 수신한 후, OPC UA 모델로 변환하기에 앞서 AutomationML 파일로부터 AutomationML 엘리먼트(Element)와 AutomationML 엘리먼트에 대한 데이터를 파싱한다. 파싱된 데이터들을 자료 구조를 생성하여 저장한다.
- [0037] AutomationML은 IEC 62424 표준으로 정의된 CAEX(Computer Aided Engineering Exchange) 클래스 모델을 기반으로 데이터 모델을 표현한다.
- [0038] 여기서, CAEX는 공장의 계층 구조와 같이, 계층적 객체 정보를 저장하는 데이터 포맷이다. 특정 추상화 레벨에서, 공장은 서로 연결된 모듈 또는 구성 요소로 구성되는데, CAEX를 이용하면 객체를 통해 해당 모듈 또는 구성 요소를 저장할 수 있다. 캡슐화, 클래스, 클래스 라이브러리, 인스턴스, 인스턴스 계층, 상속, 관계, 속성 및 인터페이스와 같은 객체 지향 개념이 명시적으로 지원된다. CAEX는 XML을 기반으로 하며 XML 스키마(xsd 파일)로 정의된다. CAEX 개발의 원래 의도는 프로세스 엔지니어링 툴과 프로세스 제어 엔지니어링 툴 간의 공통적이고 확립된 데이터 교환을 위한 것이나, CAEX는 모든 유형의 정적 객체 정보에 적용될 수 있다. 예를 들어 공장 토폴로지, 문서 토폴로지, 제품 토폴로지, 페트리 넷트등에 적용될 수 있다.
- [0039] AutomationML은 도 3과 같이 크게 4가지 영역으로 나뉘어 각 영역에 맞는 엘리먼트를 정의한다.
- [0040] Interface Class Library(340)는 포트(Port)나 External Data Connector와 같이 내, 외부의 컴포넌트와 연결하기 위한 여러 가지 Interface Class들을 정의하는 부분이다.
- [0041] Role Class Library(330)는 컴포넌트에 Group, Resource, Product, Process와 같은 의미를 부여하기 위해 사용할 Role Class들을 정의하는 부분이다.
- [0042] System Unit Library(320)는 장비나 사물의 특성을 포함한 System Unit Class(SUC)를 정의할 수 있고, 해당 클래스는 기존 정의된 Role Class와 Interface Class를 참조하여 역할과 인터페이스를 연결할 수 있다.
- [0043] InstanceHierarchy(IH, 310)에서는 SUC를 기반으로 하는 Internal Element(IE) 인스턴스를 정의할 수 있으며, Role Class와 Interface Class를 참조하여 역할과 인터페이스를 할당하거나, Internal Link를 사용하여 인스턴스 간 연결 관계를 정의할 수 있다. 각 Class 엘리먼트들은 External Interface를 사용하여 외부 데이터를 참조할 수 있고, Attribute를 통해 속성값을 가질 수 있다.
- [0044] AutomationML 파일을 변환하고자 하는 OPC UA는 OPC UA 서버에서 관리하고자 하는 노드들의 집합을 기술하는 XML 포맷의 표준(IEC 62541-5)이다. OPC UA 노드 모델은 도 4와 같이, 각 노드(410)에 대한 아이디, 이름, 설명 등의 정보를 Attribute(411)로 기술할 수 있고, 노드의 타입이나 다른 노드(420)와의 노드 간 관계 등은 Reference(412)로 추가할 수 있는 유연하고 확장적인 구조를 가지고 있다.
- [0045] 읽기부(110)는 OPC UA로 변환하기 위해 변환이 적용되는 엘리먼트들과 엘리먼트들이 어떤 데이터들을 가지고 있는지를 AutomationML 파일을 파싱하여 자료 구조로 생성한다.
- [0046] 읽기부(110)는 AutomationML 데이터 모델을 파싱함에 있어서, 스키마 파일을 이용한다. 여기서 스키마 파일은 데이터 모델의 구조와 조건에 대한 파일로, 데이터 모델을 구성하는 개체, 속성, 관계 및 데이터 조작시 데이터 값들이 갖는 제약 조건 등에 관해 전반적으로 정의하고 있는 파일이다. 스키마 파일은 데이터의 구조적 특성을 의미하며, 인스턴스에 의해 규정될 수 있다. 스키마 파일은 해당 파일이 유효한 것으로 여겨지기 위해 반드시 지켜야 하는 규칙들의 집합에 해당한다.
- [0047] 스키마(Schema)는 요소 선언(Element declarations) 및 속성 선언(Attribute declarations)을 포함할 수 있다.
- [0048] 요소 선언, 요소의 속성(properties)을 정의한다. 요소 이름과 대상 네임스페이스를 포함한다. 요소의 종류(type)가 중요한 속성으로, 요소가 어떤 속성과 자식(children)을 가질 수 있는 지를 제한한다. 요소 종류는 그 특성 값에 따라 달라질 수 있다. 요소는 대체 그룹(substitution group)에 속할 수 있다. 만약 요소 E가 요소 H의 대체 그룹에 속하면, 스키마가 H를 허용하는 곳 어디에나 E가 나타날 수 있다. 요소는 무결성 조건(integrity constraints)을 가질 수 있다. 요소 선언은 전역(global)이거나 지역(local)일 수 있다. 따라서 인

스턴스 문서(instance document)의 서로 다른 부분에서, 서로 관련이 없는 요소들을 가리키는 데 같은 이름을 쓸 수 있다.

- [0049] 속성 선언은, 속성의 속성을 정의한다. 여기서도 속성 이름과 대상 네임스페이스를 포함한다. 속성의 타입은 속성이 가질 수 있는 값을 제한한다. 속성 선언에 기본 값을 지정하거나, 값을 고정할 수 있다.
- [0050] 스키마는 외부 스키마, 개념 스키마, 내부 스키마로 나뉠 수 있다. 외부 스키마(External Schema)는 서브 스키마 또는 사용자 뷰(View)로 표현될 수 있으며, 외부 스키마는 사용자나 응용 프로그래머가 각 개인의 입장에서 필요로 하는 데이터베이스의 논리적 구조를 정의한 것이다. 외부 스키마는 전체 데이터베이스의 한 논리적인 부분으로 볼 수 있으므로 서브 스키마(Sub Schema)라고도 한다. 하나의 데이터베이스 시스템에는 여러 개의 외부 스키마가 존재할 수 있으며, 하나의 외부 스키마를 여러 개의 응용 프로그램이나 사용자가 공유할 수도 있다. 같은 데이터베이스에 대해서도 서로 다른 관점을 정의할 수 있도록 허용한다. 일반 사용자는 질의어(SQL)를 이용하여 DB를 쉽게 사용할 수 있고, 응용 프로그래머는 COBOL, C 등의 언어를 사용하여 DB에 접근할 수 있다.
- [0051] 개념 스키마(Coneptual Schema)는 전체적인 뷰(View)에 해당하고, 개념 스키마는 데이터베이스의 전체적인 논리적 구조로서, 모든 응용 프로그램이나 사용자들이 필요로 하는 데이터를 종합한 조직 전체의 데이터베이스로 하나만 존재한다. 개념 스키마는 개체 간의 관계나 제약 조건을 나타내고 데이터베이스의 접근 권한, 보안 및 무결성 규칙에 관한 명세를 정의한다. 데이터베이스 파일에 저장되는 데이터의 형태를 나타내는 것으로, 단순히 스키마(Schema)라고 하면 개념 스키마를 의미한다. 기관이나 조직체의 관점에서 데이터베이스를 정의한 것이다. 데이터베이스 관리자(DBA)에 의해서 구성된다.
- [0052] 내부 스키마(Internal Schema)는 저장 스키마(Storage Schema)로 표현될 수 있다. 내부 스키마는 물리적 저장 장치의 입장에서 본 데이터베이스 구조로, 물리적인 저장장치와 밀접한 계층이다. 내부 스키마는 실제로 데이터베이스에 저장될 레코드의 물리적인 구조를 정의하고, 저장 데이터 항목의 표현 방법, 내부 레코드의 물리적 순서 등을 나타낸다. 시스템 프로그래머나 시스템 설계자가 보는 관점의 스키마이다.
- [0053] 읽기부(110)는 CAEX 스키마 파일인 XSD (XML Schema Definition)를 이용하여 AutomationML 파일을 파싱하여 자료 구조를 생성한다.
- [0054] 이때, 읽기부(110)는 상기 수신한 AutomationML 파일의 스키마 버전을 독출하고, 상기 독출된 스키마 버전에 대응하는 스키마 파일을 저장부(140)에서 독출하거나, 통신부(150)를 통해 외부로부터 상기 독출된 스키마 버전에 대응하는 스키마 파일을 수신하고, 저장부(140)에서 독출된 스키마 파일 또는 통신부(150)를 통해 수신한 스키마 파일을 이용하여 상기 수신한 AutomationML 파일로부터 상기 AutomationML 엘리먼트와 상기 AutomationML 엘리먼트에 대한 데이터를 파싱할 수 있다.
- [0055] 스키마 파일을 이용하여 AutomationML 파일을 파싱하기 위해선, AutomationML 파일의 스키마 버전에 해당하는 스키마 파일을 이용하여야 한다. 스키마 버전이 상이한 경우, 규칙이 상이할 수 있기 때문에, 정확한 파싱이 어려워질 수 있다.
- [0056] 따라서, 읽기부(110)는 먼저 파싱할 AutomationML 파일의 스키마 버전을 독출한다. 상기 독출된 스키마 버전에 대응하는 스키마 파일이 저장부(140)에 저장되어 있는 경우, 해당 스키마 파일을 저장부(140)에서 독출하여 이용할 수 있다. 하지만, 저장부(140)에 해당 스키마 파일이 저장되어 있지 않은 경우, 통신부(150)를 통해 외부로부터 상기 독출된 스키마 버전에 대응하는 스키마 파일을 수신하여 이용한다. 저장부(140)에 저장된 스키마 파일이 존재하지 않는 초기 상태에서도 통신부(150)를 통해 외부로부터 스키마 파일을 수신할 수 있다. 해당 스키마 파일은 AutomationML 사이트에서 제공되는 스키마 파일일 수 있다. 수신한 스키마 파일은 저장부(140)에 저장할 수 있다. 스키마 파일이 업데이트 되는 경우, 통신부(150)는 업데이트된 스키마 파일을 수신하여, 저장부(140)에 저장할 수 있다. 이를 통해 AutomationML 모델의 버전 업데이트 시에도 변환 규칙만 추가 개발하여 빠른 대응이 가능하다.
- [0057] 파싱할 AutomationML 파일의 스키마 버전에 대응하는 스키마 파일을 저장부(140) 또는 통신부(150)를 통해 수신한 이후, 해당 스키마 파일을 이용하여 상기 수신한 AutomationML 파일로부터 상기 AutomationML 엘리먼트와 상기 AutomationML 엘리먼트에 대한 데이터를 파싱한다. 파싱된 엘리먼트 및 데이터는 자료 구조로 생성되어 저장된다. 이때, 생성된 자료 구조는 저장부(140)에 저장될 수 있다.
- [0058] 읽기부(110)에서 AutomationML 파일을 파싱하여 자료 구조를 생성하는 과정은 도 5와 같이 수행될 수 있다. 읽기부(110)는 AutomationML 파일(510)을 수신하고, AutomationML Library Manager(520)는 아래 예시와 같이 AutomationML 파일 내에 정의된 External Reference 엘리먼트들을 탐색하여 다른 AutomationML 파일에 기록된

외부 AutomationML 데이터 모델을 읽고 위와 동일한 과정을 거쳐 참조를 위한 자료 구조를 생성할 수 있다.

- [0059] <ExternalReference Path="Libs/RoleClass Libraries/AutomationMLBaseRoleClassLib.AutomationML" Alias="BaseRoleClassLib"/>
- [0060] <ExternalReference Path="Libs/InterfaceClass Libraries/AutomationMLInterfaceClassLib.AutomationML" Alias="BaseInterfaceClassLib"/>
- [0061] 변환부(120)는 상기 생성된 자료 구조를 기초로, AutomationML 엘리먼트를 OPC UA 노드로 맵핑하는 맵핑 규칙을 이용하여 상기 AutomationML 엘리먼트를 OPC UA 노드로 변환한다.
- [0062] 보다 구체적으로, 변환부(120)는 읽기부(110)에서 생성된 자료 구조를 순차적으로 탐색하며, AutomationML 엘리먼트를 OPC UA 노드로 재구성한다. 이때, AutomationML 엘리먼트와 OPC UA 노드는 1 대 1 맵핑 관계가 아닌바, AutomationML 엘리먼트가 가진 데이터를 이용하여 해당 데이터와 관계가 동일한 의미의 OPC UA 노드로 재구성한다. AutomationML 엘리먼트를 OPC UA 노드로 맵핑하는 맵핑 규칙은 맵핑 테이블로 저장할 수 있다. 맵핑 규칙은 AutomationML 스키마 파일 또는 OPC UA 버전이 업데이트되거나, 규칙이 달라지는 경우, 업데이트될 수 있다.
- [0063] 변환부(120)는 읽기부(110)에서 생성된 자료 구조를 순차적으로 탐색하며, 각 AutomationML 엘리먼트에 해당하는 UA 노드(Node) 타입, UA 참조(Reference) 타입, UA 참조 노드를 도출한다.
- [0064] UA 노드 타입은 AutomationML 엘리먼트가 UA 노드 중 어떤 종류의 노드인지에 따라 나누어 지는 것으로, UA 노드 타입은 도 6과 같이, UAObject, UAObjectType, UAVariable 등으로 나누어 질 수 있다. 이때, 변환부(120)는 AutomationML 엘리먼트를 UAObject, UAObjectType, UAVariable, 또는 기타 노드로 맵핑한다. 여기서, 상기 기타 노드는, 해당 엘리먼트는 AutomationML 엘리먼트를 참조하는 노드이고, 상기 AutomationML 엘리먼트에 맵핑되는 UA 노드 타입이 존재하지 않는 경우 맵핑되는 노드일 수 있다. AutomationML 엘리먼트와 UA 노드 타입의 종류가 정확히 일치하지 않기 때문에, AutomationML 엘리먼트에 대응하는 UA 노드 타입이 존재하지 않는 경우, 해당 AutomationML 엘리먼트는 기타 노드로 맵핑할 수 있다. 도 6과 같이, AutomationML 엘리먼트 중 SupportedRoleClass 또는 RoleRequirement 는 대응되는 UA 노드 타입이 존재하지 않기 때문에, 해당 UA 노드 타입을 비워두거나 기타 노드 타입으로 맵핑할 수 있다.
- [0065] UA 참조 타입은 해당 노드가 참조하는 노드가 있는 경우, 어떤 노드를 어떤 의미로 참조하는지에 따라 나누어 지는 것으로, UA 참조 타입은 HasTypeDefinition, HasSubType 등으로 나누어질 수 있다. 앞서 설명한 바와 같이, AutomationML 엘리먼트 중 대응되는 UA 노드 타입이 존재하지 않지만, 참조가 존재하는 경우, 도 6과 같이, SupportedRoleClass는 UA 참조 타입으로 1:HasAutomationMLSupportedPoleClass를 맵핑하고, RoleRequirement는 1:HasAutomationMLRoleRequirement를 맵핑할 수 있다.
- [0066] UA 참조 노드는 해당 노드가 참조하는 노드가 있는 경우, 참조하는 노드를 맵핑하는 것이다. 도 6과 같이, 1:CAEXFileType, FolderType, PropertyType, *RefSystemUnitClassPath, *RefRoleClassPath, *RefBaseRoleClassPath 등으로 맵핑될 수 있다.
- [0067] 변환부(120)는 AutomationML 파일의 엘리먼트 트리 구조에 대응되는 OPC UA 노드 트리를 생성할 수 있다. 맵핑된 AutomationML 엘리먼트의 관계에 따라 OPC UA 노드 트리를 생성할 수 있다.
- [0068] 도 6을 참조하면, InterfaceClassLib, RoleClassLib, SystemUnitClassLib 엘리먼트 하위에서 정의된 InterfaceClass, RoleClass, SystemUnitClass 엘리먼트들은 다른 노드에서 참조할 UAObjectType 노드로 맵핑된다. InstanceHierarchy 엘리먼트 하위의 InternalElement 엘리먼트 등의 노드는 UAObject 노드로 맵핑되며 SupportedRoleClass, RoleRequirements 엘리먼트 등이 포함된 경우, Reference로써 상기 기술한 UAObjectType 노드들을 참조한다. Attribute 엘리먼트는 UAVariable 노드로 맵핑되고, 해당 엘리먼트를 소유한 부모 엘리먼트가 맵핑된 노드의 자식으로 추가된다. 최종적으로 변환된 OPC UA 서버는 도 7과 같이 InstanceHierarchy에서 기술된 엘리먼트 트리 구조와 동일한 노드 트리를 가지게 된다.
- [0069] 각 맵핑된 노드 및 참조 관계들을 이용하여 각 노드들을 노드 트리로 구성하여 OPC UA 노드 트리를 생성할 수 있다. 이를 통해, OPC UA 노드 트리 중 루트 노드를 기준으로 OPC UA 모델에 따른 자료 구조를 생성할 수 있다. 도 7과 같이, AutomationML 파일의 InstanceHierarchy에서 기술된 엘리먼트 트리 구조에 대응되도록 UA 노드들을 트리 구조로 배치하여 OPC UA 노드 트리 구조를 생성할 수 있다.
- [0070] AutomationML의 IH(InstanceHierarchy)가 맵핑된 노드를 루트 노드로 하여 노드 트리를 생성한다. 예를 들어, IH에는 도 7과 같이, IE로 'Conveyor'와 'MyRobot'이 있고, 'IE:Conveyor'의 SUC(SystemUnitClass)는

'Conveyor1'이고, RC(RoleClass)는 'beltConveyor'일 수 있다. 'IE:MyRobot'의 Attribute는 'Number0'이고, IE로 'Servo Motor'를 포함할 수 있다. 'IE:Servo Motor'의 RC(RoleClass)는 'Actuator'일 수 있다.

- [0071] 이러한 AutomationML 트리구조에 대응되도록 OPA UA 노드 트리를 생성한다.
- [0072] 예를 들어 도 7과 같이, AutomationML IH에 대응되는 루트 노드인 Object(710)을 기준으로, 해당 Object(710)의 자식관계인 Object(720, 730)들을 HasComponent를 통해 연결한다. 여기서, Object(720, 730)은 AutomationML의 'IE:Conveyor' 및 'IE:MyRobot'이 맵핑된 UA 노드이다. Object(720)의 ObjectType(740)은 'IE:Conveyor'의 SUC 및 RC가 맵핑된 UA 노드로 HasTypeDefinition을 통해 연결한다.
- [0073] Object(730)에 연결된 Variable(750) 및 Object(760)는 각각 'IE:MyRobot'의 Attribute 및 'IE:Servo Motor'이 맵핑된 UA 노드로, HasComponent를 통해 연결한다. ObjectType(770)은 'IE:Servo Motor'의 RC가 맵핑된 UA 노드로 HasTypeDefinition을 통해 연결한다.
- [0074] 상기와 같이, AutomationML 엘리먼트들이 맵핑된 UA 노드들을 노드 트리 구조로 생성하고, 이를 자료 구조로 저장한다. 생성된 자료 구조는 저장부(140)에 저장될 수 있다.
- [0075] 쓰기부(130)는 상기 변환된 OPC UA 노드를 이용하여 OPC UA 모델에 따른 OPC UA 파일을 생성한다.
- [0076] 보다 구체적으로, 변환부(120)에서 AutomationML 엘리먼트들을 OPC UA 노드로 맵핑하여 생성한 노드 트리 구조를 이용하여 OPC UA 모델에 따른 OPC UA 파일을 생성한다. 쓰기부(130)는 상기 생성된 OPC UA 모델에 따른 자료 구조를 이용하여 XML 기반의 OPC UA 파일을 생성할 수 있다.
- [0077] 쓰기부(130)는 UANodeSet 엘리먼트를 루트로 하여 OPC UA 정보 모델이 가져야하는 기본적인 XML 문서의 구조를 생성하고 OPC UA 정보 모델을 위한 네임 스페이스와 AutomationML 데이터 모델을 위한 네임 스페이스를 아래와 같이 추가한다.
- [0078] <NamespaceUri>
- [0079] <Uri>http://opcfoundation.org/UA/AutomationML/</Uri>
- [0080] <Uri>http://smic.kr/Topology.xml</Uri>
- [0081] </NamespaceUri>
- [0082] 이후, 변환 과정 중에 사용된 다양한 DataType의 BrowseName과 NodeId를 기반으로 Alias 엘리먼트들을 생성하여 추가한다. 마지막으로 AutomationML 데이터 모델로부터 변환된 UAObject, UAVariable, UAObjectType 등의 노드들을 추가하고 사용자가 지정한 이름의 파일로 저장한다.
- [0083] 이와 같이, AutomationML 파일을 OPC UA 파일로 생성한 이후, 통합하고자 하는 OPC UA 파일과 함께, OPC UA 서버를 구축할 수 있다. 이를 통해, AutomationML 파일에 따른 정보 및 OPC UA 파일에 따른 정보들을 통합 관리가 가능해진다. 이와 같이 OPC UA 통신 기술을 통해 AutomationML 데이터 모델을 연결하면, 생산 계획, 구조 및 토폴로지, 연결 정보와 같은 정적 데이터는 AutomationML(*AutomationML)로부터 가져오고, 운영 데이터, 프로세스와 같은 동적 데이터는 OPC UA로부터 가져올 수 있어, 앞서 설명한 전체 생산 시스템의 유연한 변경이 가능하고 시스템 관리가 용이해진다. 예를 들어, 공장 생산 시스템에 변경 사항이 발생했을 경우, AutomationML을 사용하여 공장을 구성하는 컴포넌트의 속성 및 기능을 수정하고, 이를 제어 시스템용 OPC UA 정보 모델로 변환하면, 시스템 변경 사항을 바로 OPC UA 서버에 반영할 수 있고, 변경된 속성이나 기능을 빠르게 적용할 수 있다.
- [0085] 도 8은 본 발명의 일 실시예에 따른 데이터 모델 변환 방법의 흐름도이고, 도 9 내지 도 12는 본 발명의 다른 실시예에 따른 데이터 모델 변환 방법의 흐름도이다. 도 8 내지 도 12의 각 단계에 대한 상세한 설명은 도 1 내지 도 7의 데이터 모델 변환 장치에 대한 상세한 설명에 대응되는바, 이하, 중복되는 설명은 생략하도록 한다.
- [0086] 본 발명의 일 실시예에 따른 이미지 처리 방법은 하나 이상의 프로세서를 포함하는 데이터 모델 변환 장치 또는 데이터 모델을 변환하는 프로세서를 포함하는 다른 장치에서 데이터 모델을 변환하는 방법에 관한 것이다. 프로세서는 AutomationML 파일을 OPC UA 파일로 변환하는 프로그램을 처리할 수 있고, AutomationML 파일을 OPC UA 파일로 변환하는 프로그램은 메모리 상에 저장되어 있을 수 있다.
- [0087] 본 발명의 일 실시예에 따른 데이터 모델 변환 방법은 S11 단계에서 AutomationML(이하, AML) 모델에 따라 작성된 AutomationML 파일을 수신하고, 상기 수신한 AutomationML 파일로부터 AutomationML 엘리먼트와 상기 AutomationML 엘리먼트에 대한 데이터를 파싱하여 자료 구조를 생성한다.

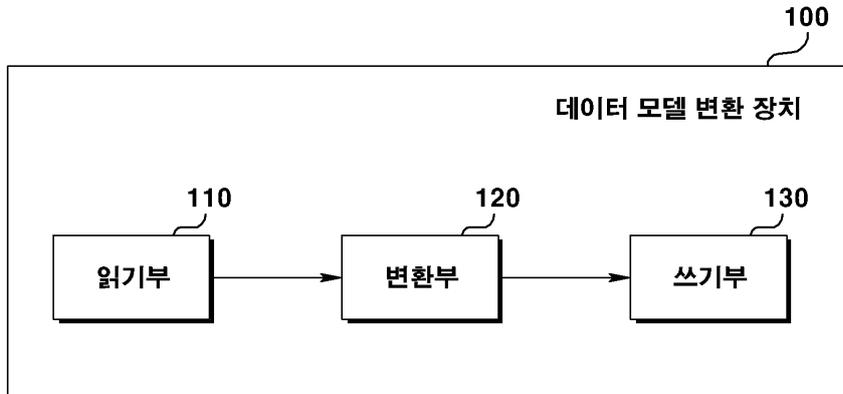
- [0088] 이때, S11 단계는 S21 단계 내지 S24 단계를 통해 수행될 수 있다. S21 단계에서 상기 수신한 AutomationML 파일의 스키마 버전을 독출하고, S22 단계에서 상기 독출된 스키마 버전에 대응하는 스키마 파일을 저장부에서 독출하거나, 외부로부터 상기 독출된 스키마 버전에 대응하는 스키마 파일을 수신하고, S23 단계에서 상기 저장부에서 독출된 스키마 파일 또는 상기 수신한 스키마 파일을 이용하여 상기 수신한 AutomationML 파일로부터 상기 AutomationML 엘리먼트와 상기 AutomationML 엘리먼트에 대한 데이터를 파싱할 수 있다. 이와 같이, 파싱된 정보들로 구성된 자료 구조를 생성할 수 있다.
- [0089] S11 단계 이후, S12 단계에서 상기 생성된 자료 구조를 기초로, AutomationML 엘리먼트를 OPC UA 노드로 맵핑하는 맵핑 규칙을 이용하여 상기 AutomationML 엘리먼트를 OPC UA 노드로 변환하고, S13 단계에서 상기 변환된 OPC UA 노드를 이용하여 OPC UA 모델에 따른 OPC UA 파일을 생성한다.
- [0090] S12 단계 이후, S31 단계에서 상기 AutomationML 파일의 엘리먼트 트리 구조에 대응되는 OPC UA 노드 트리를 생성하는 단계를 더 포함할 수 있다.
- [0091] S31 단계 이후, S41 단계에서 상기 생성된 OPC UA 노드 트리 중 루트 노드를 기준으로 OPC UA 모델에 따른 자료 구조를 생성하고, S42 단계에서 상기 생성된 OPC UA 모델에 따른 자료 구조를 이용하여 XML 기반의 OPC UA 파일을 생성할 수 있다.
- [0092] S12 단계는 S51 단계 및 S52 단계를 통해 수행될 수 있다. S51 단계에서 상기 AutomationML 엘리먼트를 UAObject, UAObjectType, UAVariable, 또는 기타 노드로 맵핑하고, S52 단계에서 상기 맵핑된 AutomationML 엘리먼트의 관계에 따라 OPC UA 노드 트리를 생성할 수 있다. 여기서, 상기 기타 노드는, 해당 엘리먼트는 AutomationML 엘리먼트를 참조하는 노드이고, 상기 AutomationML 엘리먼트에 맵핑되는 UA 노드 타입이 존재하지 않는 경우 맵핑되는 노드일 수 있다.
- [0094] 한편, 본 발명의 실시예들은 컴퓨터로 읽을 수 있는 기록 매체에 컴퓨터가 읽을 수 있는 코드로 구현하는 것이 가능하다. 컴퓨터가 읽을 수 있는 기록 매체는 컴퓨터 시스템에 의하여 읽혀질 수 있는 데이터가 저장되는 모든 종류의 기록장치를 포함한다.
- [0095] 컴퓨터가 읽을 수 있는 기록 매체의 예로는 ROM, RAM, CD-ROM, 자기 테이프, 플로피디스크, 광 데이터 저장장치 등이 있으며, 또한, 컴퓨터가 읽을 수 있는 기록 매체는 네트워크로 연결된 컴퓨터 시스템에 분산되어, 분산 방식으로 컴퓨터가 읽을 수 있는 코드가 저장되고 실행될 수 있다. 그리고 본 발명을 구현하기 위한 기능적인 (functional) 프로그램, 코드 및 코드 세그먼트들은 본 발명이 속하는 기술 분야의 프로그래머들에 의하여 용이하게 추론될 수 있다.
- [0096] 이상과 같이 본 발명에서는 구체적인 구성 요소 등과 같은 특정 사항들과 한정된 실시예 및 도면에 의해 설명되었으나 이는 본 발명의 보다 전반적인 이해를 돕기 위해서 제공된 것일 뿐, 본 발명은 상기의 실시예에 한정되는 것은 아니며, 본 발명이 속하는 분야에서 통상적인 지식을 가진 자라면 이러한 기재로부터 다양한 수정 및 변형이 가능하다.
- [0097] 따라서, 본 발명의 사상은 설명된 실시예에 국한되어 정해져서는 아니되며, 후술하는 특허청구범위뿐 아니라 이 특허청구범위와 균등하거나 등가적 변형이 있는 모든 것들은 본 발명 사상의 범주에 속한다고 할 것이다.

부호의 설명

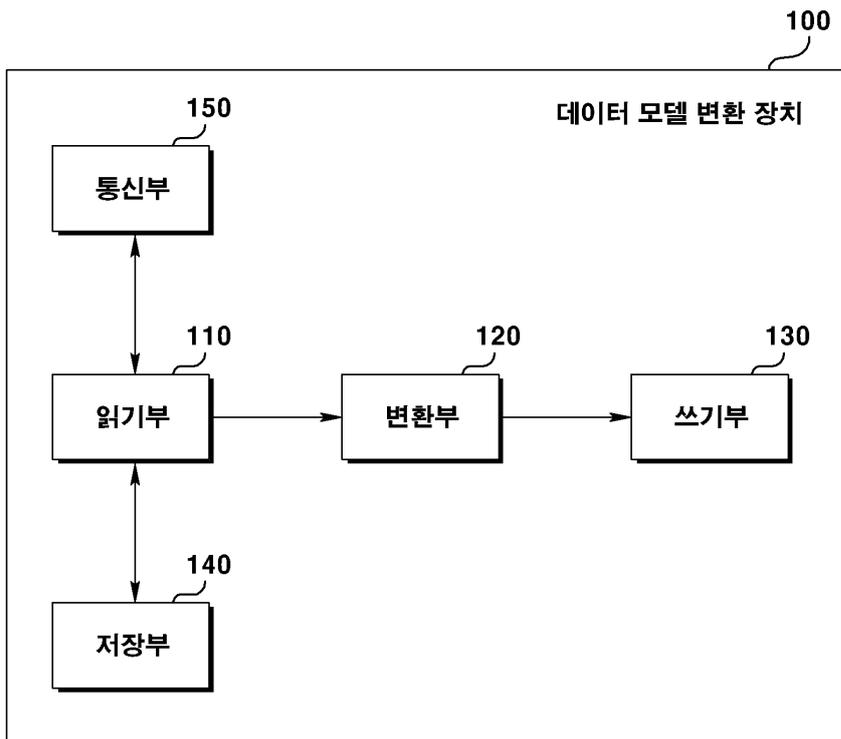
- [0098] 100: 데이터 모델 변환 장치
- 110: 읽기부
- 120: 변환부
- 130: 쓰기부
- 140: 저장부
- 150: 통신부

도면

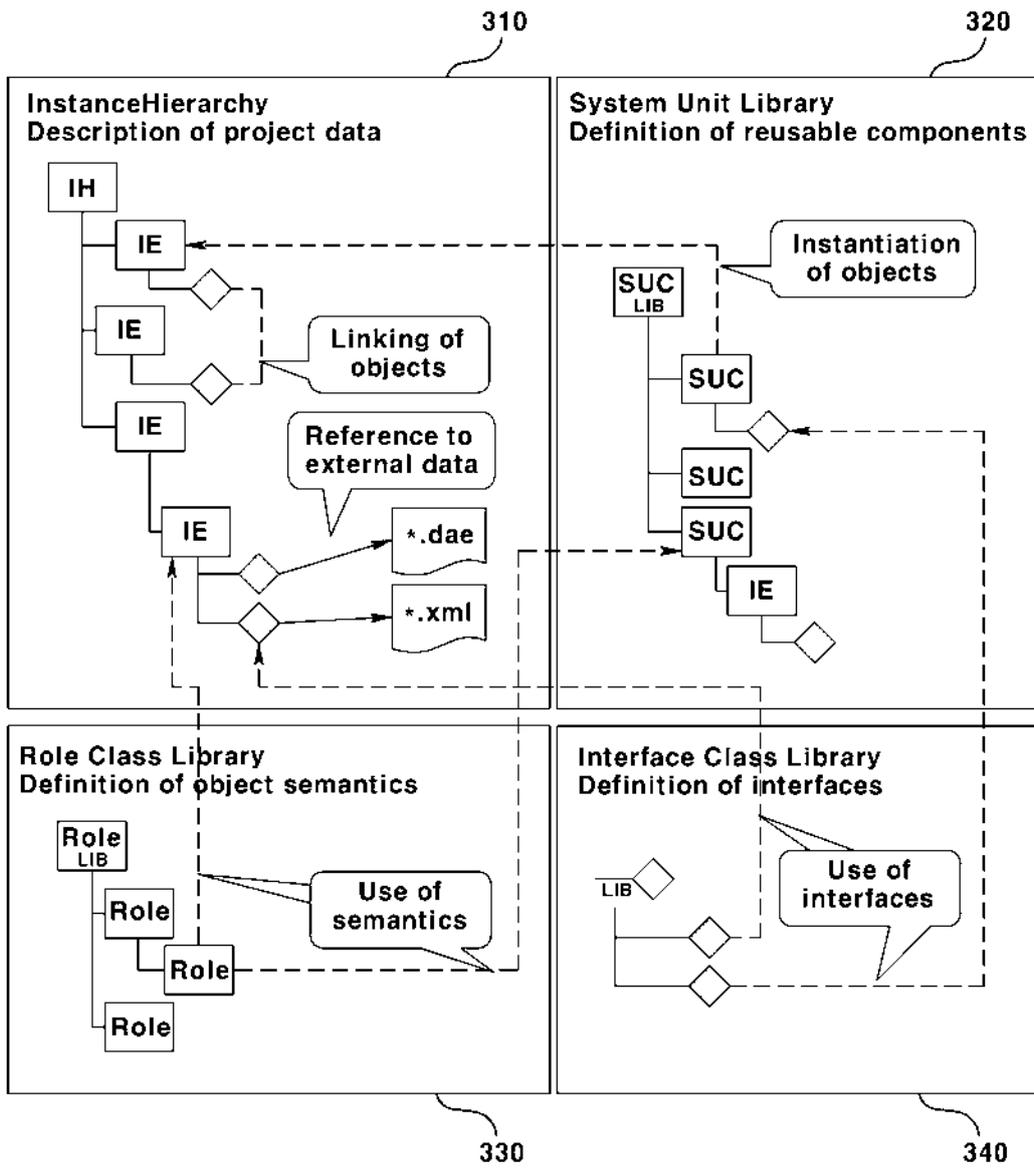
도면1



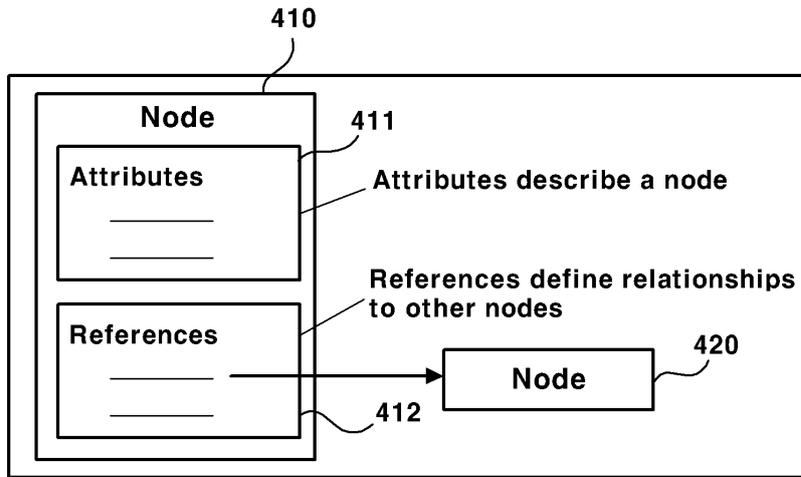
도면2



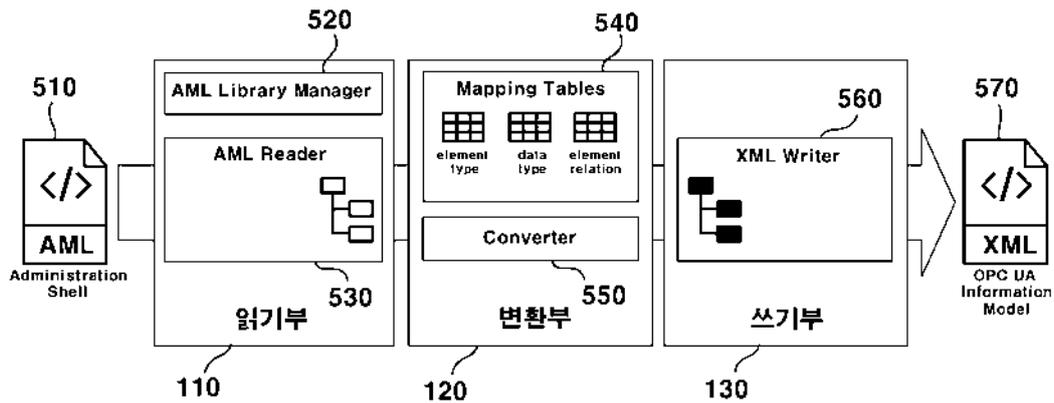
도면3



도면4



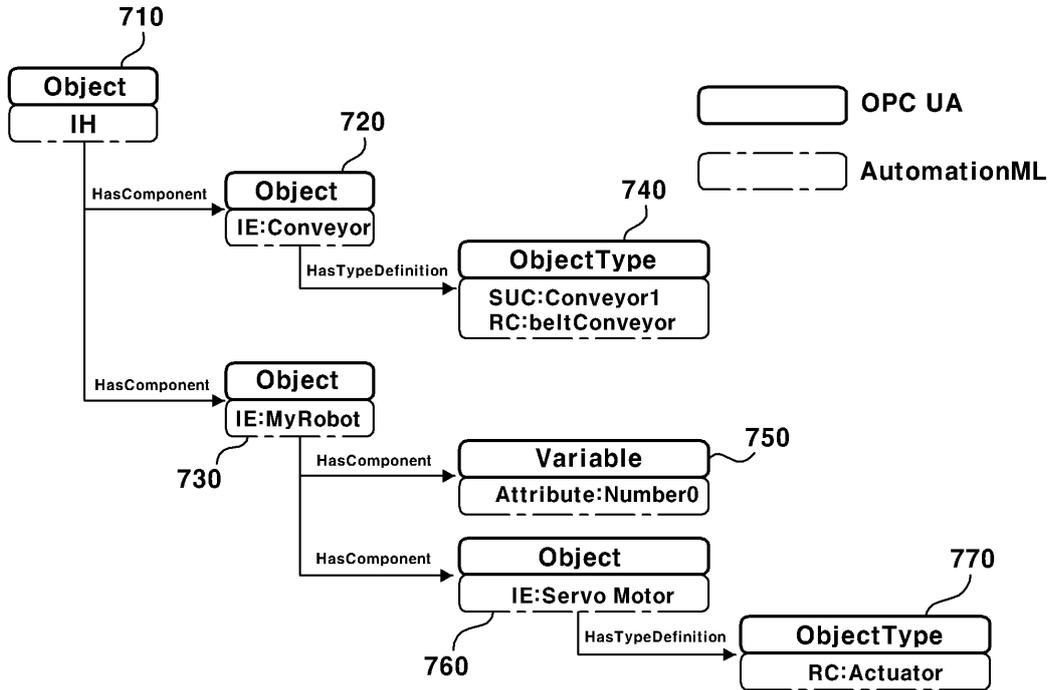
도면5



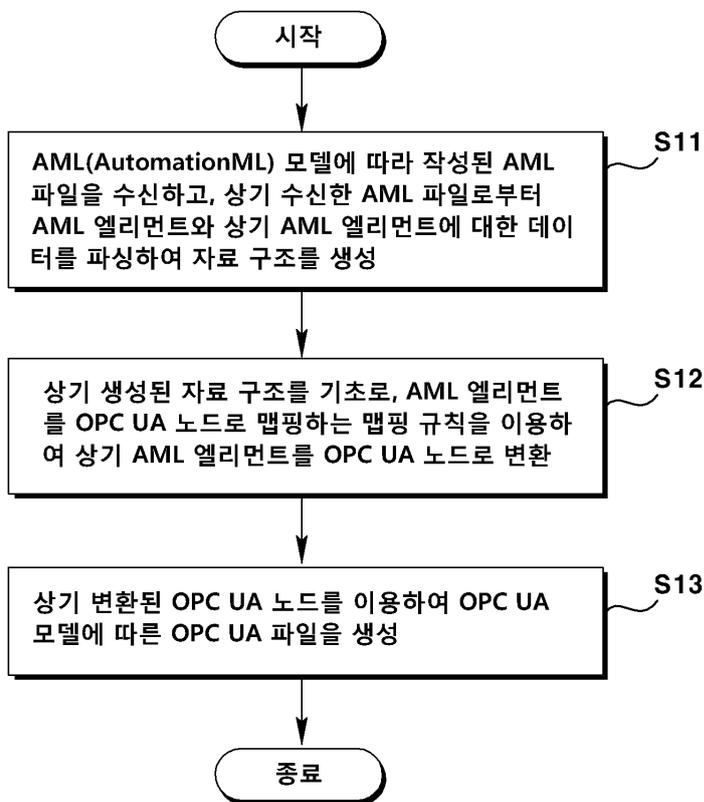
도면6

AML Element	UA Node Type	UA Reference Type	UA Reference Node
CAEXFile	UAObject	HasTypeDefinition	I:CAEXFileType
InstanceHierarchy	UAObject	HasTypeDefinition	FolderType
InternalElement	UAObject	HasTypeDefinition	*RefSystemUnitClassPath
Attribute	UAVariable	HasTypeDefinition	PropertyType
SupportedRoleClass		1:HasAMLSupportedRoleClass	*RefRoleClassPath
RoleRequirements		1:HasAMLRoleRequirement	*RefBaseRoleClassPath
InterfaceClassLib	UAObject	HasTypeDefinition	FolderType
InterfaceClass	UAObjectType	HasSubType	I:CAEXObjectType
RoleClassLib	UAObject	HasTypeDefinition	FolderType
RoleClass	UAObjectType	HasSubType	I:CAEXObjectType
SystemUnitClassLib	UAObject	HasTypeDefinition	FolderType
SystemUnitClass	UAObjectType	HasSubType	I:CAEXObjectType

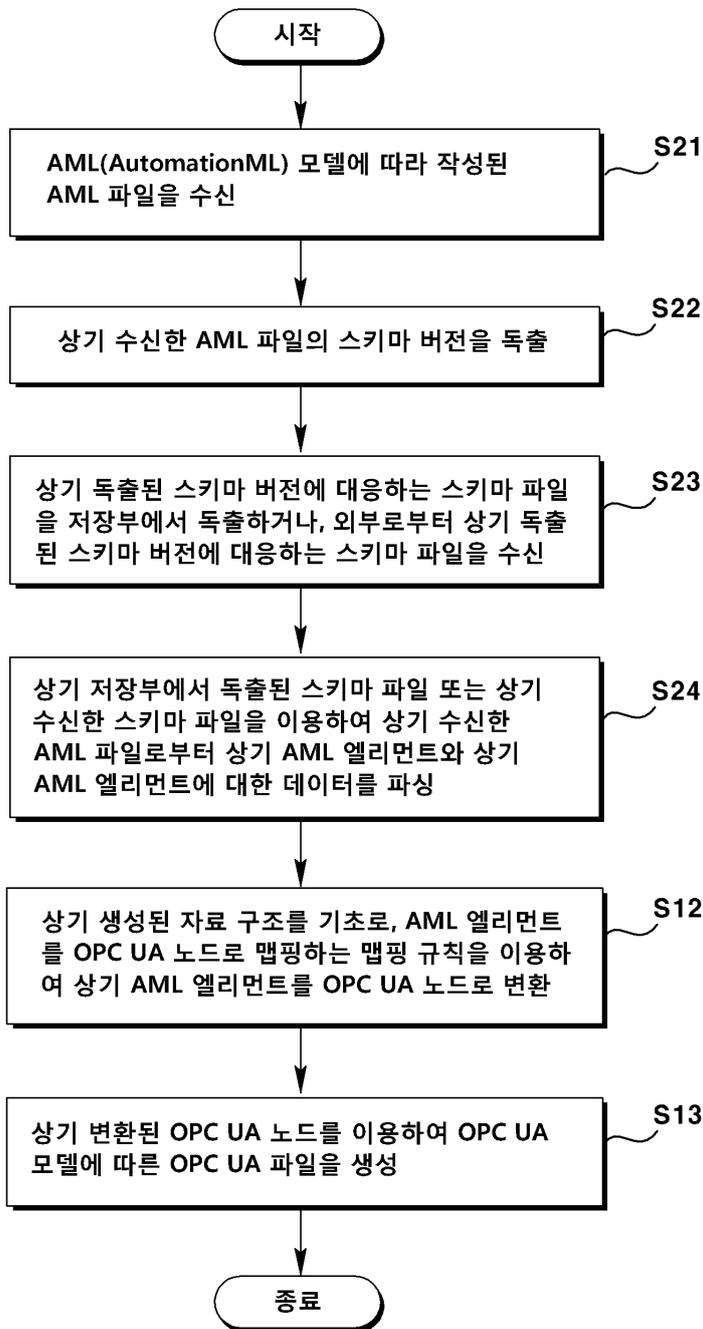
도면7



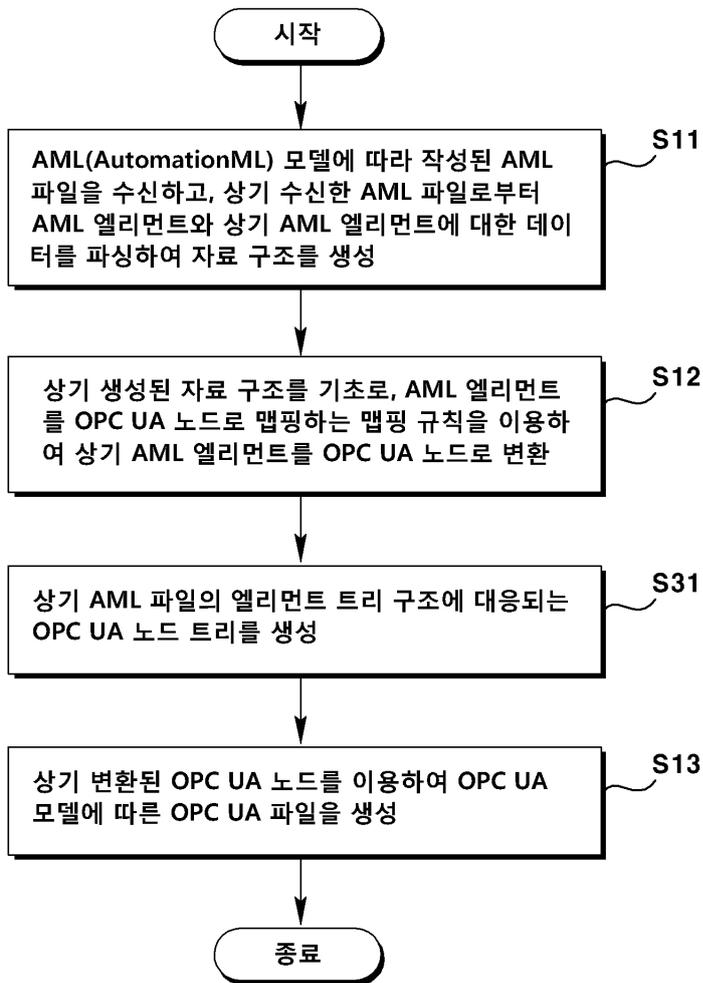
도면8



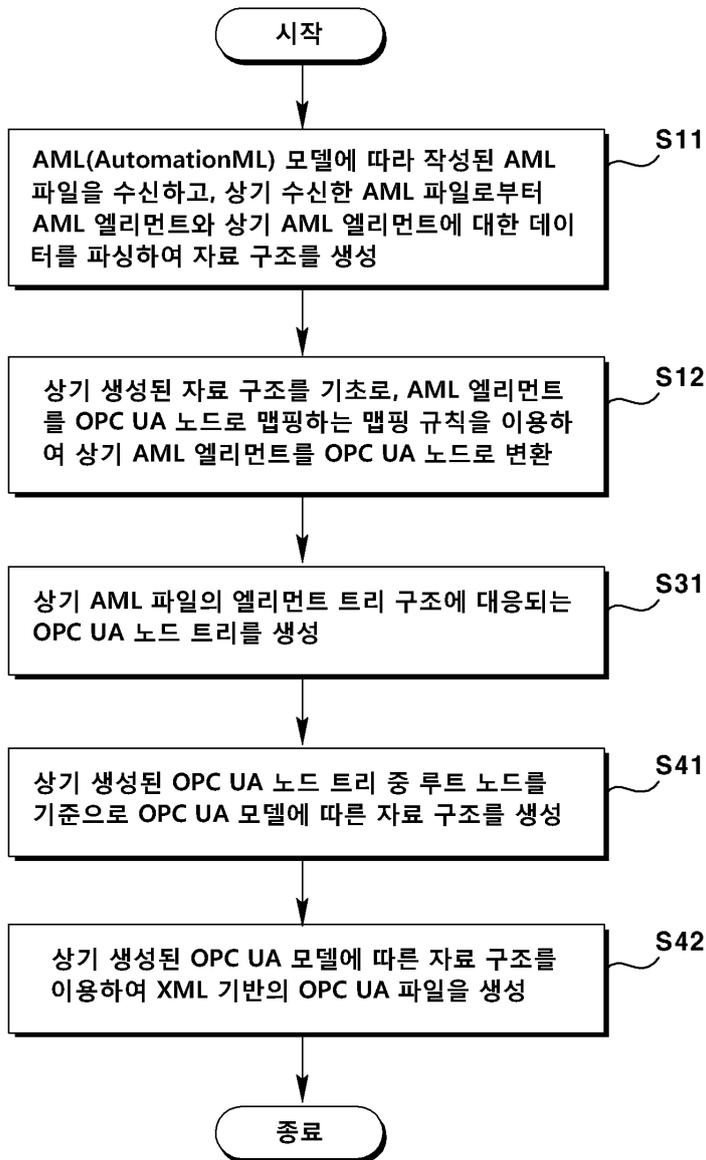
도면9



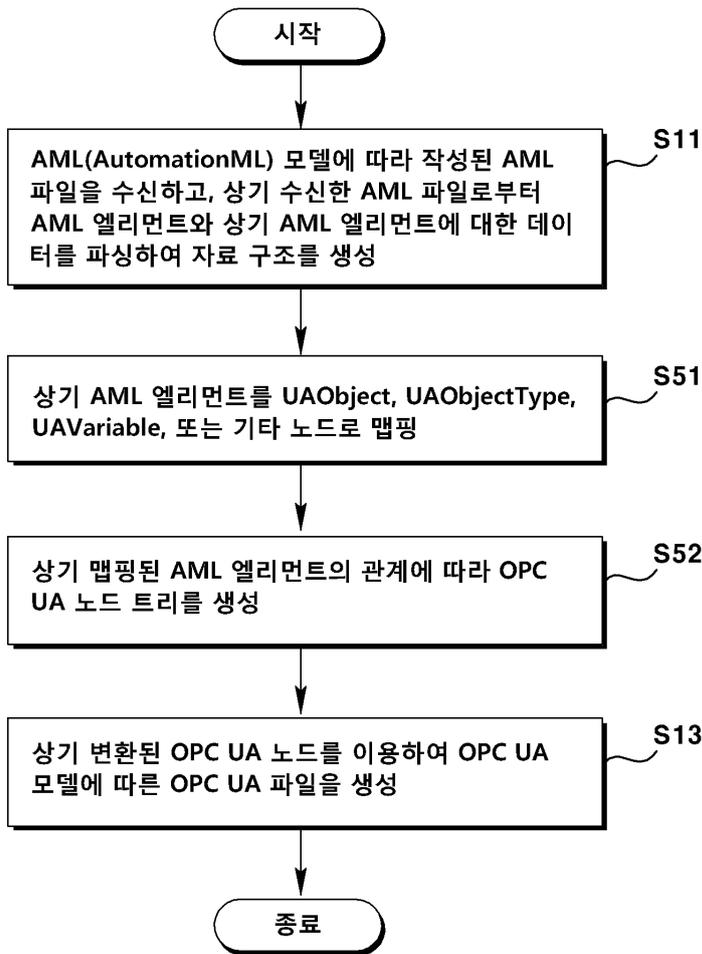
도면10



도면11



도면12



【심사관 직권보정사항】

【직권보정 1】

【보정항목】 청구범위

【보정세부항목】 청구항 1

【변경전】

데이터 모델 변환 장치가 AutomationML 모델에 따라 작성된 AutomationML 파일을 수신하는 단계;

상기 데이터 모델 변환 장치가 상기 수신한 AutomationML 파일의 스키마 버전을 추출하는 단계;

상기 데이터 모델 변환 장치가 외부로부터 상기 추출된 스키마 버전에 대응하는 스키마 파일을 수신하는 단계; 및

상기 데이터 모델 변환 장치가 상기 수신한 스키마 파일을 이용하여, 상기 수신한 AutomationML 파일로부터 AutomationML 엘리먼트와 상기 AutomationML 엘리먼트에 대한 데이터를 파싱하여 자료 구조를 생성하는 단계;

상기 데이터 모델 변환 장치가 상기 생성된 자료 구조를 기초로, AutomationML 엘리먼트를 OPC UA 노드로 맵핑하는 맵핑 규칙을 이용하여 상기 AutomationML 엘리먼트를 OPC UA 노드로 변환하는 단계; 및

상기 데이터 모델 변환 장치가 상기 변환된 OPC UA 노드를 이용하여 OPC UA 모델에 따른 OPC UA 파일을 생성하는 단계를 포함하는 데이터 모델 변환 방법.

【변경후】

데이터 모델 변환 장치가 AutomationML 모델에 따라 작성된 AutomationML 파일을 수신하는 단계;

상기 데이터 모델 변환 장치가 상기 수신한 AutomationML 파일의 스키마 버전을 추출하는 단계;

상기 데이터 모델 변환 장치가 외부로부터 상기 추출된 스키마 버전에 대응하는 스키마 파일을 수신하는 단계;
및

상기 데이터 모델 변환 장치가 상기 수신한 스키마 파일을 이용하여, 상기 수신한 AutomationML 파일로부터 AutomationML 엘리먼트와 상기 AutomationML 엘리먼트에 대한 데이터를 파싱하여 자료 구조를 생성하는 단계;

상기 데이터 모델 변환 장치가 상기 생성된 자료 구조를 기초로, AutomationML 엘리먼트를 OPC UA 노드로 맵핑하는 맵핑 규칙을 이용하여 상기 AutomationML 엘리먼트를 OPC UA 노드로 변환하는 단계; 및

상기 데이터 모델 변환 장치가 상기 변환된 OPC UA 노드를 이용하여 OPC UA 모델에 따른 OPC UA 파일을 생성하는 단계를 포함하는 데이터 모델 변환 방법.



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2019년11월04일
(11) 등록번호 10-2021180
(24) 등록일자 2019년09월05일

(51) 국제특허분류(Int. Cl.)
G05B 23/02 (2006.01) G05B 19/048 (2006.01)
G05B 19/418 (2006.01)
(52) CPC특허분류
G05B 23/02 (2013.01)
G05B 19/048 (2013.01)
(21) 출원번호 10-2017-0088892
(22) 출원일자 2017년07월13일
심사청구일자 2017년09월06일
(65) 공개번호 10-2019-0007647
(43) 공개일자 2019년01월23일
(56) 선행기술조사문헌
JP2007334506 A*
KR100985863 B1*
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
전자부품연구원
경기도 성남시 분당구 새나리로 25 (야탑동)
(72) 발명자
정지은
경기도 성남시 분당구 동판교로 123, 114동 604호
송병훈
경기도 성남시 분당구 동판교로 123, 109동 201호
신준호
경기도 수원시 영통구 영통로200번길 156, 1002동 601호
(74) 대리인
남충우

전체 청구항 수 : 총 8 항

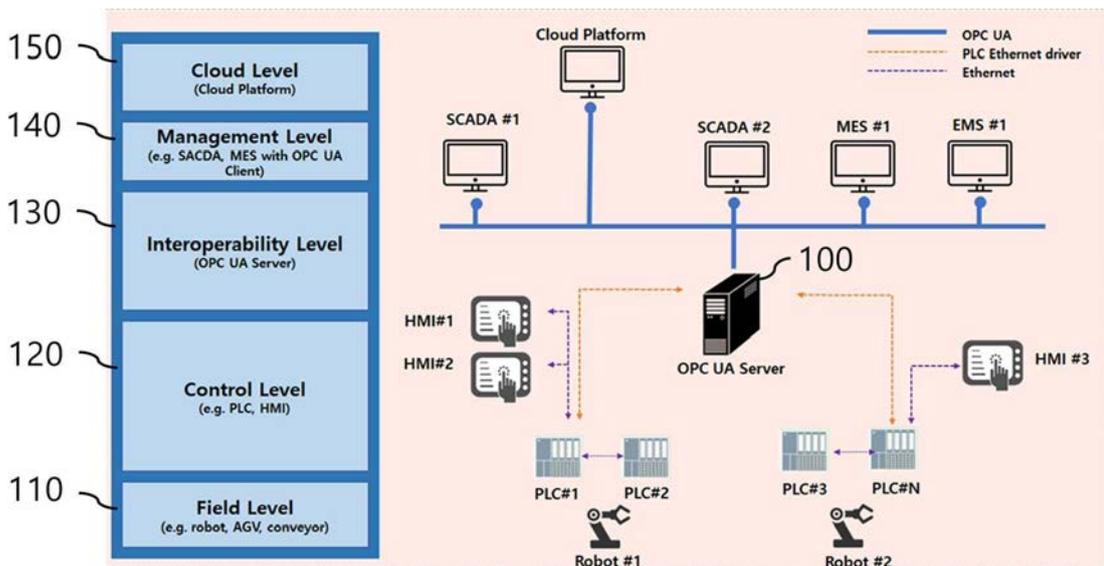
심사관 : 강석제

(54) 발명의 명칭 OPC UA 표준 상호운용성 시험 장치 및 방법

(57) 요약

상호운용성 시험 장치, 방법, 기록매체 및 상호운용성 시험 플랫폼 시스템이 제공된다. 본 상호운용성 시험 장치에 따르면 적어도 하나의 공정 장비에서 생성되는 데이터를 수집하고 공정 장비를 제어하는 적어도 하나의 제어 장비로부터 수집된 데이터를 수신하고, 수신된 데이터를 이용하여 상호운용성을 테스트할 수 있게 되어, 스마트 공장에서 지속적으로 확대되고 있는 다양한 OPC UA 기반의 생산 공정 장비들(예: 로봇, 컨트롤러 등)과 IT 시스템 장비(예: HMI, SCADA, ERP 등), 그외의 응용애플리케이션(예: 웹, 모바일 등) 간에 “상호운용성”을 사전에 검증 테스트해볼 수 있는 기준 플랫폼 및 이에 대한 표준화된 검증 방법을 제공할 수 있게 된다.

대표도 - 도1



(52) CPC특허분류

- G05B 19/418 (2013.01)
- G05B 2219/24015 (2013.01)
- G05B 2219/24039 (2013.01)
- Y02P 90/02 (2015.11)

이 발명을 지원한 국가연구개발사업

과제고유번호	1415147415
부처명	산업통상자원부
연구관리전문기관	한국산업기술평가관리원
연구사업명	스마트공장고도화기술개발
연구과제명	제조산업을 위한 개방형 IIoT 스마트공장 플랫폼 및 Factory-Thing 하드웨어 기술개발
기 여 율	1/1
주관기관	전자부품연구원
연구기간	2016.07.01 ~ 2017.06.30

명세서

청구범위

청구항 1

적어도 하나의 공정 장비에서 생성되는 데이터를 수집하고 공정 장비를 제어하는 적어도 하나의 제어 장비로부터 수집된 데이터를 수신하는 통신부; 및

수신된 데이터를 이용하여 상호운용성을 테스트하는 제어부;를 포함하고,

제어부는,

공정 장비 또는 제어 장비 중 적어도 하나의 신규 장비가 새로 추가된 경우, 추가된 신규 장비와 기존 장비 간의 상호운용성을 테스트하고,

추가된 신규 장비가 상호운용이 불가능한 것으로 판단된 경우, 기존 장비들 중 특정 기존장비와의 연결을 끊은 상태에서 상호운용성 테스트를 다시 수행하고, 해당 상태에서 상호운용이 가능한 것으로 판단되면 신규 장비와 특정 기존장비를 상호운용 불가 장비로 매칭시키며,

기존의 상호운용성 테스트 이력 정보를 이용하여, 신규 장비와 상호운용 불가 장비로 매칭된 특정 기존장비를 대체할 수 있는 장비를 추천하는 것을 특징으로 하는 상호 운용성 시험 장치.

청구항 2

삭제

청구항 3

삭제

청구항 4

삭제

청구항 5

삭제

청구항 6

청구항 1에 있어서,

통신부는,

외부의 공정 장비 또는 제어 장비와 원격 통신을 통해 연결되고,

제어부는,

연결된 외부의 공정 장비 또는 제어 장비와 상호운용성을 테스트하는 것을 특징으로 하는 상호 운용성 시험 장치.

청구항 7

청구항 1에 있어서,

공정 장비 또는 제어 장비는 적어도 하나의 가상 공정 장비 또는 가상 제어 장비를 포함하는 것을 특징으로 하는 상호 운용성 시험 장치.

청구항 8

청구항 7에 있어서,

제어부는,

특정 공정 장비가 상호운용이 불가능한 것으로 판단된 경우, 해당 특정 공정장비를 대체할 수 있고 상호운용이 가능한 가상 공정 장비를 추천하는 것을 특징으로 하는 상호 운용성 시험 장치.

청구항 9

청구항 7에 있어서,

제어부는,

제1 가상 공정 장비가 상호운용이 불가능한 경우, 제1 가상 공정장비를 대체할 수 있고 상호운용이 가능한 제2 가상 공정 장비를 추천하는 것을 특징으로 하는 상호 운용성 시험 장치.

청구항 10

상호 운용성 시험 장치에 의한 상호 운용성 시험 방법에 있어서,

적어도 하나의 공정 장비에서 생성되는 데이터를 수집하고 공정 장비를 제어하는 적어도 하나의 제어 장비로부터 수집된 데이터를 수신하는 단계; 및

수신된 데이터를 이용하여 상호운용성을 테스트하는 단계;를 포함하고,

테스트 단계는,

공정 장비 또는 제어 장비 중 적어도 하나의 신규 장비가 새로 추가된 경우, 추가된 신규 장비와 기존 장비 간의 상호운용성을 테스트하고,

추가된 신규 장비가 상호운용이 불가능한 것으로 판단된 경우, 기존 장비들 중 특정 기존장비와의 연결을 끊은 상태에서 상호운용성 테스트를 다시 수행하고, 해당 상태에서 상호운용이 가능한 것으로 판단되면 신규 장비와 특정 기존장비를 상호운용 불가 장비로 매칭시키는 단계; 및

기존의 상호운용성 테스트 이력 정보를 이용하여, 신규 장비와 상호운용 불가 장비로 매칭된 특정 기존장비를 대체할 수 있는 장비를 추천하는 단계;를 더 포함하는 것을 특징으로 하는 상호 운용성 시험 방법.

청구항 11

상호 운용성 시험 장치에 의한 상호 운용성 시험 방법을 수행하는 컴퓨터 프로그램이 수록된 컴퓨터로 읽을 수 있는 기록매체에 있어서,

적어도 하나의 공정 장비에서 생성되는 데이터를 수집하고 공정 장비를 제어하는 적어도 하나의 제어 장비로부터 수집된 데이터를 수신하는 단계; 및

수신된 데이터를 이용하여 상호운용성을 테스트하는 단계;를 포함하고,

테스트 단계는,

공정 장비 또는 제어 장비 중 적어도 하나의 신규 장비가 새로 추가된 경우, 추가된 신규 장비와 기존 장비 간의 상호운용성을 테스트하고,

추가된 신규 장비가 상호운용이 불가능한 것으로 판단된 경우, 기존 장비들 중 특정 기존장비와의 연결을 끊은 상태에서 상호운용성 테스트를 다시 수행하고, 해당 상태에서 상호운용이 가능한 것으로 판단되면 신규 장비와 특

정 기존장비를 상호운용 불가 장비로 매칭시키는 단계; 및

기존의 상호운용성 테스트 이력 정보를 이용하여, 신규 장비와 상호운용 불가 장비로 매칭된 특정 기존장비를 대체할 수 있는 장비를 추천하는 단계;를 더 포함하는 것을 특징으로 하는 상호 운용성 시험 방법을 수행하는 컴퓨터 프로그램이 수록된 컴퓨터로 읽을 수 있는 기록매체.

청구항 12

적어도 하나의 공정 장비를 포함하는 필드 레벨부;

필드 레벨부의 공정 장비에서 생성되는 데이터를 수집하고 공정 장비를 제어하기 위한 적어도 하나의 제어 장비를 포함하는 제어 레벨부;

제어 레벨부의 제어장비에서 수집된 데이터를 수신하여 상호운용성을 테스트하는 상호 운용성 시험장치를 포함하는 상호운용 레벨부; 및

상호운용 레벨부의 상호 운용성 시험장치로부터 수신된 데이터를 이용하여 필드 레벨부의 공정 장비 및 제어 레벨부의 제어 장비를 관리하는 적어도 하나의 관리 장비를 포함하는 관리 레벨부;를 포함하고,

상호운용 레벨부는,

공정 장비 또는 제어 장비 중 적어도 하나의 신규 장비가 새로 추가된 경우, 추가된 신규 장비와 기존 장비 간의 상호운용성을 테스트하고, 추가된 신규 장비가 상호운용이 불가능한 것으로 판단된 경우, 기존 장비들 중 특정 기존장비와의 연결을 끊은 상태에서 상호운용성 테스트를 다시 수행하고, 해당 상태에서 상호운용이 가능한 것으로 판단되면 신규 장비와 특정 기존장비를 상호운용 불가 장비로 매칭시키며, 기존의 상호운용성 테스트 이력 정보를 이용하여, 신규 장비와 상호운용 불가 장비로 매칭된 특정 기존장비를 대체할 수 있는 장비를 추천하는 상호 운용성 시험 장치를 포함하는 것을 특징으로 하는 상호운용성 시험 플랫폼 시스템.

발명의 설명

기술 분야

[0001] 본 발명은 상호운용성 시험 장치 및 방법에 관한 것으로, 더욱 상세하게는 OPC UA를 이용하여 필드 데이터들을 이용한 상호운용성 시험 장치 및 방법에 관한 것이다.

배경 기술

[0003] 이 부분에 기술된 내용은 단순히 본 실시 예에 대한 배경 정보를 제공할 뿐 종래기술을 구성하는 것은 아니다.

[0004] 스마트 공장은 제조과정에 정보통신기술(ICT, Information and Communications Technologies)을 적용하여 시스템을 최적화하는 개념이다.

[0005] 그 중 산업용 사물인터넷(IIoT, Industrial Internet of Things)은 공정 운영 개선에 중심적인 역할을 하게 될 기술로 주목 받고 있다. 산업용 사물인터넷 기술의 적용을 통해 기존의 제조 기계 및 장비들을 초연결 네트워크로 연결할 수 있으며, 최적화된 제조 생산 체계를 구축할 수 있다.

[0006] 보다 구체적으로, 사물인터넷 기술을 통해 실시간으로 공정 전체에서 막대한 데이터를 수집하고 해석할 수 있게 된다. 또한 해석한 데이터를 기반으로 각각의 공정을 개선할 수 있다. 이러한 과정을 통해 비용 운영이나 자재 관리 등을 효율적으로 할 수 있고, 맞춤형 제품의 생산이 용이해지며, 빅데이터 분석을 통해 시장 변화의 예측성을 높일 수 있다.

[0007] 다만, 초연결 네트워크의 구성을 위해서는 서로 다른 기반시설(infrastructure)로 구성된 공장들 간의 수직적, 수평적 통합을 이룰 필요성이 있다.

[0008] 이러한 통합을 위해서는 다양한 기업에서 공급하고 있는 상이한 장비, 기계, 센서, 시스템 등의 연결성을 향상시키고 운용 정보를 통일화하여 전달해야 한다.

[0009] 그러나 각 제조사는 서로 다른 인터페이스, 플랫폼, 시스템을 사용하여 각 장치들의 상호 호환성을 확보하기 어

려운 것이 현실이다. 또한 인터페이스, 플랫폼 등은 끊임없이 개량되고 변화하므로, 이러한 장비들로부터 추출할 수 있는 로그 데이터도 계속 변경된다.

[0010] 또한, 전통적으로 산업 자동화 솔루션은 장치 사이의 통신에 이용되는 비호환적이고 비상호운용적인 표준에 의해 독립적으로 분리되어 왔다. 그 결과 공장의 생산 장비 및 솔루션들은 독점적 시스템 환경에 의존적인 경우가 많았으며, 솔루션 공급자들은 이러한 환경 지원을 위해 본질적으로 동일한 제품을 여러 가지 버전으로 개발할 수밖에 없었다.

[0011] 또한 인터넷 프로토콜(IP) 기반의 이더넷을 근간으로 하는 다양한 통신 기술(예: 실시간 이더넷, WLAN)이 산업 자동화 현장에 확장되고 있음에도 불구하고, 종종 기술적 차이로 인하여 하위 필드 단계에서 인터넷으로 향하는 정보의 수직적 흐름이 제한되는 경우가 많다. 뿐만 아니라 실제 산업 현장 적용에 있어 다양한 공급업자와 그에 따른 인터페이스간에 표준화된 데이터 교환 방법이 부족한 실정이다.

[0012] 이에 따라, 스마트 공장에서 지속적으로 확대되고 있는 다양한 생산 장비들과 IT 시스템, 그외의 응용애플리케이션 간에 상호운용"을 사전에 검증 테스트해볼 수 있는 기준 플랫폼이 필요하며, 이에 대한 표준화된 검증 방법을 제공하기 위한 방안의 모색이 요청된다.

발명의 내용

해결하려는 과제

[0014] 본 발명은 상기와 같은 문제점을 해결하기 위하여 안출된 것으로서, 본 발명의 목적은, 적어도 하나의 공정 장비에서 생성되는 데이터를 수집하고 공정 장비를 제어하는 적어도 하나의 제어 장비로부터 수집된 데이터를 수신하고, 수신된 데이터를 이용하여 상호운용성을 테스트하는 상호운용성 시험 장치, 방법, 기록매체 및 상호운용성 시험 플랫폼 시스템을 제공함에 있다.

[0015] 본 발명에서 이루고자 하는 기술적 과제들은 이상에서 언급한 기술적 과제들로 제한되지 않으며, 언급하지 않은 또 다른 기술적 과제들은 아래의 기재로부터 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자에게 명확하게 이해될 수 있을 것이다.

과제의 해결 수단

[0017] 상기 목적을 달성하기 위한 본 발명의 일 실시예에 따른, 상호운용성 시험 장치는, 적어도 하나의 공정 장비에서 생성되는 데이터를 수집하고 공정 장비를 제어하는 적어도 하나의 제어 장비로부터 수집된 데이터를 수신하는 통신부; 및 수신된 데이터를 이용하여 상호운용성을 테스트하는 제어부;를 포함한다.

[0018] 그리고, 제어부는, 공정 장비 또는 제어 장비 중 적어도 하나의 신규 장비가 새로 추가된 경우, 추가된 신규 장비와 기존 장비 간의 상호운용성을 테스트할 수도 있다.

[0019] 또한, 제어부는, 추가된 신규 장비가 상호운용이 불가능한 것으로 판단된 경우, 기존의 상호운용성 테스트 이력 정보를 이용하여 해당 신규 장비를 대체할 수 있는 장비를 추천할 수도 있다.

[0020] 그리고, 제어부는, 추가된 신규 장비가 상호운용이 불가능한 것으로 판단된 경우, 기존 장비들 중 특정 기존장비와의 연결을 끊은 상태에서 상호운용성 테스트를 다시 수행하고, 해당 상태에서 상호운용이 가능한 것으로 판단되면 신규 장비와 특정 기존장비를 상호운용 불가 장비로 매칭시킬 수도 있다.

[0021] 또한, 제어부는, 기존의 상호운용성 테스트 이력 정보를 이용하여, 신규 장비와 상호운용 불가 장비로 매칭된 특정 기존장비를 대체할 수 있는 장비를 추천할 수도 있다.

[0022] 그리고, 통신부는, 외부의 공정 장비 또는 제어 장비와 원격 통신을 통해 연결되고, 제어부는, 연결된 외부의 공정 장비 또는 제어 장비와 상호운용성을 테스트할 수도 있다.

[0023] 또한, 공정 장비 또는 제어 장비는 적어도 하나의 가상 공정 장비 또는 가상 제어 장비를 포함할 수도 있다.

[0024] 그리고, 제어부는, 특정 공정 장비가 상호운용이 불가능한 것으로 판단된 경우, 해당 특정 공정장비를 대체할 수 있고 상호운용이 가능한 가상 공정 장비를 추천할 수도 있다.

[0025] 또한, 제어부는, 제1 가상 공정 장비가 상호운용이 불가능한 경우, 제1 가상 공정장비를 대체할 수 있고 상호운용이 가능한 제2 가상 공정 장비를 추천할 수도 있다.

[0026] 한편, 본 발명의 일 실시예에 따른, 상호운용성 시험 방법은, 적어도 하나의 공정 장비에서 생성되는 데이터를 수집하고 공정 장비를 제어하는 적어도 하나의 제어 장비로부터 수집된 데이터를 수신하는 단계; 및 수신된 데이터를 이용하여 상호운용성을 테스트하는 단계;를 포함한다.

[0027] 한편, 본 발명의 일 실시예에 따른, 컴퓨터로 읽을 수 있는 기록매체는, 적어도 하나의 공정 장비에서 생성되는 데이터를 수집하고 공정 장비를 제어하는 적어도 하나의 제어 장비로부터 수집된 데이터를 수신하는 단계; 및 수신된 데이터를 이용하여 상호운용성을 테스트하는 단계;를 포함하는 상호운용성 시험 방법을 수행하는 컴퓨터 프로그램이 수록된다.

[0028] 한편, 본 발명의 일 실시예에 따른, 상호운용성 시험 플랫폼 시스템은, 적어도 하나의 공정 장비를 포함하는 필드 레벨부; 필드 레벨부의 공정 장비에서 생성되는 데이터를 수집하고 공정 장비를 제어하기 위한 적어도 하나의 제어 장비를 포함하는 제어 레벨부; 제어 레벨부의 제어장비에서 수집된 데이터를 수신하여 상호운용성을 테스트하는 상호 운용성 시험장치를 포함하는 상호운용 레벨부; 및 상호운용 레벨부의 상호 운용성 시험장치로부터 수신된 데이터를 이용하여 필드 레벨부의 공정 장비 및 제어 레벨부의 제어 장비를 관리하는 적어도 하나의 관리 장비를 포함하는 관리 레벨부;를 포함한다.

발명의 효과

[0030] 본 발명의 다양한 실시예에 따르면, 본 발명의 목적은, 적어도 하나의 공정 장비에서 생성되는 데이터를 수집하고 공정 장비를 제어하는 적어도 하나의 제어 장비로부터 수집된 데이터를 수신하고, 수신된 데이터를 이용하여 상호운용성을 테스트하는 상호운용성 시험 장치, 방법, 기록매체 및 상호운용성 시험 플랫폼 시스템을 제공할 수 있게 되어, 스마트 공장에서 지속적으로 확대되고 있는 다양한 OPC UA 기반의 생산 공정 장비들(예: 로봇, 컨트롤러 등)과 IT 시스템 장비(예: HMI, SCADA, ERP 등), 그외의 응용애플리케이션(예: 웹, 모바일 등) 간에 “상호운용성”을 사전에 검증 테스트해볼 수 있는 기준 플랫폼 및 이에 대한 표준화된 검증 방법을 제공할 수 있게 된다.

[0031] 본 발명에서 얻을 수 있는 효과는 이상에서 언급한 효과로 제한되지 않으며, 언급하지 않은 또 다른 효과들은 아래의 기재로부터 본 발명이 속하는 기술분야에 서 통상의 지식을 가진 자에게 명확하게 이해될 수 있을 것이다.

도면의 간단한 설명

[0033] 본 발명에 관한 이해를 돕기 위해 상세한 설명의 일부로 포함되는, 첨부 도면은 본 발명에 대한 실시 예를 제공하고, 상세한 설명과 함께 본 발명의 기술적 특징을 설명한다.

- 도 1은 본 발명의 일 실시예에 따른, 상호운용성 시험 플랫폼 시스템의 구조를 도시한 도면,
- 도 2는 본 발명의 일 실시예에 따른, 상호운용 레벨에서의 OPC UA 통신 구조를 도시한 도면,
- 도 3은 본 발명의 일 실시예에 따른, 상호운용성 시험 장치의 구조를 도시한 도면,
- 도 4는 본 발명의 일 실시예에 따른, 가상 장비 및 외부 장비들이 추가된 상호운용성 시험 플랫폼 시스템의 구조를 도시한 도면,
- 도 5는 본 발명의 일 실시예에 따른, 상호운용성 시험 방법을 설명하기 위해 제공되는 흐름도이다.

발명을 실시하기 위한 구체적인 내용

[0034] 본 발명의 과제 해결 수단인 특징 및 이점을 보다 명확히 하기 위하여, 첨부된 도면에 도시된 본 발명의 특정 실시 예를 참조하여 본 발명을 더 상세하게 설명한다.

[0035] 다만, 하기의 설명 및 첨부된 도면에서 본 발명의 요지를 흐릴 수 있는 공지기능 또는 구성에 대한 상세한 설명은 생략한다. 또한, 도면 전체에 걸쳐 동일한 구성 요소들은 가능한 한 동일한 도면 부호로 나타내고 있음에 유의하여야 한다.

[0036] 이하의 설명 및 도면에서 사용된 용어나 단어는 통상적이거나 사전적인 의미로 한정해서 해석되어서는 아니 되며, 발명자는 그 자신의 발명을 가장 최선의 방법으로 설명하기 위한 용어의 개념으로 적절하게 정의할 수 있다는 원칙에 입각하여 본 발명의 기술적 사상에 부합하는 의미와 개념으로 해석되어야만 한다.

- [0037] 따라서 본 명세서에 기재된 실시 예와 도면에 도시된 구성은 본 발명의 가장바람직한 일 실시 예에 불과할 뿐이고, 본 발명의 기술적 사상을 모두 대변하는 것은 아니므로, 본 출원시점에 있어서 이들을 대체할 수 있는 다양한 균등물과 변형예들이 있을 수 있음을 이해하여야 한다.
- [0038] 또한, 제1, 제2 등과 같이 서수를 포함하는 용어는 다양한 구성요소들을 설명하기 위해 사용하는 것으로, 하나의 구성요소를 다른 구성요소로부터 구별하는 목적으로만 사용될 뿐, 상기 구성요소들을 한정하기 위해 사용되지 않는다. 예를 들어, 본 발명의 권리 범위를 벗어나지 않으면서 제2 구성요소는 제1 구성요소로 명명될 수 있고, 유사하게 제1 구성요소도 제2 구성요소로 명명될 수 있다.
- [0039] 더하여, 어떤 구성요소가 다른 구성요소에 "연결되어" 있다거나 "접속되어" 있다고 언급할 경우, 이는 논리적 또는 물리적으로 연결되거나, 접속될 수 있음을 의미한다.
- [0040] 다시 말해, 구성요소가 다른 구성요소에 직접적으로 연결되거나 접속되어 있을 수 있지만, 중간에 다른 구성요소가 존재할 수도 있으며, 간접적으로 연결되거나 접속될 수도 있다고 이해되어야 할 것이다.
- [0041] 또한, 본 명세서에서 기술되는 "포함한다" 또는 "가지다" 등의 용어는 명세서상에 기재된 특징, 숫자, 단계, 동작, 구성요소, 부품 또는 이들을 조합한 것이 존재함을 지정하려는 것이지, 하나 또는 그 이상의 다른 특징들이나 숫자, 단계, 동작, 구성요소, 부품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해되어야 한다.
- [0042] 또한, 명세서에 기재된 "...부", "...기", "모듈" 등의 용어는 적어도 하나의 기능이나 동작을 처리하는 단위를 의미하며, 이는 하드웨어나 소프트웨어 또는 하드웨어 및 소프트웨어의 결합으로 구현될 수 있다.
- [0043] 또한, "일(a 또는 an)", "하나(one)", "그(the)" 및 유사어는 본 발명을 기술하는 문맥에 있어서(특히, 이하의 청구항의 문맥에서) 본 명세서에 달리 지시되거나 문맥에 의해 분명하게 반박되지 않는 한, 단수 및 복수 모두를 포함하는 의미로 사용될 수 있다.
- [0044] 이하에서는 도면을 참조하여 본 발명을 보다 상세하게 설명한다.
- [0045] 도 1은 본 발명의 일 실시예에 따른, 상호운용성 시험 플랫폼 시스템의 구조를 도시한 도면이다. 도 1에 도시된 바와 같이, 상호운용성 시험 플랫폼 시스템은 필드 레벨부(Field Level)(110), 제어 레벨부(Control Level)(120), 상호운용 레벨부(Interoperability Level)(130), 관리 레벨(Management Level)(140), 및 클라우드 레벨(Cloud Level)(150)을 포함한다.
- [0046] 필드 레벨부(110)는 실제로 공정을 수행하는 적어도 하나의 공정 장비를 포함하는 레벨을 나타낸다. 여기에서 공정 장비는 로봇, AGV(Automatic guided vehicles), 컨베이어 등이 될 수 있다. 필드 레벨부(110)는 설치되는 과정이나 동작하는 과정에서 다양한 필드 데이터(예: 제어 위치, 각도, 온도 등)를 생성할 수 있다. 도 1에서, 필드 레벨부(110)는 로봇1(Robot#1) 및 로봇2(Robot#2)를 포함하는 것을 확인할 수 있다.
- [0047] 제어 레벨부(120)는 필드 레벨부(110)의 공정 장비에서 생성되는 데이터를 수집하고 공정 장비를 제어하기 위한 적어도 하나의 제어 장비를 포함하는 레벨을 나타낸다. 여기에서, 제어 장비는 PLC(Programmable Logic Controller) 및 HMI (Human Machine Interface)를 포함한다. PLC에는 다양한 공정 장비가 연결되며, 사용자는 HMI를 이용하여 PLC에 연결된 다양한 공정 장비를 제어할 수 있게 된다. 도 1에서, 제어 레벨부(120)는 PLC#1, PLC#2, PLC#3, PLC#N의 PLC들을 포함하고, HMI#1, HMI#2, HMI#3의 HMI들을 포함하는 것을 확인할 수 있다.
- [0048] 상호운용 레벨부(130)는 제어 레벨부(120)의 제어장비에서 수집된 데이터를 수신하여 상호운용성을 테스트하는 상호 운용성 시험 장치(100)를 포함하는 레벨을 나타낸다. 도 1에서, 상호운용 레벨부(130)는 상호운용성 시험 장치(100)로 OPC UA(OLE(Object Linking and Embedding) for Process Control Unified Architecture) 서버가 포함되어 있는 것을 확인할 수 있다. 상호운용성 시험 장치(100)는 하위 레벨인 필드 레벨부(110) 및 제어 레벨부(120)에서 수집된 데이터들을 OPC UA 표준에 따라 수신하고, 상위 레벨인 관리 레벨부(140) 및 클라우드 레벨부(150)에서 OPC UA 클라이언트를 통해 해당 데이터들에 접근할 수 있도록 해준다. 따라서, 상호운용성 시험 장치(100)는 하위 레벨의 장비들과 상위 레벨의 장비들이 서로 상호운용이 가능한지 여부를 테스트할 수 있게 된다.
- [0049] 관리 레벨부(140)는 상호운용 레벨부(130)의 상호운용성 시험 장치(100)로부터 수신된 데이터를 이용하여 필드 레벨부(110)의 공정 장비 및 제어 레벨부(120)의 제어 장비를 관리하는 적어도 하나의 관리 장비를 포함하는 레벨을 나타낸다. 이를 위해, 관리 레벨부(140)에 포함된 관리 장비들은 OPC UA 클라이언트가 설치되어 있으며, OPC UA 클라이언트를 통해 상호운용성 시험 장치(100)의 데이터에 접근할 수 있게 된다. 예를 들어, 관리 레벨

부(140)는 데이터 모니터링 시스템, MES(Manufacturing Execution System), ERP(Enterprise Resource Planning), EMS(Energy Management System), SCADA(Supervisory Control and Data Acquisition) 등을 포함한다. 도 1에 도시된 바와 같이, 관리 레벨부(140)는 SCADA#1, SCADA#2, MES#1, EMS#1를 포함하는 것을 확인할 수 있다.

[0050] 클라우드 레벨(150)은 상호운용성 시험 장치(100)의 데이터에 접근하는 클라우드 플랫폼 기반의 장비들을 포함하는 레벨을 나타낸다. 클라우드 레벨(150)의 클라우드 플랫폼은 OPC UA 클라이언트가 설치되어 있으며, OPC UA 클라이언트를 통해 상호운용성 시험 장치(100)의 데이터에 접근할 수 있게 된다. 그리고, 클라우드 레벨(150)에 포함된 클라우드 플랫폼 기반의 장비들은 해당 데이터를 수집하여 빅데이터 분석을 수행하게 된다.

[0051] 도 2는 본 발명의 일 실시예에 따른, 상호운용 레벨(130)에서의 OPC UA 통신 구조를 도시한 도면이다.

[0052] 도 2에 도시된 바와 같이, 상호운용성 시험 장치(100)는 OPC UA 서버 기능을 수행하고, 제어 레벨부(120)에 해당되는 HMI, 관리 레벨부(140)에 해당되는 MES/SCADA, 및 클라우드 레벨부(150)에 해당되는 Cloud Application은 모두 OPC UA 클라이언트가 설치되어, OPC UA 클라이언트를 통해 상호운용성 시험 장치(100)에 접근하는 것을 확인할 수 있다. 이와 같이, 상호운용성 시험 장치(100)는 OPC UA 서버기능을 통하여 로봇, PLC, HMI, SCADA 및 이외의 다양한 애플리케이션 간 정보 공유가 가능하다. 그리고, 상호운용성 시험장치(100)가 특정 장비에 대해 상호운용이 가능하도록 제작되면, OPC UA 클라이언트를 통해 접근할 수 있는 모든 애플리케이션이나 장비들에 의해서도 상호운용 가능하도록 사용될 수 있게 된다. 즉, 상호운용성 시험 장치(100)는 OPC UA 표준을 이용하여 상호운용성을 향상시킬 수 있게 된다.

[0053] 이와 같은, 상호운용성 시험 장치(100)는 사용자 인터페이스를 제공한다. 상호운용성 시험 장치(100)는 사용자 인터페이스를 이용해서 공정 데이터를 열람하고 응용과 목적에 맞는 데이터를 사용할 수 있도록 지원해준다. 또한, 상호운용성 시험 장치(100)는 데이터의 도식으로 그래프나 적절한 시각적 환경 사용자의 요구에 맞게 구성할 수 있으며 인터페이스는 별도의 소프트웨어의 설치가 요구되지 않는 웹 브라우저를 사용하여 지원할 수도 있다. 그리고, 상호운용성 시험 장치(100)는 생산 전략에 기반을 둔 제조 여건 변화에 대응할 수 있도록 지원하며, 생산 환경의 동적 변화에 보다 유연하게 대응할 수 있도록 PLC와의 실시간 인터페이스 지원하게 된다.

[0054] 또한, 상호운용성 시험 장치(100)는 암호화 소프트웨어 모듈을 포함할 수도 있다. 구체적으로, 상호운용성 시험 장치(100)는 OPC UA 클라이언트와 서버 구간이 암호화를 통한 보안 서비스를 제공한다. 이를 통해, 상호운용성 시험 장치(100)는 모든 자동화 공정에서 발생하는 데이터통신 및 정보의 흐름을 암호화하며 암호화 된 정보는 공정 데이터의 관리 주체만이 열람되고, 가공 될 수 있도록 지원하게 된다.

[0055] 그리고, 상호운용성 시험 장치(100)는 커넥션 모듈을 포함할 수도 있다. 커넥션 모듈은 인증된 장치로 암호화 데이터베이스에 접근하고 공정정보를 이동시키는 대리자 역할을 하는 모듈로 암호화 소프트웨어 모듈을 사용하여 데이터베이스를 액세스하는 역할을 수행한다. 그리고, 상호운용성 시험 장치(100)는 커넥션 모듈을 통해 데이터를 발생시키는 장치와 지정된 통신 프로토콜을 이용하여 구성에 맞게 동작 및 연동하여 공정 데이터를 실제로 수집하여 가공하게 된다.

[0056] 또한, 상호운용성 시험 장치(100)는 데이터 관리자 모듈을 포함할 수도 있다. 데이터 관리자 모듈은 사용자가 원하는 데이터 통신방법 및 데이터 소스와 화면을 필요에 따라 변경하고, 구성 및 확장할 수 있는 기능을 지원한다.

[0057] 그리고, 상호운용성 시험 장치(100)는 고속 대용량 데이터베이스를 포함할 수도 있다. 상호운용성 시험 장치(100)는 고속 대용량 데이터베이스를 통해 공정데이터나 데이터 관리자 모듈에서 구성 된 데이터는 암호화 방법과 정책을 통해 관리되며 지역적으로 분산되어 저장되고 장애에 의한 데이터 손실과 커넥션 하드웨어 모듈의 연결에 지역적 제약이 없도록 지원할 수 있게 된다.

[0058] 이와 같이, 상호운용성 시험 플랫폼 시스템은 총 5개의 레벨로 구성될 수 있다. 단, 클라우드 레벨(150)은 포함되지 않을 수도 있으므로, 상호운용성 시험 플랫폼 시스템은 총 4개의 레벨 또는 5개의 레벨로 구성될 수 있다. 이와 같이 구성된 상호운용성 시험 플랫폼 시스템은 어떤 장비가 공장의 다른 장비들과 서로 상호운용이 가능한 것인지 여부를 테스트할 수 있는 환경을 제공할 수 있게 된다.

[0059] 이하에서는 상호운용성 시험장치의 동작에 대해 상세히 설명한다.

[0060] 도 3은 본 발명의 일 실시예에 따른, 상호운용성 시험 장치(100)의 구조를 도시한 도면이다. 도 3에 도시된 바

와 같이, 상호운용성 시험 장치(100)는 통신부(101) 및 제어부(102)를 포함한다.

- [0061] 통신부(101)는 적어도 하나의 공정 장비에서 생성되는 데이터를 수집하고 공정 장비를 제어하는 적어도 하나의 제어 장비로부터 수집된 데이터를 수신한다. 구체적으로, 통신부(101)는 제어레벨(120)의 제어장비로부터 데이터를 수신한다. 여기에서, 데이터는 필드 레벨(110)에 포함된 공정장비들로부터 수집된 공정이나 장비들의 상태나 상황을 나타내는 필드 데이터를 나타내며, 예를 들어, 설치되는 과정이나 동작하는 과정에서 생성되는 다양한 필드 데이터(예: 제어 위치, 각도, 온도 등)를 나타낸다. 통신부(101)는 이더넷(Ethernet), 내부 네트워크, 인터넷, 이동통신 등의 다양한 통신 방식을 이용하여 데이터를 수신하게 된다.
- [0062] 또한, 통신부(101)는 상위 레벨인 관리 레벨부(140) 및 클라우드 레벨부(150)에서 OPC UA 클라이언트를 통해 해당 데이터들에 접근할 수 있도록, 관리 레벨부(140) 및 클라우드 레벨부(150)의 장비들과 통신을 하게 된다.
- [0063] 제어부(102)는 상호운용성 시험 장치(100)의 전반적인 동작을 제어한다. 구체적으로, 제어부(102)는 수신된 데이터를 이용하여 상호운용성을 테스트한다. 제어부(102)는 제어부(102)는 상호운용성 시험 플랫폼 시스템에 포함된 모든 장비들이 서로 상호운용이 가능한지 여부를 판단하고, 상호운용이 불가능한 경우 어떤 장비가 상호운용이 불가능한 장비인지를 메시지나 화면 UI 등을 통해 사용자에게 알려주게 된다.
- [0064] 구체적으로, 제어부(102)는 공정 장비 또는 제어 장비 중 적어도 하나의 신규 장비가 새로 추가된 경우, 추가된 신규 장비와 기존 장비 간의 상호운용성을 테스트하게 된다. 제어부(102)는 추가된 신규 장비에 대한 데이터를 기존 장비에 입력하여 정상적으로 인식이 되는지 여부를 확인함으로써, 신규 장비의 상호운용성을 테스트할 수 있게 된다. 예를 들어, 도 1의 상호운용성 시험 플랫폼 시스템에서 Robot#1이 새로 추가된 경우, 제어부(102)는 신규추가된 Robot#1이 기존장비인 PLC#1, PLC#2, HMI#1, HMI#2와 서로 상호운용이 가능한지 여부를 테스트하게 된다.
- [0065] 그리고, 제어부(102)는 추가된 신규 장비가 상호운용이 불가능한 것으로 판단된 경우, 기존의 상호운용성 테스트 이력 정보를 이용하여 해당 신규 장비를 대체할 수 있는 장비를 추천하게 된다. 예를 들어, 신규추가된 Robot#1이 상호운용이 불가능한 것으로 판단된 경우, 제어부(102)는 기존의 상호운용성 테스트 이력 정보를 이용하여 Robot#1과 동일한 기능을 하며 이미 상호운용성 시험 플랫폼 시스템의 상호운용이 가능한 것으로 판단되어 사용 중인 Robot#2와 동일한 모델 제품을 대체 공정장비로 추천할 수도 있다.
- [0066] 또한, 제어부(102)는 추가된 신규 장비가 상호운용이 불가능한 것으로 판단된 경우, 기존 장비들 중 특정 기존 장비와의 연결을 끊은 상태에서 상호운용성 테스트를 다시 수행하고, 해당 상태에서 상호운용이 가능한 것으로 판단되면 신규 장비와 특정 기존장비를 상호운용 불가 장비로 매칭시킬 수도 있다. 예를 들어, 제어부(102)는 Robot#1이 상호운용이 불가능한 것으로 판단된 경우, PLC#1, PLC#2, HMI#1, 및 HMI#2를 각각 하나씩 연결을 끊은 상태에서 상호운용성 테스트를 다시 수행한다. 만약 PLC#2의 연결을 끊은 상태일때 신규 장비인 Robot#1은 상호운용이 가능한 것으로 판단되었다면, 제어부(102)는 신규장비 Robot#1와 기존장비 중 PLC#2를 서로 상호운용이 불가능한 장비로 매칭시키게 된다.
- [0067] 이 경우, 제어부(102)는 기존의 상호운용성 테스트 이력 정보를 이용하여, 신규 장비와 상호운용 불가 장비로 매칭된 특정 기존장비를 대체할 수 있는 장비를 추천할 수도 있다. 예를 들어, 신규장비 Robot#1과 기존장비 중 PLC#2를 서로 상호운용이 불가능한 장비로 매칭된 경우, 제어부(102)는 기존의 상호운용성 테스트 이력 정보를 이용하여, PLC#2를 대체할 수 있는 장비를 추천할 수도 있게 된다.
- [0068] 이와 같이, 제어부(102)는 상호운용성 테스트를 통해 신규장비가 상호운용이 불가능한 것으로 판단된 경우 이를 대체할 수 있는 장비를 추천해주게 된다.
- [0069] 한편, 상호운용성 시험 플랫폼 시스템에는 외부 장비가 원격으로 연결되어 상호운용성 테스트가 수행될 수도 있고, 가상 장비가 포함될 수도 있다. 이와 같은 환경의 상호운용성 시험 플랫폼 시스템은 도 4에 도시되어 있다. 도 4는 본 발명의 일 실시예에 따른, 가상 장비 및 외부 장비들이 추가된 상호운용성 시험 플랫폼 시스템의 구조를 도시한 도면이다.
- [0070] 일단, 통신부(101)는 외부의 공정 장비 또는 제어 장비와 원격 통신을 통해 연결된다. 도 4에 따르면, 외부 장비(410)는 상호운용성 시험 장치(100)의 통신부(101)와 통신 가능하도록 원격으로 연결되고, 외부 장비(410)는 외부PLC#1, 외부PLC#2, 및 외부Robot#1이 포함되어 있는 것을 확인할 수 있다.
- [0071] 그리고, 제어부(102)는 연결된 외부의 공정 장비 또는 제어 장비와 상호운용성을 테스트하게 된다. 예를 들어, 제어부(102)는 외부 장비(410)의 외부PLC#1, 외부PLC#2, 및 외부Robot#1가 기존 장비인 Robot#1, PLC#1,

PLC#2, HMI#1, HMI#2 등과 상호운용이 가능한지 여부를 원격으로 테스트하게 된다.

- [0072] 이를 통해, 사용자는 자신의 장비를 직접 가져오지 않고, 원격으로 다른 장비들과의 상호운용성을 쉽게 테스트해볼 수 있게 된다.
- [0073] 또한, 도 4에 도시된 바와 같이, 상호운용성 시험 플랫폼 시스템의 필드 레벨부(110) 및 제어 레벨부(120)는 적어도 하나의 가상 공정 장비 또는 가상 제어 장비를 포함할 수도 있다. 여기에서, 가상 공정 장비 또는 가상 제어장비는 물리적 장비와 동일하게 인식되도록 제어부(102)에 의해서 만들어진 가상 장비이다. 도 4에서는 Virtual Robot#2(420)와 Virtual PLC#3(430)이 각각 가상 공정 장비 및 가상 제어장비인 것을 확인할 수 있다.
- [0074] 제어부(102)는 특정 공정 장비가 상호운용이 불가능한 것으로 판단된 경우, 해당 특정 공정장비를 대체할 수 있고 상호운용이 가능한 가상 공정 장비를 추천해줄 수 있다.
- [0075] 또한, 제어부(102)는 제1 가상 공정 장비가 상호운용이 불가능한 경우, 제1 가상 공정장비를 대체할 수 있고 상호운용이 가능한 제2 가상 공정 장비를 추천할 수도 있다.
- [0076] 이와 같이, 가상 공정 장비 및 가상 제어 장비를 이용하여 상호운용성을 테스트할 수 있게 됨으로써, 사용자는 장비를 구입하여 직접 연결해보지 않더라도 상호운용성을 가상으로 테스트해볼 수 있게 된다.
- [0077] 이하에서는, 도 5를 참고하여, 상호운용성 시험 방법에 대해 설명한다. 도 5는 본 발명의 일 실시예에 따른, 상호운용성 시험 방법을 설명하기 위해 제공되는 흐름도이다.
- [0078] 먼저, 상호운용성 시험 장치(100)는 적어도 하나의 공정 장비에서 생성되는 데이터를 수집하고 공정 장비를 제어하는 적어도 하나의 제어 장비로부터 수집된 데이터를 수신한다(S210).
- [0079] 그리고, 상호운용성 시험 장치(100)는 수신된 데이터를 이용하여 상호운용성을 테스트하게 된다(S220).
- [0080] 이외에 상호운용성 시험 장치(100)는 상술한 제어부(102)의 기능을 통해 상호운용성 시험 방법을 수행하게 된다.
- [0081] 한편, 본 실시예에 따른 상호운용성 시험 장치(100)의 기능 및 방법을 수행하게 하는 컴퓨터 프로그램을 수록한 컴퓨터로 읽을 수 있는 기록매체에도 본 발명의 기술적 사상이 적용될 수 있음은 물론이다. 또한, 본 발명의 다양한 실시예에 따른 기술적 사상은 컴퓨터로 읽을 수 있는 기록매체에 기록된 컴퓨터로 읽을 수 있는 프로그래밍 언어 코드 형태로 구현될 수도 있다. 컴퓨터로 읽을 수 있는 기록매체는 컴퓨터에 의해 읽을 수 있고 데이터를 저장할 수 있는 어떤 데이터 저장 장치이더라도 가능하다. 예를 들어, 컴퓨터로 읽을 수 있는 기록매체는 ROM, RAM, CD-ROM, 자기 테이프, 플로피 디스크, 광디스크, 하드 디스크 드라이브, 플래시 메모리, 솔리드 스테이트 디스크(SSD) 등이 될 수 있음은 물론이다. 또한, 컴퓨터로 읽을 수 있는 기록매체에 저장된 컴퓨터로 읽을 수 있는 코드 또는 프로그램은 컴퓨터간에 연결된 네트워크를 통해 전송될 수도 있다.
- [0082] 본 명세서와 도면에서는 예시적인 장치 구성을 기술하고 있지만, 본 명세서에서 설명하는 기능적인 동작과 주제의 구현물은 다른 유형의 디지털 전자 회로로 구현되거나, 본 명세서에서 개시하는 구조 및 그 구조적인 등가물들을 포함하는 컴퓨터 소프트웨어, 펌웨어 혹은 하드웨어로 구현되거나, 이들 중 하나 이상의 결합으로 구현 가능하다.
- [0083] 따라서, 상술한 예를 참조하여 본 발명을 상세하게 설명하였지만, 본 발명이 속하는 분야의 통상의 기술자라면 본 발명의 범위를 벗어나지 않으면서도 본 예들에 대한 개조, 변경 및 변형을 가할 수 있다.
- [0084] 또한, 이상에서는 본 발명의 바람직한 실시예에 대하여 도시하고 설명하였지만, 본 발명은 상술한 특징의 실시예에 한정되지 아니하며, 청구범위에서 청구하는 본 발명의 요지를 벗어남이 없이 당해 발명이 속하는 기술분야에서 통상의 지식을 가진자에 의해 다양한 변형실시가 가능한 것은 물론이고, 이러한 변형실시들은 본 발명의 기술적 사상이나 전망으로부터 개별적으로 이해되어서는 안될 것이다.

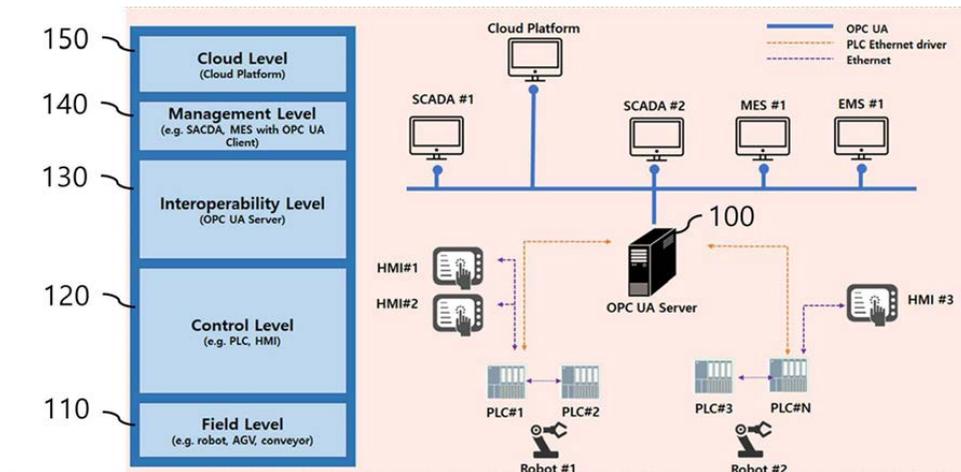
부호의 설명

- [0086] 100 : 상호운용성 시험 장치
- 101 : 통신부
- 102 : 제어부
- 110 : 필드 레벨부

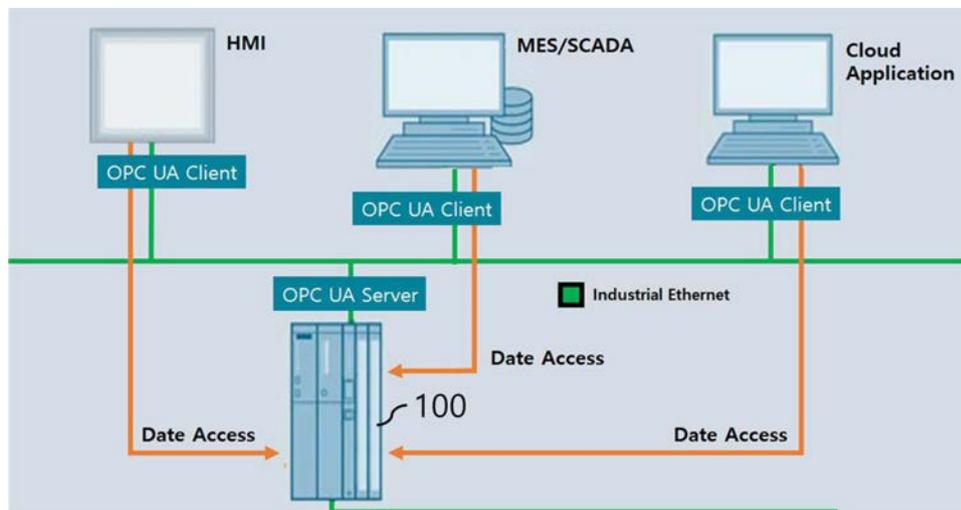
- 120 : 제어 레벨부
- 130 : 상호운용 레벨부
- 140 : 관리 레벨부
- 150 : 클라우드 레벨부

도면

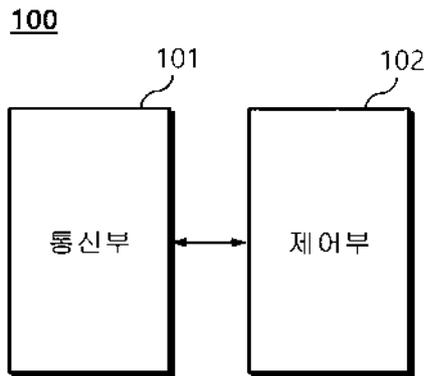
도면1



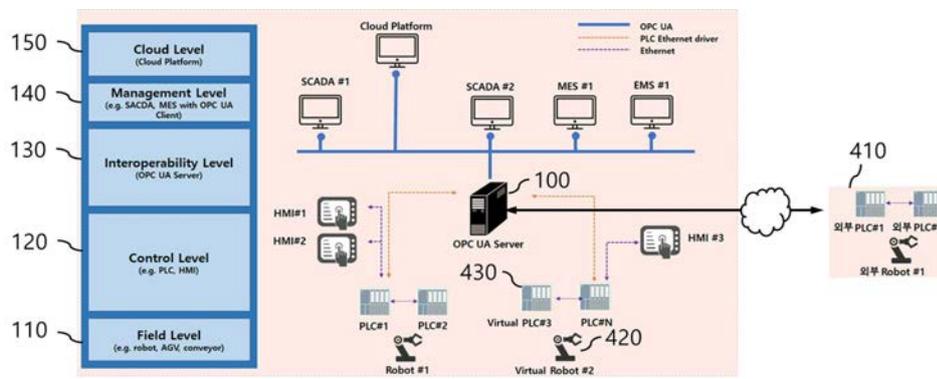
도면2



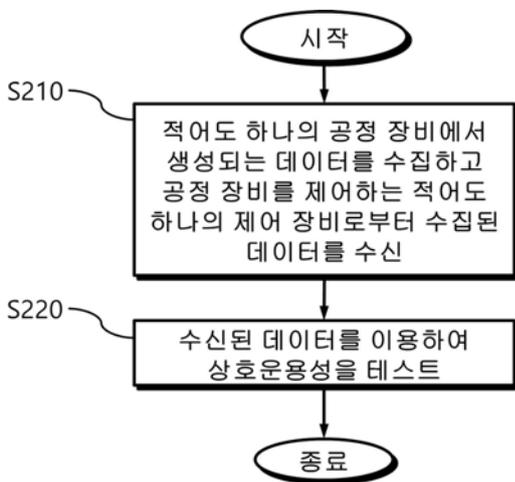
도면3



도면4



도면5



【심사관 직권보정사항】

【직권보정 1】

【보정항목】 청구범위

【보정세부항목】 청구항 11의 첫번째줄

【변경전】

상호 운용성 시험 장치에 의한 상호 운용성 시험 방법에 있어서,

【변경후】

상호 운용성 시험 장치에 의한 상호 운용성 시험 방법을 수행하는 컴퓨터 프로그램이 수록된 컴퓨터로 읽을 수 있는 기록매체에 있어서,



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2020-0065475
(43) 공개일자 2020년06월09일

(51) 국제특허분류(Int. Cl.)
G06Q 50/04 (2012.01) G05B 19/418 (2006.01)
G06Q 10/06 (2012.01)

(52) CPC특허분류
G06Q 50/04 (2013.01)
G05B 19/4183 (2013.01)

(21) 출원번호 10-2018-0151995
(22) 출원일자 2018년11월30일
심사청구일자 2019년02월13일

(71) 출원인
전자부품연구원
경기도 성남시 분당구 새나리로 25 (야탑동)

(72) 발명자
남승욱
서울특별시 마포구 신촌로 150, 907호
송병훈
경기도 성남시 분당구 동판교로 123, 109동 201호
신준호
경기도 수원시 영통구 영통로200번길 156, 1002동 601호

(74) 대리인
남충우

전체 청구항 수 : 총 12 항

(54) 발명의 명칭 관리-셀을 이용한 모듈 조합형 자동화 공정장비 관리 방법

(57) 요약

관리-셀을 이용한 모듈 조합형 자동화 공정장비 관리 방법이 제공된다. 본 발명의 실시예에 따른 관리-셀 기반 공정장비 관리 방법은, 공정 장비들의 정보들을 관리-셀들로 표현하고, 관리-셀들을 이용하여 공정 장비들에 접속하고 통신하며, 관리-셀은 관리-셀 헤더와 관리-셀 바디를 포함하고, 관리-셀 헤더는 공정 장비의 Asset ID 및 관리-셀 ID를 포함하며, 관리-셀 바디는 공정 장비의 Hardware Configuration 정보, Operating Data 및 Management Data를 포함한다. 이에 의해, 정보 모델 표준화를 통한 초기 엔지니어링 및 시운전을 위한 노동력 및 시간 감축이 가능하고, 제조라인의 유연성을 확대하여 제조사 독립적인 상호운용을 가능하게 하며, 기존 산업 솔루션 영역을 넘어 다양한 서비스 플랫폼과의 확장 연계가 가능해진다.

대표도 - 도7

InstanceHierarchy		
[+] [X] [Home] [Refresh] Search ...		
<ul style="list-style-type: none"> ▲ ShellHeader <ul style="list-style-type: none"> ▶ [IE] SystemID ▶ [IE] ShellID 	관리-셀 헤더(Header) System ID, Shell ID	관리-셀 헤더(Header)
<ul style="list-style-type: none"> ▲ APC Sample Instance Hierarchy <ul style="list-style-type: none"> ▶ [IE] AML_TEST {Role: AutomationProject} 	관리-셀 바디1(Body1) Hardware Configuration	관리-셀 바디(Body)
<ul style="list-style-type: none"> ▲ I4.0Factory <ul style="list-style-type: none"> ▶ [IE] Pankyo {Role: Area} 	관리-셀 바디2(Body2) Operating Data	
<ul style="list-style-type: none"> ▲ I4.0Management <ul style="list-style-type: none"> ▶ [IE] Field Station Management ▶ [IE] RawMaterialManagement ▶ [IE] EnergyResourceManagemet 	관리-셀 바디3(Body3) Management Data	

(52) CPC특허분류

G05B 19/4185 (2013.01)

G06Q 10/063 (2013.01)

이 발명을 지원한 국가연구개발사업

과제고유번호 1415158451

부처명 산업통상자원부

연구관리전문기관 한국산업기술평가관리원

연구사업명 전자부품산업핵심기술개발

연구과제명 전자 디바이스 모듈화 장비 가상화를 위한 CPS 연동 및 검증 기술 개발

기 여 율 1/1

주관기관 전자부품연구원

연구기간 2018.04.01 ~ 2018.12.31

명세서

청구범위

청구항 1

제1 서버들이, 공정 장비들의 정보들을 관리-셀들로 표현하는 단계;
제2 서버가, 제1 서버들에 의해 표현된 관리-셀들을 이용하여 공정 장비들에 접속하고 통신하는 단계;를 포함하고,
관리-셀은,
관리-셀 헤더와 관리-셀 바디를 포함하고,
관리-셀 헤더는,
공정 장비의 Asset ID 및 관리-셀 ID를 포함하며,
관리-셀 바디는,
공정 장비의 Hardware Configuration 정보, Operating Data 및 Management Data를 포함하는 것을 특징으로 하는 관리-셀 기반 공정장비 관리 방법.

청구항 2

청구항 1에 있어서,
관리-셀은,
공정 장비들 간 상호 운용, 공정 장비와 상위 애플리케이션 간 상호 운용, 상위 애플리케이션들 간 상호 운용을 위한 정보 체계를 갖는 것을 특징으로 하는 관리-셀 기반 공정장비 관리 방법.

청구항 3

청구항 2에 있어서,
제2 서버는,
제1 서버들에 의해 표현된 관리-셀들을 이용하여 생성되는 것을 특징으로 하는 관리-셀 기반 공정장비 관리 방법.

청구항 4

청구항 3에 있어서,
제2 서버는,
관리-셀들로부터 Node 정보를 추출하고, 추출한 Node 정보를 참조로 공정 장비의 데이터를 획득하고, 획득한 데이터를 다른 공정 장비의 데이터와 교환하는 것을 특징으로 하는 관리-셀 기반 공정장비 관리 방법.

청구항 5

청구항 4에 있어서,
제2 서버는,

획득한 데이터를 상위 애플리케이션에 전달하는 것을 특징으로 하는 관리-셀 기반 공정장비 관리 방법.

청구항 6

청구항 1에 있어서,

공정 장비의 Asset ID는,

OPC UA Node 규격에 따라 할당되고, 이름, 속성, 공정 정보를 포함하고,

관리-셀 ID는,

OPC UA Node 규격에 따라 할당되고, 제조사 고유 코드 형태로 표현되는 것을 특징으로 하는 관리-셀 기반 공정 장비 관리 방법.

청구항 7

청구항 1에 있어서,

Hardware Configuration 정보는,

물리적 Configuration 기준의 Topology Structure, 통신 Port와 IO Signal에 대한 속성 정보를 포함하는 것을 특징으로 하는 관리-셀 기반 공정장비 관리 방법.

청구항 8

청구항 1에 있어서,

Operating Data 정보는,

공정, line, Station, Sub station, 기능, 역할 정보를 포함하는 것을 특징으로 하는 관리-셀 기반 공정장비 관리 방법.

청구항 9

청구항 1에 있어서,

Management Data 정보는,

상위 애플리케이션과 연동을 위한 데이터를 포함하는 것을 특징으로 하는 관리-셀 기반 공정장비 관리 방법.

청구항 10

공정 장비들의 정보들을 관리-셀들로 표현하는 제1 서버들;

제1 서버들에 의해 표현된 관리-셀들을 이용하여 공정 장비들에 접속하고 통신하는 제2 서버;를 포함하고,

관리-셀은,

관리-셀 헤더와 관리-셀 바디를 포함하고,

관리-셀 헤더는,

공정 장비의 Asset ID 및 관리-셀 ID를 포함하며,

관리-셀 바디는,

공정 장비의 Hardware Configuration 정보, Operating Data 및 Management Data를 포함하는 것을 특징으로 하

는 관리-셀 기반 공장 관리 시스템.

청구항 11

제1 서버들이, 공정 장비들의 정보들을 획득하는 단계; 및
 제1 서버들이, 수집한 공정 장비들의 정보들을 관리-셀들로 표현하는 단계;를 포함하고,
 관리-셀은,
 관리-셀 헤더와 관리-셀 바디를 포함하고,
 관리-셀 헤더는,
 공정 장비의 Asset ID 및 관리-셀 ID를 포함하며,
 관리-셀 바디는,
 공정 장비의 Hardware Configuration 정보, Operating Data 및 Management Data를 포함하는 것을 특징으로 하는 관리-셀 기반 공정장비 관리 방법.

청구항 12

공정 장비들의 정보들을 관리-셀들로 표현하는 제1 서버들;을 포함하고,
 관리-셀은,
 관리-셀 헤더와 관리-셀 바디를 포함하고,
 관리-셀 헤더는,
 공정 장비의 Asset ID 및 관리-셀 ID를 포함하며,
 관리-셀 바디는,
 공정 장비의 Hardware Configuration 정보, Operating Data 및 Management Data를 포함하는 것을 특징으로 하는 관리-셀 기반 공장 관리 시스템.

발명의 설명

기술 분야

[0001] 본 발명은 스마트 팩토리 관련 기술에 관한 것으로, 더욱 상세하게는 상호운용 등을 위해 스마트 팩토리의 모듈 조합형 자동화 공정장비(CNC, Conveyor, PLC, RFID 등)에 대한 정보를 관리하는 방법에 관한 것이다.

배경 기술

[0003] 기존 공장의 제조라인은 초기 Engineering 단계 및 실제 Operating 단계에서 Device와 Device 간, Device와 공장의 각종 Software 솔루션(MES, SCADA, Engineering Tool등)과의 통신 구성 및 Node 정보의 Matching 과정에서 많은 노동력과 시간이 소요되며 이는 공장 운영측면에서 큰 비용 손실을 초래하는 요소이다.

[0004] 궁극적인 원인으로는 첫째, 표준 통신프로토콜의 부재에 있다. 현재 공장 내부에는 산재된 통신 매개체(Ethernet, RS485, RS232 등)와 프로토콜(Profinet, CCLink, EtherCat, ModbusTCP/RTU 등)의 사용되고 있으며 서로 상이한 통신 프로토콜을 사용하는 다양한 제조사의 Hardware/Software가 혼용되어 사용되어야 하는 공장 환경에서는 프로토콜 변환 및 Matching을 위한 별도의 Engineering 작업이 필요한 실정이다.

[0005] 둘째, 통신프로토콜의 비표준화 뿐 아니라 정보 모델의 비표준화로 인하여 공장 내 각 Asset의 Capability,

Node information 등 정보의 의미론적 관점에서 Matching 작업이 필요한 상황이다.

[0006] 이러한 원인은 제품의 전 생산 주기(Manufacturing Life Cycle) 과정에서 초기 설계 및 Engineering에서부터 제품 생산 및 서비스에 이르는 과정에 이르기까지의 자산(Asset) 간 정보의 교환을 위해, 또한 공장 내 Device 간, Device(PLC, Sensor 등)와 IT Solution(MES, SCADA 등)간 정보의 교환을 위해 수동 Engineering 작업(노동력, 작업시간) 발생을 초래하고 있다.

발명의 내용

해결하려는 과제

[0008] 본 발명은 상기와 같은 문제점을 해결하기 위하여 안출된 것으로서, 본 발명의 목적은, 공장 내 모듈형 공정 장비, 나아가 공정 디바이스 및 설비(통칭 OT)의 상호운용 및 다양한 IT 솔루션과의 Plug&Works를 위한 정보 모델인 관리-셸(Administration Shell)을 이용한 공정장비 관리 방법을 제공함에 있다.

과제의 해결 수단

[0010] 상기 목적을 달성하기 위한 본 발명의 일 실시예에 따른, 관리-셸 기반 공정장비 관리 방법은, 제1 서버들이, 공정 장비들의 정보들을 관리-셸들로 표현하는 단계; 제2 서버가, 제1 서버들에 의해 표현된 관리-셸들을 이용하여 공정 장비들에 접속하고 통신하는 단계;를 포함하고, 관리-셸은, 관리-셸 헤더와 관리-셸 바디를 포함하고, 관리-셸 헤더는, 공정 장비의 Asset ID 및 관리-셸 ID를 포함하며, 관리-셸 바디는, 공정 장비의 Hardware Configuration 정보, Operating Data 및 Management Data를 포함한다.

[0011] 그리고, 관리-셸은, 공정 장비들 간 상호 운용, 공정 장비와 상위 애플리케이션 간 상호 운용, 상위 애플리케이션들 간 상호 운용을 위한 정보 체계를 갖을 수 있다.

[0012] 제2 서버는, 제1 서버들에 의해 표현된 관리-셸들을 이용하여 생성될 수 있다.

[0013] 제2 서버는, 관리-셸들로부터 Node 정보를 추출하고, 추출한 Node 정보를 참조로 공정 장비의 데이터를 획득하고, 획득한 데이터를 다른 공정 장비의 데이터와 교환할 수 있다.

[0014] 제2 서버는, 획득한 데이터를 상위 애플리케이션에 전달할 수 있다.

[0015] 공정 장비의 Asset ID는, OPC UA Node 규격에 따라 할당되고, 이름, 속성, 공정 정보를 포함하고, 관리-셸 ID는, OPC UA Node 규격에 따라 할당되고, 제조사 고유 코드 형태로 표현될 수 있다.

[0016] Hardware Configuration 정보는, 물리적 Configuration 기준의 Topology Structure, 통신 Port와 IO Signal에 대한 속성 정보를 포함할 수 있다.

[0017] Operating Data 정보는, 공정, line, Station, Sub station, 기능, 역할 정보를 포함할 수 있다.

[0018] Management Data 정보는, 상위 애플리케이션과 연동을 위한 데이터를 포함할 수 있다.

[0019] 한편, 본 발명의 다른 실시예에 따른, 관리-셸 기반 공장 관리 시스템은, 공정 장비들의 정보들을 관리-셸들로 표현하는 제1 서버들; 제1 서버들에 의해 표현된 관리-셸들을 이용하여 공정 장비들에 접속하고 통신하는 제2 서버;를 포함하고, 관리-셸은, 관리-셸 헤더와 관리-셸 바디를 포함하고, 관리-셸 헤더는, 공정 장비의 Asset ID 및 관리-셸 ID를 포함하며, 관리-셸 바디는, 공정 장비의 Hardware Configuration 정보, Operating Data 및 Management Data를 포함한다.

[0020] 한편, 본 발명의 다른 실시예에 따른, 관리-셸 기반 공정장비 관리 방법은, 제1 서버들이, 공정 장비들의 정보들을 획득하는 단계; 및 제1 서버들이, 수집한 공정 장비들의 정보들을 관리-셸들로 표현하는 단계;를 포함하고, 관리-셸은, 관리-셸 헤더와 관리-셸 바디를 포함하고, 관리-셸 헤더는, 공정 장비의 Asset ID 및 관리-셸 ID를 포함하며, 관리-셸 바디는, 공정 장비의 Hardware Configuration 정보, Operating Data 및 Management Data를 포함한다.

[0021] 한편, 본 발명의 다른 실시예에 따른, 관리-셸 기반 공장 관리 시스템은, 공정 장비들의 정보들을 관리-셸들로 표현하는 제1 서버들;을 포함하고, 관리-셸은, 관리-셸 헤더와 관리-셸 바디를 포함하고, 관리-셸 헤더는, 공정

장비의 Asset ID 및 관리-셸 ID를 포함하며, 관리-셸 바디는, 공정 장비의 Hardware Configuration 정보, Operating Data 및 Management Data를 포함한다.

발명의 효과

[0023] 이상 설명한 바와 같이, 본 발명의 실시예들에 따르면, 정보 모델 표준화를 통한 초기 엔지니어링 및 시운전을 위한 노동력 및 시간 감축이 가능하고, 제조라인의 유연성을 확대하여 제조사 독립적인 상호운용을 가능하게 하며, 기존 산업 솔루션 영역을 넘어 다양한 서비스 플랫폼과의 확장 연계가 가능해진다.

도면의 간단한 설명

[0025] 도 1은 IEC62541(OPC UA) 아키텍처,
 도 2는 IEC62714 아키텍처(IEC62714 Whitepaper),
 도 3은 CAEX 개념(IEC62714 Whitepaper),
 도 4는 관리-셸(Administration Shell)의 개념,
 도 5는 관리-셸을 통한 상호운용의 개념,
 도 6은 관리-셸의 일반적인 구조의 개념도,
 도 7은 AML을 통한 관리-셸 구현 예,
 도 8은 관리-셸 헤더 선언(좌) 및 상세 속성 값 지정(우),
 도 9는 관리-셸 바디 중 Configuration 정보 및 상세 속성 값 지정,
 도 10은 관리-셸 바디 중 Operating Data 및 상세 속성 값 지정,
 도 11은 관리-셸 바디 중 Management 정보 및 상세 속성 값 지정,
 도 12는 IEC62714 RoleClassLib,
 도 13은 본 발명의 실시예가 적용가능한 자동화 공장의 구성도,
 도 14에는 실제 구현된 Aggregation 서버의 구동 화면, 그리고,
 도 15는 IoT 플랫폼과의 Data 연동 예를 나타낸 도면이다.

발명을 실시하기 위한 구체적인 내용

[0026] 이하에서는 도면을 참조하여 본 발명을 보다 상세하게 설명한다.

[0027] 본 발명의 실시예에서는, 공장 자산 간 표준화된 정보 모델과 표준 통신기술을 통하여 단순화된 엔지니어링 작업만으로 데이터를 교환하고, 특히 기기종 간의 연결성 및 상호운용이 가능하게 하는 기법을 제시한다.

[0029] 가. 적용된 국제 표준규격

[0030] 1. IEC62541 (OPC UA)

[0031] 1) IEC62541(OPC UA) 아키텍처

[0032] OPC UA는 구성 요소, 응용 프로그램 및 기능 간의 수직 및 수평 연결을 지원하기 위한 규격으로, 산업 센서 및 액추에이터, 제어 시스템, MES(Manufacturing Execution Systems) 및 ERP(Enterprise Resource Planning) 시스템 등과 같은 모든 산업 영역의 구성 요소에 적용할 수 있음. OPC UA는 정보 교환 촉진을 위해 다음을 명시하고 있다.

[0033] - 구조, 행동 및 의미를 나타내는 정보 모델

- [0034] - 응용 프로그램간에 상호 작용할 수있는 인터페이스 모델
- [0035] - 엔드 포인트간에 데이터를 전송하는 통신 모델
- [0036] - 시스템 간의 상호 운용성을 보장하기위한 적합성 모델
- [0037] 도 1은 IEC62541(OPC UA) 아키텍처를 나타낸 도면이다. 도 1에 도시된 바와 같이, OPC UA 시리즈는 다중 파트 사양으로 구성된다. 파트 1~7과 파트14는 OPC-UA의 핵심 기능을 지정하며 이러한 핵심 기능은 OPC AddressSpace의 구조와 이에 작동하는 서비스를 정의한다.
- [0038] IEC 62541-14는 IEC 62541-4의 서비스에 의해 정의 된 클라이언트 서버 패턴 외에도 OPC-UA 게시 구독 패턴을 정의한다. 파트 8~11은 데이터 액세스(DA), 알람 및 이벤트(A&E) 및 히스토리 데이터 액세스(HDA)와 같은 별도의 OPC COM 사양에 의해 핵심 기능을 적용한다. IEC 62541-4는 OPC UA의 Discovery 메커니즘을 설명하고 IEC 62541-13은 데이터 집계 방법을 설명한다.
- [0040] 2) OPC UA, AutomationML 정보 모델의 가능성
- [0041] OPC 통합 아키텍처 및 자동화 마크 업 언어 결합을 통해, AML과 OPC UA 요소 간의 매핑을 구현할 수 있다. 또한 AML은 특히 런타임 OPC UA 서버 또는 다른 런타임 통신 시스템 내에서 런타임 데이터 소스의 모델링을 가능하게 하며, 이를 통해 관리 셸의 AML로 설명 된 데이터 모델과 OPC UA가 나타내는 관련 정보 모델 간의 공식적인 전환을 허용한다.
- [0042] 이를 통해 OPC UA를 통해 AML로 기술 된 데이터 모델을 의사 소통 및 운영할 수 있다. OPC UA는 여러 관리 셸 간의 스마트 제조 상호운용 통신 요구 사항을 충족하며 따라서 AML은 OPC UA가 해당 도메인의 연결을 위한 것과 동일한 방식으로 스마트 제조 도메인에서 상호 운용성을 위한 중요한 원동력이 될 수 있다.
- [0044] 2. IEC62714(Automation ML)
- [0045] 1) IEC62714 AutomationML 개요
- [0046] AutomationML(AML)은 IEC 62714에 표준화된 확장성, 개방성, 중립성 및 무료 데이터 교환 형식으로 이중 엔지니어링 도구 환경에서 데이터 교환을 지원하기 위한 XML 기반 표준기술이다. 일관된 방식으로 데이터 손실 없이 이기종 소프트웨어 툴 간에 엔지니어링 데이터를 전송할 수 있도록 지원한다. 도 2는 IEC62714 아키텍처 (IEC62714 Whitepaper)를 나타낸 도면이다.
- [0047] AML이 제공하는 메타 모델을 사용하면 생산 시스템 구조를 AutomationML 객체의 계층으로 모델링 할 수 있다. 외부 문서에 저장된 정보에 대한 참조뿐만 아니라 형상 및 기구학, 논리 데이터(동작 및 시퀀싱), 구동 데이터 및 AutomationML 객체 간의 관계를 포함할 수 있다. 이 객체 지향 패러다임을 따르면, 각 AutomationML 객체는 서로 다른 분야와 관련된 다른 의미를 가진 다른 정보를 통합할 수 있다.
- [0048] CAEX는 AML의 최상위 형식이다. 도 3은 CAEX의 개념(IEC62714 Whitepaper)을 나타낸 도면이다. CAEX는 AML 객체가 InternalElements(IE)로 표시되는 다음 네 가지 개념으로 특징지어 진다.
- [0049] - RoleClasses(RC)를 사용함으로써 AML 객체에 대한 의미론을 모델링 할 수 있으며 RoleClass는 기술적인 구현과 독립적인 물리적 또는 논리적 객체의 기능을 설명한다. 추상적인 방법으로, 그리고 제조자 독립적으로 객체를 지정할 수 있는 기능성을 제공하며 각 객체의 의미론은 RoleClass 라이브러리에 저장된다.
- [0050] - InterfaceClasses(IC)를 사용하여 AML 객체의 인터페이스에 대한 의미를 모델링 할 수 있다. "인터페이스-클래스"는 기술적 구현과 독립적으로 물리적 객체 또는 논리적 객체의 관계 또는 문서에 대한 참조를 설명하며 인터페이스 클래스 라이브러리에 저장된다.
- [0051] - SystemUnitClasses(SUC)를 사용하여 AML 객체의 유형 또는 템플릿을 모델링 할 수 있다. 객체의 유형, 템플릿은 시스템 단위 클래스 라이브러리에 저장된다.
- [0052] - InstanceHierarchy(IH)는 실제 프로젝트를 저장한다.
- [0053] 일관된 데이터 교환을 위하여 객체 식별 개념이 필수적임에 따라 AML이 제공하는 개념은 모든 객체가 UUID(범용 고유 식별자)를 가지며 객체가 존재하는 한 보관해야한다는 것을 규칙으로 한다. 이 ID 지속성을 통해 전체 라

이프 사이클 동안 특정 AML 오브젝트를 식별할 수 있다.

[0054] 외부 문서를 참조하는 AML 메커니즘을 사용하면 AML 객체에 추가 정보를 통합할 수 있다. 따라서 문서 유형과 첨부된 정보의 의미를 고유하게 식별하는 데 사용되는 인터페이스 클래스 및 역할 클래스의 특수 세트를 제공하여 참조된 문서가 해당 내용에 대한 고유한 식별 개념을 갖는 경우 참조된 문서를 하위에 다시 참조할 수도 있다. AML 메타 모델 외에도 사용할 수 있는 데이터 교환 형식에 대한 사양이 있으며, AML은 XML 기반 형식이므로 XML 암호화, XML 서명 또는 XML 직렬화와 같은 XML 관련 기능을 사용할 수 있다.

[0056] 나. 관리-셸(Administration Shell) 개념

[0057] 1. 관리-셸의 정의

[0058] 관리-셸은 자산(Asset)의 표준화된 디지털 표현이며, 제조 시스템을 관리하는 응용 프로그램 및 자산 간의 상호운용을 위한 정보 모델이다. 도 4는 관리-셸(Administration Shell)의 개념을 도시한 도면이다. 도시된 바와 같이, 스마트 모듈화 장비는 물리적 자산과 논리적 디지털 정보로 구성된다. 즉, 관리-셸이 결합된 형태를 의미한다.

[0059] 도 5는 관리-셸을 통한 상호운용의 개념을 도시한 도면이다. 도시된 바와 같이, 관리-셸은 공정 내 각각의 구성요소, 기계 또는 설비의 논리적 표현이자 표준화된 정보 모델이며 제조 수명 주기(Manufacturing life cycle) 전반에 걸친 상호운용, 공장 내 수직/수평적 통합을 위한 상호운용을 제공한다.

[0061] 2. 관리-셸의 구조

[0062] 다음의 요구사항을 적용하여 관리-셸의 일반적인 구조를 개발할 수 있다. 이는 자동화 및 ICT 기술 모두에 적용할 수 있는 표준 개념을 기반으로 해야 하며, 관련 스마트 제조 측면(수평적 통합, 수직적 통합, 엔지니어링 통합)에 관한 요구사항을 만족해야 한다.

[0063] - 관리-셸은 광범위한 기술 도메인의 속성을 포함할 수 있어야 하고 어느 도메인에서 파생된 것인지 식별할 수 있어야 한다.

[0064] - 각 관련 기술 도메인 내에서 정의를 찾는 경우, 표준, 컨소시엄 규격 및 제조업체 사양 세트의 요구사항을 각각 충족하는 다양한 절차 모델을 허용해야 한다.

[0065] 도 6은 관리-셸의 일반적인 구조의 개념도이다. 도시된 바와 같이, 관리-셸은 "헤더"와 "본문"으로 구성되며, "헤더"는 자산 및 관리-셸의 식별자 또는 고유 아이디를 제공하고, "본문"은 관리-셸 내부의 개별 서브 정보 모델을 포함한다.

[0066] 각 하위 모델은 계층적 형태로 구성되며 각각의 속성(Property)과 기능 (Capability)를 가진다.

[0067] 헤더(Header)는 자산의 산업분야 적용에 적합한 정보보안에 의해 안전하게 보호되어야 한다.

[0068] - 식별자(ID)에 대한 최소한의 정보를 포함하여 고유하게 식별

[0069] - 식별자는 정보 및 기능을 검색하기 위한 API의 루트 진입점 역할을 수행

[0070] - 자산의 유형(Hardware, Software, Component 등)을 표기

[0071] 본문(Body)은 자산의 구동(Operating) 정보, 기능(Capability) 정보가 포함되며, 실제적인 정보의 운반자 역할을 수행한다.

[0072] 관리-셸은 공장/플랜트, 생산라인, 스테이션, 제어 장치 및 현장 장치와 같은 공장의 모든 계층 구성요소의 가상화 표현이 될 수 있다.

[0073] 상호운용을 위해 타 자산에 의해 정보 및 기능에 대해 일반적인 데이터 형식 및 액세스 방식이 가능해야 한다.

[0074] 관리-셸을 활용한 서비스지향 API는 다음을 포함하여 외부에 서비스를 제공할 수 있다.

[0075] - 관리-셸 내의 정보 및 기능(Capability)

[0076] - 관리-셀과 자산의 Addressing 및 식별/검색 메커니즘

[0078] 다. 실제 공장설비 적용을 위한 관리-셀 방법론

[0079] 1. 본 발명의 실시예에 따른 공정장비 관리/상호운용 방법을 위한 관리-셀의 기능 정의

[0080] 1) 수직/수평 연계를 위한 상호운용 : 공장 내 구성요소의 IT/OT 간 상호운용을 구현하기 위한 관리-셀

[0081] - 동일 시점(현 Operating 시점)에서의 Asset 간 상호운용

[0082] - 표준 통신 프로토콜(IEC 62541) 및 정보 모델을 기반으로 한 제조사 독립적인 상호운용 구현

[0083] - 수평적 연계의 예: 머신 to 머신의 상호운용, 전력 스카다 to 구동 스카다의 상호운용

[0084] - 수직적 연계의 예: 머신 to MES의 상호운용

[0085] 2) 제조 수명 주기(Manufacturing Life Cycle) 전반에 걸친 상호운용 : 제품 설계 및 생산 주문에서 최종 서비스까지의 상호운용

[0086] - 관리-셀을 통한 설계 단계에서 장비 구동 및 시운전까지의 상호운용

[0087] - 공정 구동 데이터의 관리 플랫폼 및 서비스 플랫폼까지의 상호운용

[0089] 2. 본 발명의 실시예에 따른 공정장비 관리/상호운용 방법에서 이용할 관리-셀의 구조

[0090] ① 관리-셀의 전체 구조(도 7 참조)

[0091] 1) 관리-셀 헤더(Header)와 관리-셀 바디(Body)로 구성

[0092] 2) 관리-셀 헤더는 고유 식별 ID를 구현

[0093] 3) 관리-셀 바디는 아래와 같이 세 가지 Sub Item으로 구성

[0094] - 관리-셀 바디1(Body1): Hardware Configuration 정보

[0095] - 관리-셀 바디2(Body2): Operation Data

[0096] - 관리-셀 바디3(Body3): Management Data

[0097] 4) 각 Element는 IEC62714에서 제공하는 RoleClass를 참조하여 Capability 및 상세 속성을 부여

[0099] ② 관리-셀 헤더(도 8 참조)

[0100] 1) 자산(Asset) 식별자 ID

[0101] - OPC UA Node 규격을 따름

[0102] - 자산의 고유 식별 번호 체계를 제조사 고유 코드 형태로 구현

[0103] - 식별자 ID의 포함 정보: Asset 이름, Asset속성, 공정 번호 등을 기업 고유체계로 구현

[0104] - ID의 Data Type은 unsigned_int32로써 0 ~ 4,294,967,295 범위로 표현

[0105] 2) 관리-셀(Administration Shell) 식별자 ID

[0106] - OPC UA Node 규격을 따름

[0107] - 관리-셀 식별 번호 체계를 제조사 고유 코드 형태로 구현

[0108] - ID의 Data Type은 unsigned_int32 로써, 0~4,294,967,295 범위로 표현

[0110] ③ 관리-셀 바디

- [0111] 1) 관리-셀 바디1(Body1): Hardware Configuration 정보(도 9 참조)
- [0112] - PLC, Remote IO, Sensor, Actuator 등 물리적 Configuration 기준의 Topology Structure
- [0113] - Device 별 통신 Port(RJ45, RS485, Fiber Optic등), IO Signal 등에 대한 상세 속성 정보(DC 24V input/output, Relay, 4-20mA Analog input/output 등) 등
- [0114] - 본 정보를 통해 AML을 지원하는 공장 자동화 Engineering Tool 또는 CAD Tool 상호간의 Hardware configuration 정보 교환을 단순 Import작업만으로 적용 가능함.
- [0115] - 각 Internal Element는 IEC62714에서 규정하는 RoleClass에 따라 해당하는 Role이 적용되어야 함(예: PLC는 Device, 하위 모듈은 DeviceItem, IO Signal은 Channel)
- [0116] 2) 관리-셀 바디2(Body2): Operating Data(도 10 참조)
- [0117] - 논리적 Operating 기준의 Hierarchy/Structure
- [0118] - 공정, line, Station, Sub station의 이름, 기능, 역할 (예: Pangyo Factory, AFB Line, Filling Station, Gripper)
- [0119] - Operating을 위한 OPC UA Node정보 구현 (OPC UA 라이브러리 모델 참조)
- [0120] - 본 정보를 통해 OPC UA를 활용하는 다양한 Application과의 제조사 독립적인 상호운용 및 통신 가능
- [0121] 3) 관리-셀 바디3(Body3): Management Data(도 11 참조)
- [0122] - 제조 IT 솔루션 및 각종 서비스 플랫폼과의 연동을 위한 자산 관리 데이터
- [0123] - 서비스를 위해 추가해야할 정보: 인증 및 Risk Assessment에 활용
- [0124] - 설치 연월일, 제조사, 작업자, 가동누적시간, 정비 이력 등이 포함될 수 있음
- [0125] - 그 밖에 현장 현황에 맞는 사용자 정의 OPC UA 데이터를 자유롭게 확장 생성 가능
- [0126] 4) IEC62714의 RoleClassLib(도 12 참조)
- [0127] - 각 Element별 고유 역할은 RoleClassLib에서 참조되어야함.
- [0128] - 사용자 정의 Role은 IEC62714를 통하여 공식 등록되어야함.

- [0130] 라. 본 발명의 실시예가 적용가능한 자동화 공정 시스템
- [0131] 1. Plug&Works 구현
- [0132] 1) Plug&Works의 구현 절차
- [0133] 도 13은 본 발명의 실시예가 적용가능한 자동화 공장의 구성도이다. 공장 내 공정장비(110)를 포함한 OT, IT 계층의 모든 자산은 표준화된 정보 모델(관리-셀, AML file)을 통하여 정보화 된다. OPC UA 서버들(120)이 제공하는 모델 정보들을 이용하여 Aggregation 서버(130)를 생성한다.
- [0134] Aggregation 서버(130)는 모델 정보들을 AML-to-OPC UA Converter에 import하여 생성할 수 있으며, Aggregation Execution file을 실행하면 관리-셀에 등록된 자산에 자동 접속하여 통신할 수 있다.
- [0135] 공장은 기존 엔지니어링 방식인 수동 통신 설정 및 Node Matching 작업이 필요 없으며, 마치 USB Device를 PC에 장착하였을 때 Driver가 자동 설치되어 사용하는 것과 같이, 관리-셀을 통하여 Aggregation 서버(130)를 생성함으로써 자산 간 상호운용이 가능하다.

- [0137] 2. AML기반 관리-셀을 통한 Aggregation 서버 생성 및 구성 상세
- [0138] 1) 관리-셀로부터 OPC UA Node정보 Parsing
- [0139] AML-to-OPC UA Converting 기술을 활용하여 AML 기반의 관리-셀로부터 OPC UA Node 정보를 추출한다.

- [0140] 2) AML-to-OPC UA Converter를 통하여 Aggregation 서버 생성
- [0141] Parsing된 UA Node 정보를 참조로 생성된 Aggregation 서버(130)는, 공장 현장의 OPC UA 서버(120)를 내장한 Field Device(110)에서 값을 취합하여, 다른 Field Device(110)와 교환하거나, 상위 IT Solution 및 서비스 플랫폼(140)과의 연동을 위해, Middle Layer에 위치한다.
- [0142] 생성된 Aggregation 서버(130)는 exe파일 형태이며, PC 환경을 갖는 어떠한 Device에서도 운용할 수 있다. 도 14에는 실제 구현된 Aggregation 서버(130)의 구동 화면을 예시하였다.
- [0143] 3) 다양한 IT 솔루션과의 연동
- [0144] Aggregation 서버(130)는 IoT 플랫폼, MS Azure, UA Expert 등 다양한 상용솔루션과의 상호운용이 가능하다. 도 15는 IoT 플랫폼과의 Data 연동 예를 나타낸 도면이다.
- [0146] 지금까지, 관리-셸을 이용한 모듈 조합형 자동화 공정장비 관리 방법에 대해 바람직한 실시예를 들어 상세히 설명하였다.
- [0147] 구체적으로, 본 발명의 실시예에서는, 장비 운영 및 관리를 위한 표준화된 정보 모델을 정의하고, 상호운용 통신 프로토콜인 IEC62541(OPC UA) 기반의 Node 정보를 포함하며, 전체적인 정보 모델은 IEC62714(AutomationML)을 기준으로 공장 내 자산(Asset)의 OT/OT 간 수평적 상호 운용 및 OT/IT 간 수직적 상호 운용을 가능하게 하는 관리-셸(Administration Shell)의 상세 정의를 위한 방법론을 제시하였다.
- [0148] 본 발명의 실시예에 따른 관리-셸 방법론은, 공장 내 Asset간의 표준화된 정보 모델을 구축하여 전 생산 주기(Manufacturing Life Cycle)에 걸쳐 수직/수평적 상호운용이 가능한 Plug&Works를 가능하게 한다.
- [0149] 또한, 본 발명의 실시예에 따른 관리-셸 방법론은, 기존의 산재되어 있는 통신 프로토콜이 아닌, 글로벌 표준 통신기술이며 한국의 국가기술표준원, 독일의 Industry4.0 등 제조기술의 상호운용을 위한 표준 통신기술인 IEC62541(OPC UA)를 기반으로 한 모델링 방법론이다.
- [0150] 아울러, 본 발명의 실시예에 따른 관리-셸 방법론은, 기존 공장에 없었던 정보 모델을 제안하며 설비 엔지니어링 단계에서의 Configuration 정보의 교환, 설비 구동을 위한 Operating 정보의 교환, 설비 관리 및 서비스를 위한 Management 정보의 교환을 위한 구조를 갖는다.
- [0151] 이에 의해, 정보 모델 표준화를 통한 초기 엔지니어링 및 시운전을 위한 노동력 및 시간 감축이 가능하고, 제조라인의 유연성 확대를 통한 제조사 독립적인 상호운용이 가능해지며, 기존 산업 솔루션 영역을 넘어 다양한 서비스 플랫폼과의 확장 연계가 가능해진다.
- [0152] 한편, 본 실시예에 따른 장치와 방법의 기능을 수행하게 하는 컴퓨터 프로그램을 수록한 컴퓨터로 읽을 수 있는 기록매체에도 본 발명의 기술적 사상이 적용될 수 있음은 물론이다. 또한, 본 발명의 다양한 실시예에 따른 기술적 사상은 컴퓨터로 읽을 수 있는 기록매체에 기록된 컴퓨터로 읽을 수 있는 코드 형태로 구현될 수도 있다. 컴퓨터로 읽을 수 있는 기록매체는 컴퓨터에 의해 읽을 수 있고 데이터를 저장할 수 있는 어떤 데이터 저장 장치이더라도 가능하다. 예를 들어, 컴퓨터로 읽을 수 있는 기록매체는 ROM, RAM, CD-ROM, 자기 테이프, 플로피 디스크, 광디스크, 하드 디스크 드라이브, 등이 될 수 있음은 물론이다. 또한, 컴퓨터로 읽을 수 있는 기록매체에 저장된 컴퓨터로 읽을 수 있는 코드 또는 프로그램은 컴퓨터간에 연결된 네트워크를 통해 전송될 수도 있다.
- [0153] 또한, 이상에서는 본 발명의 바람직한 실시예에 대하여 도시하고 설명하였지만, 본 발명은 상술한 특정의 실시예에 한정되지 아니하며, 청구범위에서 청구하는 본 발명의 요지를 벗어남이 없이 당해 발명이 속하는 기술분야에서 통상의 지식을 가진자에 의해 다양한 변형실시가 가능한 것은 물론이고, 이러한 변형실시들은 본 발명의 기술적 사상이나 전망으로부터 개별적으로 이해되어져서는 안될 것이다.

부호의 설명

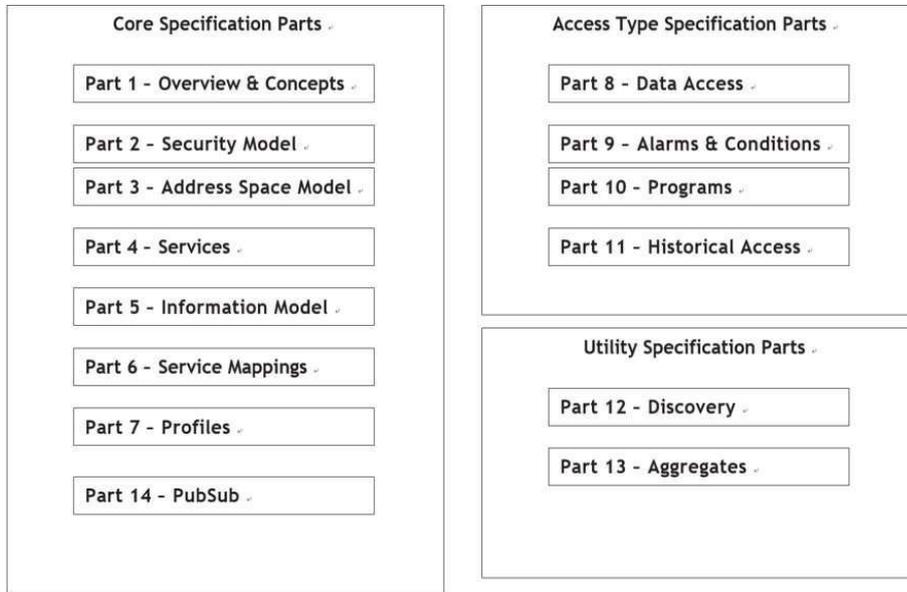
- [0155] 110 : Field Device
- 120 : OPC UA 서버

130 : Aggregation 서버

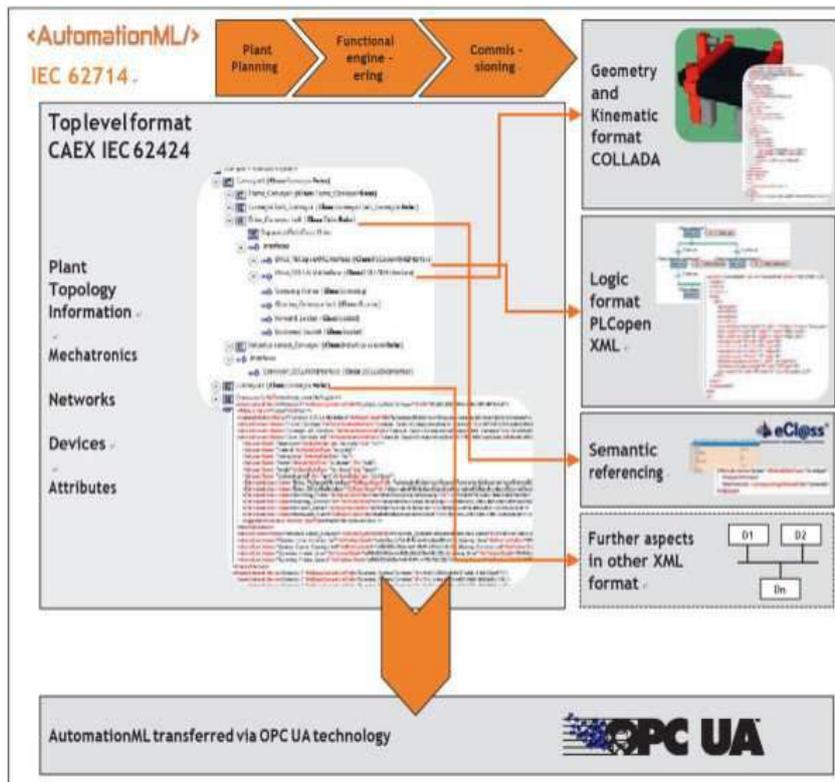
140 : Application

도면

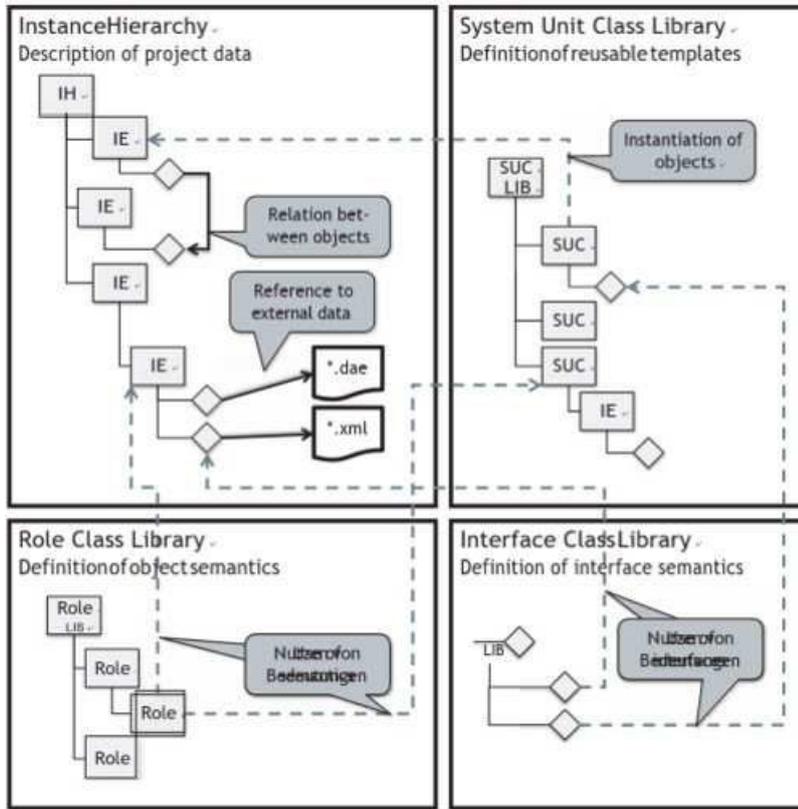
도면1



도면2



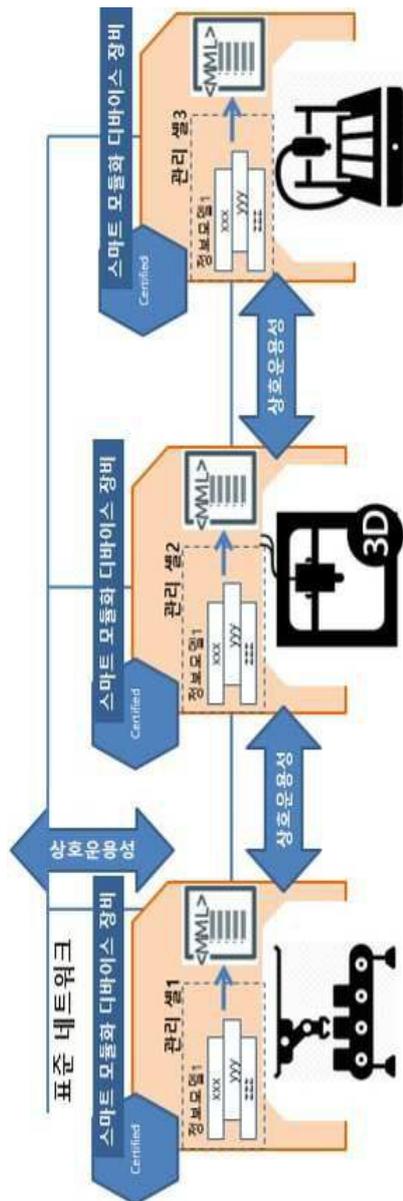
도면3



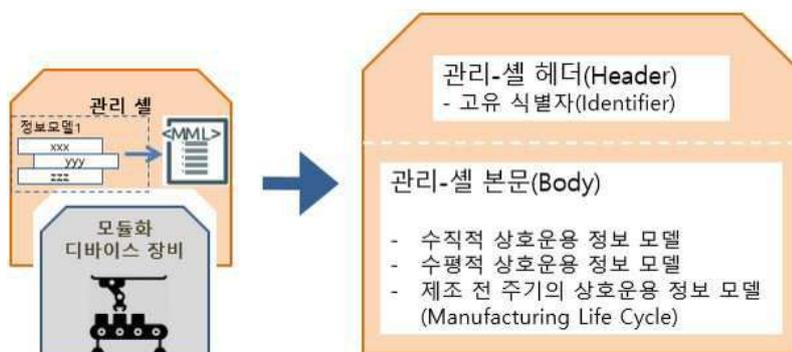
도면4



도면5



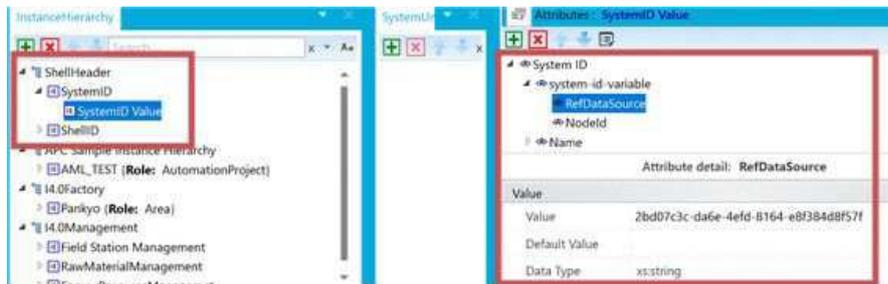
도면6



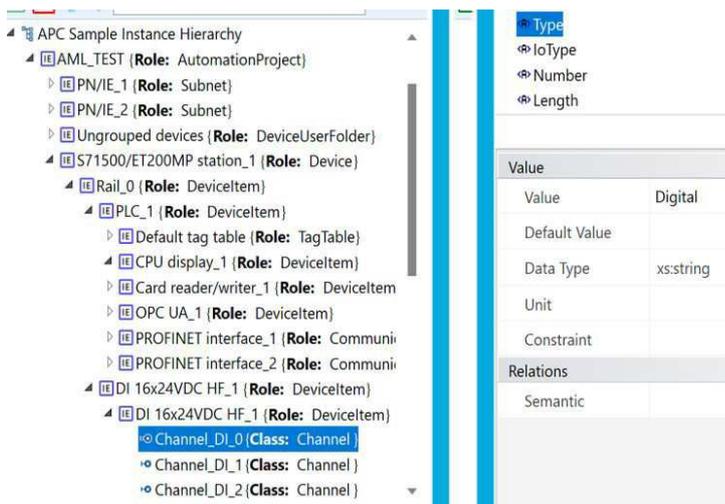
도면7

InstanceHierarchy		
Instance Name	Description	Category
ShellHeader	관리-셸 헤더(Header) System ID, Shell ID	관리-셸 헤더(Header)
APC Sample Instance Hierarchy	관리-셸 바디1(Body1) Hardware Configuration	관리-셸 바디(Body)
I4.0Factory	관리-셸 바디2(Body2) Operating Data	
I4.0Management	관리-셸 바디3(Body3) Management Data	

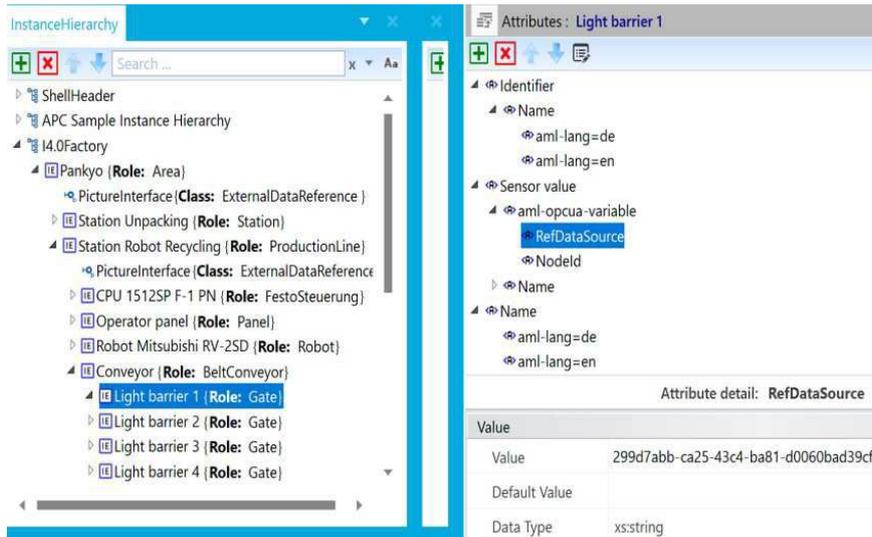
도면8



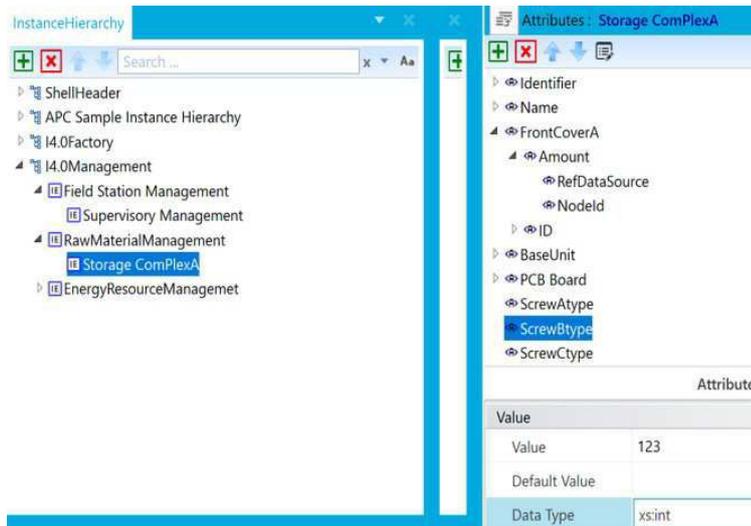
도면9



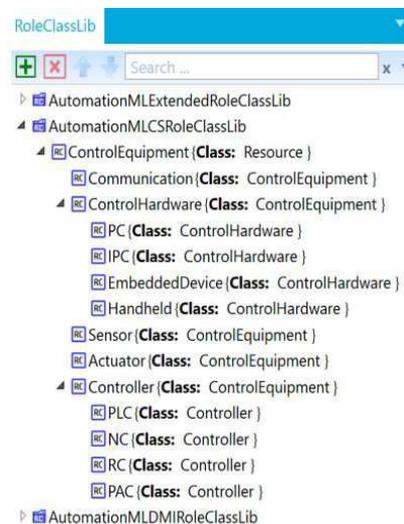
도면10



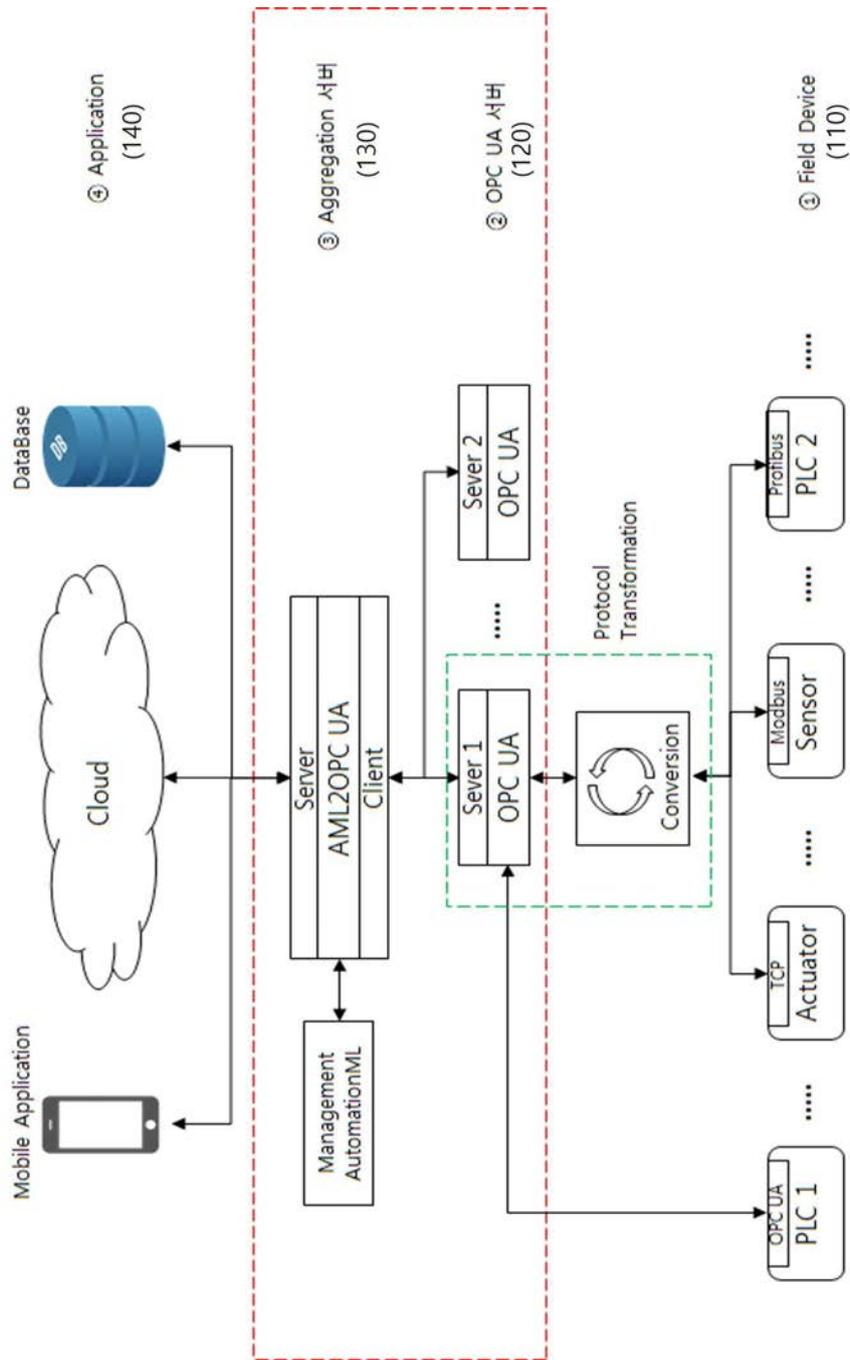
도면11



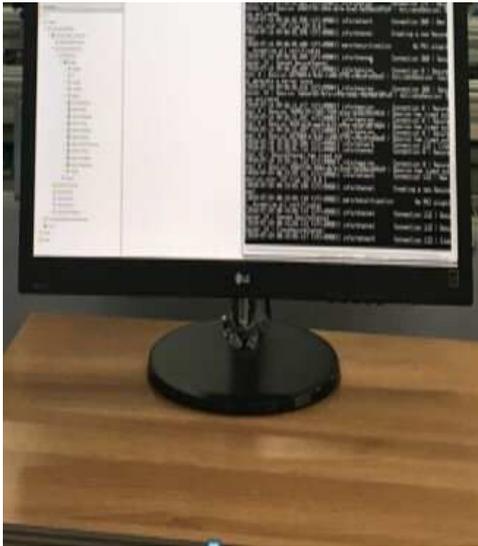
도면12



도면13



도면14



도면15





(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2021-0075898
(43) 공개일자 2021년06월23일

- | | |
|---|--|
| <p>(51) 국제특허분류(Int. Cl.)
H04L 29/06 (2006.01) G05B 19/418 (2006.01)
H04L 29/08 (2006.01)</p> <p>(52) CPC특허분류
H04L 67/42 (2013.01)
G05B 19/4186 (2013.01)</p> <p>(21) 출원번호 10-2020-0174718</p> <p>(22) 출원일자 2020년12월14일
심사청구일자 없음</p> <p>(30) 우선권주장
1020190166330 2019년12월13일 대한민국(KR)</p> | <p>(71) 출원인
한국전자기술연구원
경기도 성남시 분당구 새나리로 25 (야탑동)</p> <p>(72) 발명자
정지은
경기도 성남시 분당구 동판교로 153, 807동 603호
이재현
서울특별시 송파구 삼전로8길 4-17, 302호</p> <p>(74) 대리인
남충우</p> |
|---|--|

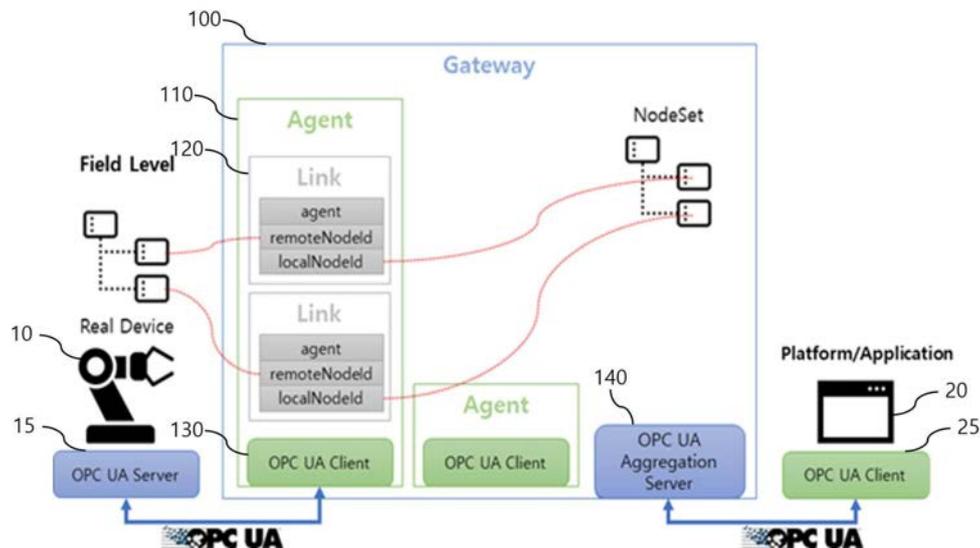
전체 청구항 수 : 총 8 항

(54) 발명의 명칭 효율적인 OPC UA 서버의 데이터 접근 및 통합을 위한 통합 게이트웨이

(57) 요약

효율적인 OPC UA 서버의 데이터 접근 및 통합을 위한 통합 게이트웨이가 제공된다. 본 발명의 실시예에 따른 필드 데이터 통합 방법은, OPC UA Aggregation 서버를 생성하고, 필드 디바이스들에서 생성된 OPC UA 정보 모델들을 OPC UA Aggregation 서버에 OPC UA 노드들로 등록하며, 등록된 OPC UA 노드들을 탐색하여 연결이 필요한 필드 디바이스의 OPC UA 서버에 접속하기 위한 에이전트를 생성한다. 이에 의해, 많은 수의 제조 디바이스 및 장치가 동시에 클라이언트 어플리케이션과 연결되어도 안정적인 서비스를 제공할 수 있게 된다.

대표도



(52) CPC특허분류

H04L 67/16 (2013.01)

이 발명을 지원한 국가연구개발사업

과제고유번호	1415162047
과제번호	20000825
부처명	산업통상자원부
과제관리(전문)기관명	한국산업기술평가관리원
연구사업명	전자부품산업핵심기술개발(R&D)
연구과제명	전자 디바이스 모듈화 장비 가상화를 위한 CPS 연동 및 검증 기술 개발
기여율	1/1
과제수행기관명	전자부품연구원
연구기간	2019.01.01 ~ 2019.12.31

명세서

청구범위

청구항 1

OPC UA Aggregation 서버를 생성하는 단계;

필드 디바이스들에서 생성된 OPC UA 정보 모델들을 OPC UA Aggregation 서버에 OPC UA 노드들로 등록하는 단계;
등록된 OPC UA 노드들을 탐색하여, 연결이 필요한 필드 디바이스의 OPC UA 서버에 접속하기 위한 에이전트를 생성하는 단계;를 포함하는 것을 특징으로 하는 필드 데이터 통합 방법.

청구항 2

청구항 1에 있어서,

OPC UA 노드들은,

OPC UA Aggregation 서버에서 통합되어 계층화된 노드 셋으로 구성되는 것을 특징으로 하는 필드 데이터 통합 방법.

청구항 3

청구항 1에 있어서,

등록된 OPC UA 노드들을 탐색하여, 모니터링할 OPC UA 노드를 선정하여 해당 에이전트에 연결하는 단계;

에이전트를 통해 필드 디바이스의 OPC UA 서버로부터 데이터 변경 알림을 수신하면, 해당 필드 디바이스에 연결된 OPC UA 노드에서 해당 데이터를 갱신하는 단계;를 포함하는 것을 특징으로 하는 필드 데이터 통합 방법.

청구항 4

청구항 2에 있어서,

에이전트는,

필드 디바이스들의 OPC UA 서버들에 연결하기 위한 정보들 및 필드 디바이스들의 OPC UA 노드들에 연결하기 위한 정보들이 각각 매핑되어 있는 링크 정보들; 및

링크 정보들을 참조하여 필드 디바이스의 OPC UA 서버에 연결하기 위한 OPC UA 클라이언트;를 포함하는 것을 특징으로 하는 필드 데이터 통합 방법.

청구항 5

청구항 3에 있어서,

에이전트는,

필드 디바이스의 OPC UA 서버로부터 데이터 변경 사항을 구독하는 것을 특징으로 하는 필드 데이터 통합 방법.

청구항 6

청구항 5에 있어서,

에이전트는,

필드 디바이스의 OPC UA 서버로부터 데이터 변경 알림을 수신하면, 해당 OPC UA 노드에서 해당 데이터를 갱신하도록 OPC UA Aggregation 서버에 콜백하는 것을 특징으로 하는 필드 데이터 통합 방법.

청구항 7

청구항 1에 있어서,

플랫폼 또는 어플리케이션의 OPC UA 클라이언트 요청에 따라, 해당 OPC UA 노드들의 데이터들을 전송하는 단계;를 더 포함하는 것을 특징으로 하는 필드 데이터 통합 방법.

청구항 8

필드 디바이스들에서 생성된 OPC UA 정보 모델들을 등록하는 OPC UA Aggregation 서버 모듈;

등록된 OPC UA 노드들을 탐색하여, 연결이 필요한 필드 디바이스의 OPC UA 서버에 접속하는 에이전트 모듈;를 포함하는 것을 특징으로 하는 게이트웨이.

발명의 설명

기술 분야

[0001] 본 발명은 스마트 공장/제조 기술에 관한 것으로, 더욱 상세하게는 제조 현장 내 많은 수의 OPC UA 장치 및 시스템들을 안정적으로 연동하기 위한 방법에 관한 것이다.

배경 기술

[0003] 오늘날 대부분의 제조 산업 분야에서 스마트 제조 혁신 트렌드에 따라 많은 수의 필드 디바이스, 센서, 장비 등이 시스템의 구성 요소로써 상호 연동되고 있으며, 이와 같은 다양한 필드 디바이스 간의 상호운용성 제공을 위한 기술로써 OPC UA(Open Platform Communication Unified Architecture)가 적용되고 있다.

[0004] 이에 현재의 제조 현장에서는 서버-클라이언트 구조 모델 기반의 OPC UA가 적용된 PLC를 통해서 시스템간의 상호 연동을 지원하고 있기 때문에, 중재(Relay) 및 통합(Aggregation) 프로세스 없이 많은 수의 디바이스 및 장치와 다양한 클라이언트 어플리케이션 간에 직접적인 연결이 이뤄질 경우, 연결 및 구독 서비스 안정성, 실시간성, 정확성에 문제가 발생할 수 밖에 없다.

[0005] 이에 따라 제조 디바이스/장비들과 응용 어플리케이션 간의 연결, 갱신, 구독 등의 전반적인 서비스 관리를 지원할 수 있는 방안이 필요하다.

발명의 내용

해결하려는 과제

[0007] 본 발명은 상기와 같은 문제점을 해결하기 위하여 안출된 것으로서, 본 발명의 목적은, 제조 현장 내 많은 수의 OPC UA 장치 및 시스템들을 안정적으로 연동할 수 있도록 하기 위한 방안으로, 산업 현장에서 클라이언트 어플리케이션의 요청에 따라 복수 개의 장비 내 OPC UA 서버들에게 연결하여 데이터를 효율적으로 취합, 전달 및 관리하는 통합 게이트웨이를 제공함에 있다.

과제의 해결 수단

- [0009] 상기 목적을 달성하기 위한 본 발명의 일 실시예에 따른, 필드 데이터 통합 방법은, OPC UA Aggregation 서버를 생성하는 단계; 필드 디바이스들에서 생성된 OPC UA 정보 모델들을 OPC UA Aggregation 서버에 OPC UA 노드들로 등록하는 단계; 등록된 OPC UA 노드들을 탐색하여, 연결이 필요한 필드 디바이스의 OPC UA 서버에 접속하기 위한 에이전트를 생성하는 단계;를 포함한다.
- [0010] OPC UA 노드들은, OPC UA Aggregation 서버에서 통합되어 계층화된 노드 셋으로 구성될 수 있다.
- [0011] 본 발명의 실시예에 따른 필드 데이터 통합 방법은, 등록된 OPC UA 노드들을 탐색하여, 모니터링할 OPC UA 노드를 선정하여 해당 에이전트에 연결하는 단계; 에이전트를 통해 필드 디바이스의 OPC UA 서버로부터 데이터 변경 알림을 수신하면, 해당 필드 디바이스에 연결된 OPC UA 노드에서 해당 데이터를 갱신하는 단계;를 포함할 수 있다.
- [0012] 에이전트는, 필드 디바이스들의 OPC UA 서버들에 연결하기 위한 정보들 및 필드 디바이스들의 OPC UA 노드들에 연결하기 위한 정보들이 각각 매핑되어 있는 링크 정보들; 및 링크 정보들을 참조하여 필드 디바이스의 OPC UA 서버에 연결하기 위한 OPC UA 클라이언트;를 포함할 수 있다.
- [0013] 에이전트는, 필드 디바이스의 OPC UA 서버로부터 데이터 변경 사항을 구독할 수 있다.
- [0014] 에이전트는, 필드 디바이스의 OPC UA 서버로부터 데이터 변경 알림을 수신하면, 해당 OPC UA 노드에서 해당 데이터를 갱신하도록 OPC UA Aggregation 서버에 콜백할 수 있다.
- [0015] 본 발명의 실시예에 따른 필드 데이터 통합 방법은, 플랫폼 또는 어플리케이션의 OPC UA 클라이언트 요청에 따라, 해당 OPC UA 노드들의 데이터들을 전송하는 단계;를 더 포함할 수 있다.
- [0016] 한편, 본 발명의 다른 실시예에 따른, 게이트웨이는, 필드 디바이스들에서 생성된 OPC UA 정보 모델들을 등록하는 OPC UA Aggregation 서버 모듈; 등록된 OPC UA 노드들을 탐색하여, 연결이 필요한 필드 디바이스의 OPC UA 서버에 접속하는 에이전트 모듈;를 포함한다.

발명의 효과

- [0018] 이상 설명한 바와 같이, 본 발명의 실시예들에 따르면, 통합 게이트웨어 내에 내장된 OPC UA 클라이언트 인스턴스를 통해 클라이언트의 생성, 연결, 연결 끊기 등의 라이프사이클을 관리함으로써, 많은 수의 제조 디바이스 및 장치가 동시에 클라이언트 어플리케이션과 연결되어도 안정적인 서비스를 제공할 수 있게 된다.
- [0019] 또한, 본 발명의 실시예들에 따르면, 통합 게이트웨어에 연결된 디바이스 및 장비의 운영 데이터를 실시간 모니터링하고 업데이트 함으로써, 데이터 송수신의 신뢰도를 향상시킬 수 있게 된다.

도면의 간단한 설명

- [0021] 도 1은 본 발명의 일 실시예에 따른 통합 게이트웨어의 설명에 제공되는 도면,
- 도 2는 OPC UA Aggregation 서버의 주요 기능 구현을 위한 프로그램 코드,
- 도 3은 에이전트의 주요 기능 구현을 위한 프로그램 코드,
- 도 4는 에이전트의 링크의 주요 기능 구현을 위한 프로그램 코드,
- 도 5는 통합 게이트웨어의 동작 프로세스를 나타낸 흐름도, 그리고,
- 도 6은 통합 게이트웨어의 실제 동작 프로세스를 예시한 도면이다.

발명을 실시하기 위한 구체적인 내용

- [0022] 이하에서는 도면을 참조하여 본 발명을 보다 상세하게 설명한다.
- [0023] 본 발명의 실시예에서는 제조 환경 내에 존재하는 OPC UA 장치와 시스템들을 안정적으로 연동하기 위한 방안으

로, 효율적인 OPC UA 서버의 데이터 접근 및 통합을 위한 통합 게이트웨이를 제시한다.

- [0024] 본 발명의 실시예에 따른 통합 게이트웨이는 외부 플랫폼이나 어플리케이션의 OPC UA 클라이언트 요청에 따라, 해당되는 필드 디바이스들의 OPC UA 서버에 접속하여 데이터를 전달 및 관리한다.
- [0025] 이를 수행하기 위한 통합 게이트웨이의 기능은 다음과 같다.
- [0026] 1) 연결된 필드 디바이스들의 OPC UA 서버들로부터 OPC UA 정보 모델들을 확보하여, 정보 구조에 따라 통합된 OPC UA 노드 셋을 구성
- [0027] 2) 구성된 OPC UA 노드 셋에 저장/갱신되는 정보를 외부 플랫폼이나 어플리케이션의 OPC UA 클라이언트의 요청에 따라 서비스로 제공
- [0028] 3) 연결하고자 하는 필드 디바이스들의 OPC UA 서버들에 접속하는 에이전트들을 생성하고, 연결, 연결 끊기 등의 라이프사이클을 관리
- [0029] 4) 연결된 필드 디바이스들의 운영 데이터를 모니터링하고, 데이터가 변경되면 OPC UA Aggregation 서버의 해당 OPC UA 노드의 데이터를 갱신
- [0030] 도 1은 본 발명의 일 실시예에 따른 통합 게이트웨이의 설명에 제공되는 도면이다.
- [0031] 이해와 설명의 편의를 위해, 도 1에서는 통합 게이트웨이(100) 외에 그와 연동하는 필드 디바이스(10)와 외부 플랫폼/어플리케이션(20)을 더 도시하였다. 도 1에서 필드 디바이스(10)와 외부 플랫폼/어플리케이션(20)은 하나씩만 도시되어 있지만, 이는 도시와 설명의 편의를 위한 것에 불과하며, 실제로는 더 많은 수가 존재할 수 있다.
- [0032] 이는, 통합 게이트웨이(100)의 경우도 마찬가지이다. 즉, 통합 게이트웨이(100)를 다수로 구현하는 경우에도 본 발명의 기술적 사상이 적용될 수 있음은 물론이다.
- [0033] 본 발명의 실시예에 따른 통합 게이트웨이(100)는, 도시된 바와 같이, 에이전트(Agent)(110)와 OPC UA Aggregation 서버(140)를 포함하여 구성된다. 에이전트(110)는 다수 개가 생성될 수 있다.
- [0034] 통합 게이트웨이(100)는 초기화하면 OPC UA Aggregation 서버 인스턴스(140)를 생성하고, 시작, 정지 등의 라이프사이클을 관리한다.
- [0035] OPC UA Aggregation 서버(140)는 필드 디바이스(10)의 OPC UA 정보 모델들을 OPC UA 노드들로 등록하는 통합 게이트웨이(100)의 구성/모듈이다. 도시된 바와 같이, OPC UA 노드들은 OPC UA Aggregation 서버(140)에서 통합되어 계층화된 노드 셋으로 구성된다.
- [0036] OPC UA Aggregation 서버(140)는 OPC UA 노드들에 저장된 필드 디바이스(10)들의 데이터들을 이용하여, 외부 플랫폼/어플리케이션(20)의 OPC UA 클라이언트(25)의 요청에 따라 필드 디바이스(10)들의 데이터들을 제공하여 준다.
- [0037] 도 2에는 OPC UA Aggregation 서버(140)의 주요 기능 구현을 위한 프로그램 코드를 제시하였다.
- [0038] 통합 게이트웨이(100)는 연결이 필요한 필드 디바이스(10), 즉, OPC UA Aggregation 서버(140)에 등록된 OPC UA 노드들에 해당하는 필드 디바이스(10)의 OPC UA 서버(15)에 접속하기 위한 에이전트(110)를 생성한다.
- [0039] 에이전트(110)는 필드 디바이스(10)의 OPC UA 서버(15)에 연결하기 위한 OPC UA 클라이언트 인스턴스(130)를 통해 클라이언트의 생성, 연결, 연결 끊기 등의 라이프사이클을 관리하는 통합 게이트웨이(100)의 구성/모듈이다.
- [0040] 그리고, 에이전트(110)는 필드 디바이스(10)의 OPC UA 서버(15)로부터 데이터 변경 사항을 구독하기 위한 정보를 포함한다.
- [0041] 도 3에는 에이전트(110)의 주요 기능 구현을 위한 프로그램 코드를 제시하였다.
- [0042] 에이전트(110)는 필드 디바이스(10)들의 OPC UA 서버(15)들에 연결하기 위한 정보(remoteNodeId)들과 필드 디바이스(10)들의 OPC UA 노드들에 연결하기 위한 정보(localNodeId)들이 각각 매핑되어 있는 링크(120)들을 생성/관리한다.
- [0043] 에이전트(110)의 링크(120)에 의해, 필드 디바이스(10)의 데이터가 변경될 때마다, 변경된 값이 해당 OPC UA 노드에 반영될 수 있다. 이를 위해, 모니터링 대상 링크는 통합 게이트웨이(100)에 일괄 등록된다.

- [0044] 도 4에는 에이전트(110)의 링크(120)의 주요 기능 구현을 위한 프로그램 코드를 제시하였다.
- [0045] OPC UA 클라이언트(130)는 링크(120)들을 참조하여 필드 디바이스(10)의 OPC UA 서버(15)에 연결하여 데이터를 수신한다.
- [0046] 나아가, 통합 게이트웨이(100)를 구성하기 위해서는 다음의 파일들이 더 필요하다.
- [0047] 1) 연결하고자 하는 필드 디바이스(10)가 인증을 요구할 때 필요한 인증서와 개인키
- [0048] 2) 필드 디바이스(10)의 모델 정보 파일을 변환하여 생성된 OPC UA 정보 모델 파일(XML)
- [0049] 3) OPC UA SDK와 라이브러리
- [0050] 4) OPC UA SDK의 도구를 사용하여 OPC UA 정보 모델 파일을 기반으로 생성한 소스 코드와 헤더 파일
- [0051] 통합 게이트웨이(100)가 구성을 설정하고 동작하는 과정에 대해, 이하에서 도 5를 참조하여 상세히 설명한다. 도 5는 통합 게이트웨이(100)의 동작 프로세스를 나타낸 흐름도이다.
- [0052] 도시된 바와 같이, 먼저, 통합 게이트웨이(100)는 초기화 되면, OPC UA Aggregation 서버 인스턴스(140)를 생성한다(S210).
- [0053] OPC UA Aggregation 서버(140)는 필드 디바이스(10)들에서 생성된 OPC UA 정보 모델들을 OPC UA Aggregation 서버 인스턴스(140)에 OPC UA 노드들로 등록한다(S220).
- [0054] S220단계에서, OPC UA 노드들은 OPC UA Aggregation 서버(140)에서 통합되어 계층화된 노드 셋으로 구성된다.
- [0055] 다음, 통합 게이트웨이(100)는 OPC UA Aggregation 서버(140)에 등록된 OPC UA 노드들을 탐색하여, 연결이 필요한 필드 디바이스(10)의 OPC UA 서버(15)에 접속하기 위한 정보를 수집하고 에이전트(110)를 생성한다(S230).
- [0056] 이때, 필요한 접속 정보가 부족하거나 연결에 실패하면 해당 에이전트를 제거하고, 연결 가능한 에이전트가 하나도 없다면 에러와 함께 종료할 수 있다.
- [0057] 다음, 등록된 OPC UA 노드들을 탐색하여, 모니터링할 OPC UA 노드를 선정하여 해당 에이전트(110)의 링크(120)에 연결한다(S240).
- [0058] 이에 의해, 에이전트(110)의 링크(120)에는 필드 디바이스(10)들의 OPC UA 서버(15)들에 연결하기 위한 정보들과 필드 디바이스(10)들의 OPC UA 노드들에 연결하기 위한 정보들이 각각 매핑되게 된다.
- [0059] 또한, 필드 디바이스(10)로부터 데이터 변경 알림을 수신하면, 링크(120)에 매핑된 OPC UA 노드에 해당 데이터를 갱신하도록 에이전트(110)에 콜백을 설정한다(S250).
- [0060] 이때, single-thread 방식으로 동작하는 open62541 기반 OPC UA 서버의 특성 상, 서버 루프의 유희 시간 동안 에이전트에 등록된 콜백을 수행하도록 주기를 설정한다.
- [0061] 이후, 통합 게이트웨이(100)의 동작을 개시한다(S260). 이에 따라, 외부 플랫폼/어플리케이션(20)의 OPC UA 클라이언트(25)의 서비스 요청이 있으면, 그에 따라 해당 OPC UA 노드들의 데이터들을 전송한다(S270).
- [0062] 또한, S250단계에서 설정된 주기마다 필드 디바이스(10)의 데이터 변경 여부를 체크하고, 변경된 데이터를 OPC UA 노드에 갱신한다(S280).
- [0063] S270단계 및 S280단계는 통합 게이트웨이(100)가 종료할 때까지 반복된다(S290). 통합 게이트웨이(100)에 종료 시그널이 주입되면(S290-Y), 생성된 OPC UA Aggregation 서버 인스턴스(140), 에이전트(110), 링크(120) 및 할당 메모리를 정리하고 종료한다.
- [0064] 도 6에는 통합 게이트웨이의 실제 동작 프로세스를 예시한 도면이다.
- [0065] 지금까지, 효율적인 OPC UA 서버의 데이터 접근 및 통합을 위한 통합 게이트웨이에 대해 바람직한 실시예를 들어 상세히 설명하였다.
- [0066] 위 실시예에서는, 산업 현장에서 클라이언트 어플리케이션의 요청에 따라 복수개의 장비 내 OPC UA 서버들에게 연결하여 데이터를 효율적으로 취합, 전달 및 관리하는 통합 게이트웨이를 통해, 제조 현장 내 많은 수의 OPC UA 장치 및 시스템들을 안정적으로 연동할 수 있도록 하였다.
- [0067] 또한, 통합 게이트웨어 내에 내장된 OPC UA 클라이언트 인스턴스를 통해 클라이언트의 생성, 연결, 연결 끊기

등의 라이프사이클을 관리함으로써, 많은 수의 제조 디바이스 및 장치가 동시에 클라이언트 어플리케이션과 연결되어도 안정적인 서비스를 제공할 수 있도록 하였다.

[0068] 나아가, 통합 게이트웨이에 연결된 필드 디바이스 및 장비의 운영 데이터를 실시간 모니터링하고 업데이트 함으로써, 데이터 송수신의 신뢰도를 향상 시켰다.

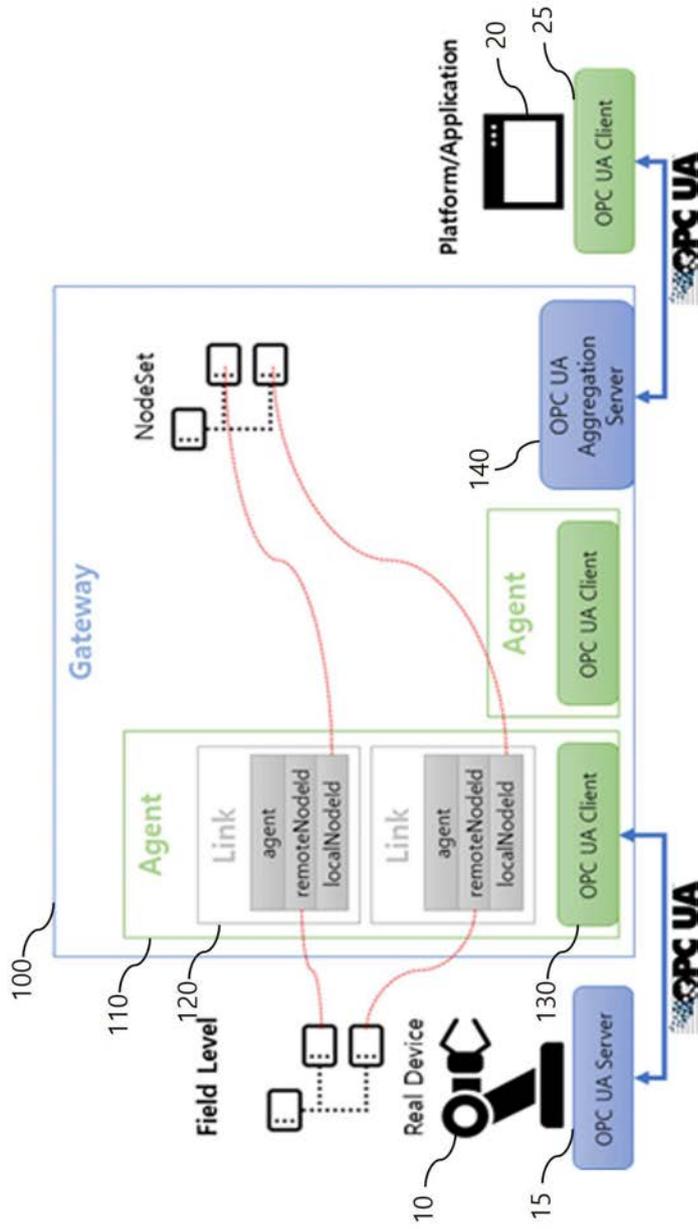
[0069] 한편, 본 실시예에 따른 장치와 방법의 기능을 수행하게 하는 컴퓨터 프로그램을 수록한 컴퓨터로 읽을 수 있는 기록매체에도 본 발명의 기술적 사상이 적용될 수 있음은 물론이다. 또한, 본 발명의 다양한 실시예에 따른 기술적 사상은 컴퓨터로 읽을 수 있는 기록매체에 기록된 컴퓨터로 읽을 수 있는 코드 형태로 구현될 수도 있다. 컴퓨터로 읽을 수 있는 기록매체는 컴퓨터에 의해 읽을 수 있고 데이터를 저장할 수 있는 어떤 데이터 저장 장치이더라도 가능하다. 예를 들어, 컴퓨터로 읽을 수 있는 기록매체는 ROM, RAM, CD-ROM, 자기 테이프, 플로피 디스크, 광디스크, 하드 디스크 드라이브, 등이 될 수 있음은 물론이다. 또한, 컴퓨터로 읽을 수 있는 기록매체에 저장된 컴퓨터로 읽을 수 있는 코드 또는 프로그램은 컴퓨터간에 연결된 네트워크를 통해 전송될 수도 있다.

[0070] 또한, 이상에서는 본 발명의 바람직한 실시예에 대하여 도시하고 설명하였지만, 본 발명은 상술한 특정의 실시예에 한정되지 아니하며, 청구범위에서 청구하는 본 발명의 요지를 벗어남이 없이 당해 발명이 속하는 기술분야에서 통상의 지식을 가진자에 의해 다양한 변형실시가 가능한 것은 물론이고, 이러한 변형실시들은 본 발명의 기술적 사상이나 전망으로부터 개별적으로 이해되어져서는 안될 것이다.

부호의 설명

- [0072] 10 : 필드 디바이스
- 15 : OPC UA 서버
- 20 : 외부 플랫폼/어플리케이션
- 25 : OPC UA 클라이언트
- 100 : 통합 게이트웨이
- 110 : 에이전트(Agent)
- 120 : 링크
- 130 : OPC UA 클라이언트
- 140 : OPC UA Aggregation 서버

도면
도면1



도면2

```

typedef struct _AAS_Gateway_ {
    UA_Boolean running;
    UA_Server *server;
    UA_ServerConfig *config;
} AAS_Gateway;

UA_StatusCode AAS_Gateway_initialize(UA_UInt16 portNumber);
void AAS_Gateway_finalize();

UA_StatusCode AAS_Gateway_start();
void AAS_Gateway_stop();

UA_Server *AAS_Gateway_getServer();
UA_StatusCode AAS_Gateway_setupNodes();
UA_StatusCode AAS_Gateway_writeValue(
    const UA_NodeId nodeId, const UA_Variant value
);
    
```

도면3

```

typedef struct _AAS_Agent_ {
    LIST_ENTRY(_AAS_Agent_) next;
    UA_String id;
    UA_String endpointUrl;
    UA_String username;
    UA_String password;
    UA_Client *client;
    UA_ClientConfig *config;
    UA_UInt32 subscriptionId;
} AAS_Agent;

typedef struct _AAS_AgentParam_ {
    const UA_String *certificate;
    const UA_String *privateKey;
    const UA_String *username;
    const UA_String *password;
} AAS_AgentParam;

AAS_Agent *AAS_Agent_create(
    const UA_String *id, const UA_String *endpointUrl,
    const UA_String *securityMode,
    const UA_String *securityPolicy,
    const AAS_AgentParam *param
);
void AAS_Agent_destroy(AAS_Agent *agent);

UA_StatusCode AAS_Agent_connect(AAS_Agent *agent);
UA_StatusCode AAS_Agent_disconnect(AAS_Agent *agent);
    
```

도면4

```

typedef struct _AAS_Link_ {
    LIST_ENTRY(_AAS_Link_) next;
    AAS_Agent *agent;
    UA_UInt32 monitoredItemId;
    struct {
        UA_NodeId local;
        UA_NodeId remote;
    } nodeId;
} AAS_Link;

AAS_Link *AAS_Link_create(
    AAS_Agent *agent, UA_NodeId *remoteNodeId,
    UA_NodeId *localNodeId
);

void AAS_Link_destroy(AAS_Link *link);
    
```

도면5

