Jean Monnet University

Ecole de Mines Saint Etienne

M1 Master Internship Report

Industrial Control with Knowledge Graphs

Submitted by

Sivaratnam PACHAVA

May 2022 - August 2022

Cyber-Physical Social Systems (CPS2)

Internship Supervisor

Victor CHARPENAY

associate professor in Computer Science, Mines Saint Etienne

Academic Supervisor

Pierre Maret

Professor in Computer Science, Université Jean Monnet, Saint Etienne

Abstract

The objective of SIRAM project (Integrated Systems for Mobile Assistant Robots) a regional project involving Meca-concept, Creative'IT and Mines Saint- Etienne is to develop an interoperable, adaptive information system that integrates mobile assistant robots (RAM) in the Industry 4.0 environment of IT'm Factory. The prototype developed in SIRAM aims at showing how a control system can automatically adapt to contextual evolution and deal with heterogeneous objects including production equipment equipped with a pre-programmed industrial controller, low-power connected devices mounted on that equipment and industrial robotic arms on the same factory floor. To accomplish this adaptive system, they decided to use a multi-agent oriented perspective that encapsulating the entities in charge of the management of shared storage and/or processing resources within the environment, into autonomous agents capable of perceiving, deciding and acting on the physical environment, interacting with each other and participating in a coordinated way to global industrial processes. In my internship period I developed a "KNOWLEDGE GRAPH" that describes the data and actions of the entities involved in industrial processes. The agents will use these entities involved in KG.

This report takes us through all the details of my completed work during this internship period.

Contents

List of Figures

1 Introduction

This report presents the work done during the internship project with the Ecole des Mines de Saint Etienne. The goal of this internship project is to develop a knowledge graph for the description of the machines. The main tasks are as follow,

1. Understand the content of a knowledge graph in RDF
2. Extract relevant data from a knowledge graph
3. Exploit a distributed knowledge graph using a linked data approach
4. Add flexibility by exploiting the KG and ontologies

By rapidly increasing of an Industrial Internet of Things (IIoT) and the connectivity of industrial equipment (sensors and actuators) now industrial control is at the intersection between information technologies (IT) and operational technologies (OT). Industrial processes can be controlled with agility and efficiency by remote software components. To implement certain services and reach autonomous decisions the Internet of Things (IoT) allows communication and exchange of data between physical objects. To achieve self-controlled and self-optimizing processes the idea of Industry 4.0 is similarly based on the close interaction of decentralized systems such as production systems. Big Data comes into play due to the huge amount of different kinds of data's are continuously generated, exchanged, and to be processed. This data has to be standardized to enable their autonomous decisions. Also, the different kinds of data can be collected, transformed, and integrated to support a entire analysis and optimization of the different production lines. So, we are focus on knowledge graphs because of that can semantically interrelate many entities of different types for data analysis. knowledge graphs are particular kinds of databases designed to capture knowledge from various sources, represented as a set of interlinked entities.

The application scenario that we are use in this challenge is a manufacturing product line in the Industry 4.0 for filling and packaging yogurt[1]. To achieve the goal in this scenario, the factory has different kinds of workshops which have different functions: conveying workshop, filling workshop, potting workshop and the packaging workshop.

Thing is an abstraction of physical or virtual entity that gives interactions to participate in the web of things. Each workshop contains different Things that can be used to sense the current state of the working process using sensors and perform actuations on actuators. The factory also includes external suppliers those are responsible for providing empty yogurt cups, yogurt supplies and package providers.

In next sections I will explain about how these different workshops will work to achieve goal of filling and packaging yogurt and how I created Knowledge Graph for those machines.

2 Objective of the internship

The main objective of the internship is to extend an existing Knowledge Graph (KG) that describes the IT'm Factory, such that a remote agent-based control system can observe the real time state of the factory and acts on it in a unified manner. In addition to the directly stated knowledge, Knowledge Graphs also include logical rules to derive indirect knowledge, which makes them a suitable extraction to represent factory lines in Industry 4.0 scenarios. Industrial equipment, sensors and actuators are represented as entities that are related to each other via spatial relationships ('is adjacent to', 'contains', etc.) and structural relationships in terms of system engineering ('is part of', 'connects to'). The existing KG for the IT'm factory already includes these entities, but it is not sufficient for industrial control. Specifically, it does not contain interdependencies between the different properties of the overall production system. So, the main objective of the internship is to extend the KG, such that the full state of the system can be obtained from an evaluation of its observable properties on the overall machines.

3 Context

In this section I will explain about the knowledge graph, Semantic Web, PLC Tags and some tools which i have used in my work to create sufficient knowledge Graph of SIRAM.

3.1 Knowledge Graph

Knowledge Graphs (KG) and other semantic technologies have become a usable option for companies to represent complex information in a meaningful and understandable data structure in the form of a graph. It represents a collection of entities

[1] https://gitlab.emse.fr/ai4industry/hackathon/-/wikis/scenario/

that are described with interlinked properties, objects, events, or concepts. The data together form a connected graph and put data in context via linking with semantic metadata. The links represent structured metadata that enables better search over graphs and analytics. Entities can be enriched with more linked properties that makes KG much more valuable for analysis, visualization, and reporting.

This provides a framework for data integrity, analytics, and reasoning capability over data. The semantic representation of data can improve understandability of complex data. It helps to making development of new technologies more efficient[3], and improves the quality control in manufacturing processes[10]. Not only software companies like Facebook, Google and Microsoft, but also production companies like Siemens[8] turn towards semantic representations of their data. Aibel, a service company in the energy sector, has reportedly saved more than 100 million Euros through better representation of their products using ontologies[9].

The standard that is used to create KGs with their entities and relationships is called RDF (Resource Description Framework), which is a recommendation of the W3C (World Wide Web Consortium). An RDF graph is a set of triples, each comprised of a subject, a predicate, and an object. An object will be connected to a subject using a predicate. Using such triples we can make statements about entities and their relations. A set of statements can be created using such a relation. This structured representation of the data can help to represent the relation between disability document contents and glossary terms. Also, it is possible to link documents with regions, countries, subtopics, and so on.

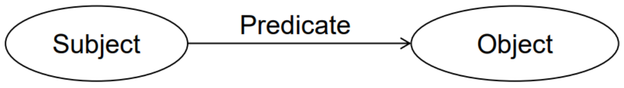


Figure 1: RDF Graph consist of single triple with subject, predicate and object

An RDF Graph have three different kinds of nodes: IRIs (Internationalized Resource identifiers), literals or blank nodes. IRIs are generalizations of URIs and they give each resource as a unique identifier. To represents values such as strings, dates or numbers literals are used. RDF allows the user to indicate the datatype and if the literal is a string a language tag can be provided. Blank nodes are anonymous resources, that enable more complex structures.

3.2 PLC and HMI

PLC stands for Programmable Logic Controller. PLCs are using to replace hard-coded relay systems with more flexible programmable controllers. PLCs are used in a variety of industries because they're fast, easy to operate and are considered easy to program. PLCs are used in industrial automation to increased automation process and to increase reliability, system stability and performance in the control of industrial processes, minimizing the need for human operators and the chances of human errors. In our Machines PLCs are programmed in ladder logic.

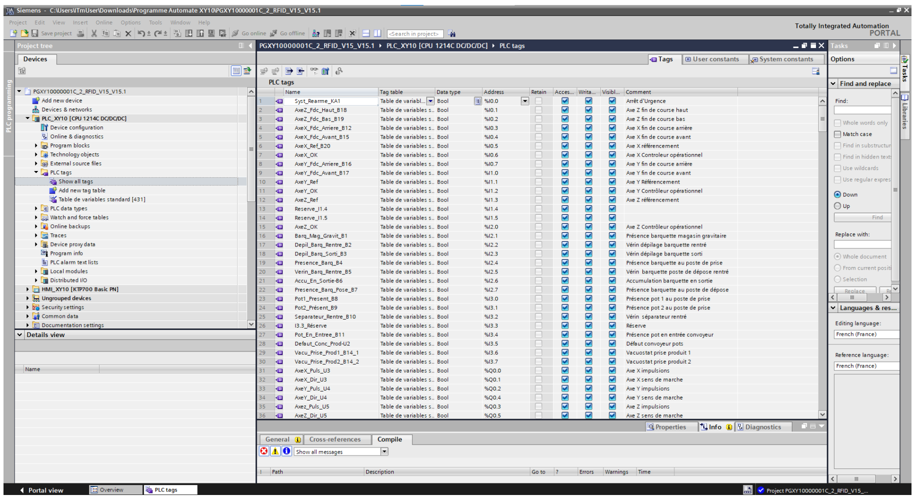
HMI systems enable users to view data from the manufacturing floor and provide an interface for users to provide control input and PLCs are an essential hardware component element in these systems. PLC's act as the physical interfaces between devices on the manufacturing floor and a HMI system. PLCs communicate, monitor and control automated processes like machine functions, or robotic devices.

3.3 TIA Portal, PLC and HMI Tags

Totally Integrated Automation (TIA) Portal provides an engineering framework for implementing automation solutions in all industries, and includes a variety of user-friendly functions. TIA Portal encloses several software packages to efficiently program various Siemens automation and drive components. SIMATIC STEP 7 in the TIA Portal is the software for the configure, program, test and verify all modular and PC-based SIMATIC controllers. We are using TIA Portal in our production process to connect to machine PLCs.

Tag is a name you assign to an address of PLC. In PLC designs, a more structured programming approach is taken with the use of "tags," which are stored within the PLC's memory in a tag database. PLC Tags are names that you assign to variables of any type stored in the PLC memory.

Human-Machine Interface (HMI) is a user interface that connects a person to a machine or system and optimize their industrial processes. HMIs communicate with Programmable Logic Controllers (PLCs) and input/output sensors to get and display information for users to view. The HMI tag can technically be applied to any screen that allows a user to interact with a device.

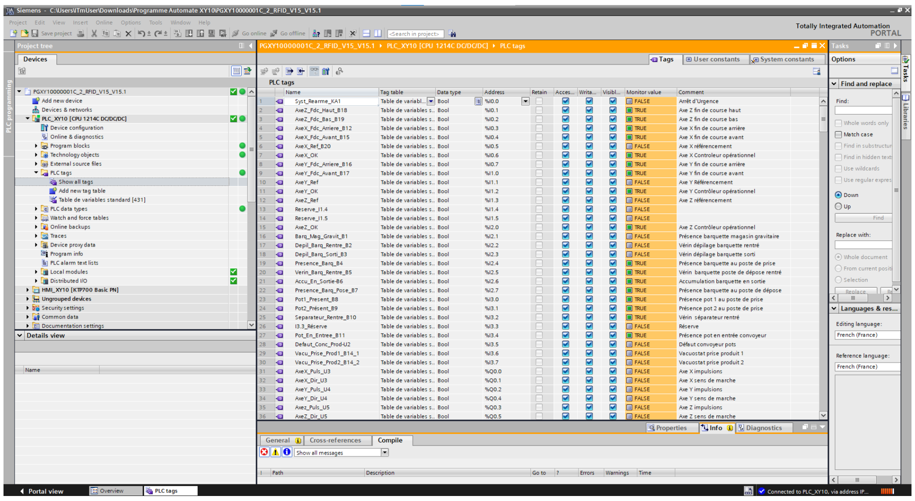


PGXY10000001C_2_RFID_V15_V15.1 ▸ PLC_XY10 [CPU 1214C DC/DC/DC] ▸ PLC tags

PLC tags

	Name	Tag table	Data type	Address	Retain	Acces...	Writa...	Visibl...	Comment
1	Syst_Rearme_KA1	Table de variabl...	Bool	%I0.0		✓	✓	✓	Arrêt d'Urgence
2	AxeZ_Fdc_Haut_B18	Table de variables s...	Bool	%I0.1		✓	✓	✓	Axe Z fin de course haut
3	AxeZ_Fdc_Bas_B19	Table de variables s...	Bool	%I0.2		✓	✓	✓	Axe Z fin de course bas
4	AxeX_Fdc_Arriere_B12	Table de variables s...	Bool	%I0.3		✓	✓	✓	Axe X fin de course arrière
5	AxeX_Fdc_Avant_B15	Table de variables s...	Bool	%I0.4		✓	✓	✓	Axe X fin de course avant
6	AxeX_Ref_B20	Table de variables s...	Bool	%I0.5		✓	✓	✓	Axe X référencement
7	AxeX_OK	Table de variables s...	Bool	%I0.6		✓	✓	✓	Axe X Controleur opérationnel
8	AxeY_Fdc_Arriere_B16	Table de variables s...	Bool	%I0.7		✓	✓	✓	Axe Y fin de course arrière
9	AxeY_Fdc_Avant_B17	Table de variables s...	Bool	%I1.0		✓	✓	✓	Axe Y fin de course avant
10	AxeY_Ref	Table de variables s...	Bool	%I1.1		✓	✓	✓	Axe Y Référencement
11	AxeY_OK	Table de variables s...	Bool	%I1.2		✓	✓	✓	Axe Y Contrôleur opérationnel
12	AxeZ_Ref	Table de variables s...	Bool	%I1.3		✓	✓	✓	Axe Z référencement
13	Reserve_I1.4	Table de variables s...	Bool	%I1.4		✓	✓	✓	
14	Reserve_I1.5	Table de variables s...	Bool	%I1.5		✓	✓	✓	
15	AxeZ_OK	Table de variables s...	Bool	%I2.0		✓	✓	✓	Axe Z Contrôleur opérationnel
16	Barq_Mag_Gravit_B1	Table de variables s...	Bool	%I2.1		✓	✓	✓	Présence barquette magasin gravitaire
17	Depil_Barq_Rentre_B2	Table de variables s...	Bool	%I2.2		✓	✓	✓	Vérin dépilage barquette rentré
18	Depil_Barq_Sorti_B3	Table de variables s...	Bool	%I2.3		✓	✓	✓	Vérin dépilage barquette sorti
19	Presence_Barq_B4	Table de variables s...	Bool	%I2.4		✓	✓	✓	Présence barquette au poste de prise
20	Verin_Barq_Rentre_B5	Table de variables s...	Bool	%I2.5		✓	✓	✓	Vérin barquette poste de dépose rentré
21	Accu_En_Sortie-B6	Table de variables s...	Bool	%I2.6		✓	✓	✓	Accumulation barquette en sortie
22	Presence_Barq_Pose_B7	Table de variables s...	Bool	%I2.7		✓	✓	✓	Présence barquette au poste de dépose
23	Pot1_Present_B8	Table de variables s...	Bool	%I3.0		✓	✓	✓	Présence pot 1 au poste de prise
24	Pot2_Présent_B9	Table de variables s...	Bool	%I3.1		✓	✓	✓	Présence pot 2 au poste de prise
25	Separateur_Rentre_B10	Table de variables s...	Bool	%I3.2		✓	✓	✓	Vérin séparateur rentré
26	I3.3_Réserve	Table de variables s...	Bool	%I3.3		✓	✓	✓	Réserve
27	Pot_En_Entree_B11	Table de variables s...	Bool	%I3.4		✓	✓	✓	Présence pot en entrée convoyeur
28	Defaut_Conc_Prod-U2	Table de variables s...	Bool	%I3.5		✓	✓	✓	Défaut convoyeur pots
29	Vacu_Prise_Prod1_B14_1	Table de variables s...	Bool	%I3.6		✓	✓	✓	Vacuostat prise produit 1
30	Vacu_Prise_Prod2_B14_2	Table de variables s...	Bool	%I3.7		✓	✓	✓	Vacuostat prise produit 2
31	AxeX_Puls_U3	Table de variables s...	Bool	%Q0.0		✓	✓	✓	Axe X impulsions
32	AxeX_Dir_U3	Table de variables s...	Bool	%Q0.1		✓	✓	✓	Axe X sens de marche
33	AxeY_Puls_U4	Table de variables s...	Bool	%Q0.2		✓	✓	✓	Axe Y impulsions
34	AxeY_Dir_U4	Table de variables s...	Bool	%Q0.4		✓	✓	✓	Axe Y sens de marche
35	Axez_Puls_U5	Table de variables s...	Bool	%Q0.3		✓	✓	✓	Axe Z impulsions
36	AxeZ_Dir_U5	Table de variables s...	Bool	%Q0.5		✓	✓	✓	Axe Z sens de marche

Figure 2: PLC Tags in TIA Portal



PGXY10000001C_2_RFID_V15_V15.1 ▸ PLC_XY10 [CPU 1214C DC/DC/DC] ▸ PLC tags

PLC tags

	Name	Tag table	Data type	Address	Retain	Acces...	Writa...	Visibl...	Monitor value	Comment
1	Syst_Rearme_KA1	Table de variabl...	Bool	%I0.0		✓	✓	✓	FALSE	Arrêt d'Urgence
2	AxeZ_Fdc_Haut_B18	Table de variables s...	Bool	%I0.1		✓	✓	✓	TRUE	Axe Z fin de course haut
3	AxeZ_Fdc_Bas_B19	Table de variables s...	Bool	%I0.2		✓	✓	✓	TRUE	Axe Z fin de course bas
4	AxeX_Fdc_Arriere_B12	Table de variables s...	Bool	%I0.3		✓	✓	✓	TRUE	Axe X fin de course arrière
5	AxeX_Fdc_Avant_B15	Table de variables s...	Bool	%I0.4		✓	✓	✓	TRUE	Axe X fin de course avant
6	AxeX_Ref_B20	Table de variables s...	Bool	%I0.5		✓	✓	✓	FALSE	Axe X référencement
7	AxeX_OK	Table de variables s...	Bool	%I0.6		✓	✓	✓	TRUE	Axe X Controleur opérationnel
8	AxeY_Fdc_Arriere_B16	Table de variables s...	Bool	%I0.7		✓	✓	✓	TRUE	Axe Y fin de course arrière
9	AxeY_Fdc_Avant_B17	Table de variables s...	Bool	%I1.0		✓	✓	✓	TRUE	Axe Y fin de course avant
10	AxeY_Ref	Table de variables s...	Bool	%I1.1		✓	✓	✓	FALSE	Axe Y Référencement
11	AxeY_OK	Table de variables s...	Bool	%I1.2		✓	✓	✓	TRUE	Axe Y Contrôleur opérationnel
12	AxeZ_Ref	Table de variables s...	Bool	%I1.3		✓	✓	✓	FALSE	Axe Z référencement
13	Reserve_I1.4	Table de variables s...	Bool	%I1.4		✓	✓	✓	FALSE	
14	Reserve_I1.5	Table de variables s...	Bool	%I1.5		✓	✓	✓	FALSE	
15	AxeZ_OK	Table de variables s...	Bool	%I2.0		✓	✓	✓	TRUE	Axe Z Contrôleur opérationnel
16	Barq_Mag_Gravit_B1	Table de variables s...	Bool	%I2.1		✓	✓	✓	FALSE	Présence barquette magasin gravitaire
17	Depil_Barq_Rentre_B2	Table de variables s...	Bool	%I2.2		✓	✓	✓	FALSE	Vérin dépilage barquette rentré
18	Depil_Barq_Sorti_B3	Table de variables s...	Bool	%I2.3		✓	✓	✓	FALSE	Vérin dépilage barquette sorti
19	Presence_Barq_B4	Table de variables s...	Bool	%I2.4		✓	✓	✓	TRUE	Présence barquette au poste de prise
20	Verin_Barq_Rentre_B5	Table de variables s...	Bool	%I2.5		✓	✓	✓	TRUE	Vérin barquette poste de dépose rentré
21	Accu_En_Sortie-B6	Table de variables s...	Bool	%I2.6		✓	✓	✓	TRUE	Accumulation barquette en sortie
22	Presence_Barq_Pose_B7	Table de variables s...	Bool	%I2.7		✓	✓	✓	TRUE	Présence barquette au poste de dépose
23	Pot1_Present_B8	Table de variables s...	Bool	%I3.0		✓	✓	✓	TRUE	Présence pot 1 au poste de prise
24	Pot2_Présent_B9	Table de variables s...	Bool	%I3.1		✓	✓	✓	TRUE	Présence pot 2 au poste de prise
25	Separateur_Rentre_B10	Table de variables s...	Bool	%I3.2		✓	✓	✓	TRUE	Vérin séparateur rentré
26	I3.3_Réserve	Table de variables s...	Bool	%I3.3		✓	✓	✓	FALSE	Réserve
27	Pot_En_Entree_B11	Table de variables s...	Bool	%I3.4		✓	✓	✓	TRUE	Présence pot en entrée convoyeur
28	Defaut_Conc_Prod-U2	Table de variables s...	Bool	%I3.5		✓	✓	✓	FALSE	Défaut convoyeur pots
29	Vacu_Prise_Prod1_B14_1	Table de variables s...	Bool	%I3.6		✓	✓	✓	FALSE	Vacuostat prise produit 1
30	Vacu_Prise_Prod2_B14_2	Table de variables s...	Bool	%I3.7		✓	✓	✓	FALSE	Vacuostat prise produit 2
31	AxeX_Puls_U3	Table de variables s...	Bool	%Q0.0		✓	✓	✓	FALSE	Axe X impulsions
32	AxeX_Dir_U3	Table de variables s...	Bool	%Q0.1		✓	✓	✓	FALSE	Axe X sens de marche
33	AxeY_Puls_U4	Table de variables s...	Bool	%Q0.2		✓	✓	✓	FALSE	Axe Y impulsions
34	AxeY_Dir_U4	Table de variables s...	Bool	%Q0.4		✓	✓	✓	FALSE	Axe Y sens de marche
35	Axez_Puls_U5	Table de variables s...	Bool	%Q0.3		✓	✓	✓	FALSE	Axe Z impulsions
36	AxeZ_Dir_U5	Table de variables s...	Bool	%Q0.5		✓	✓	✓	FALSE	Axe Z sens de marche

Connected to PLC_XY10, via address IP...

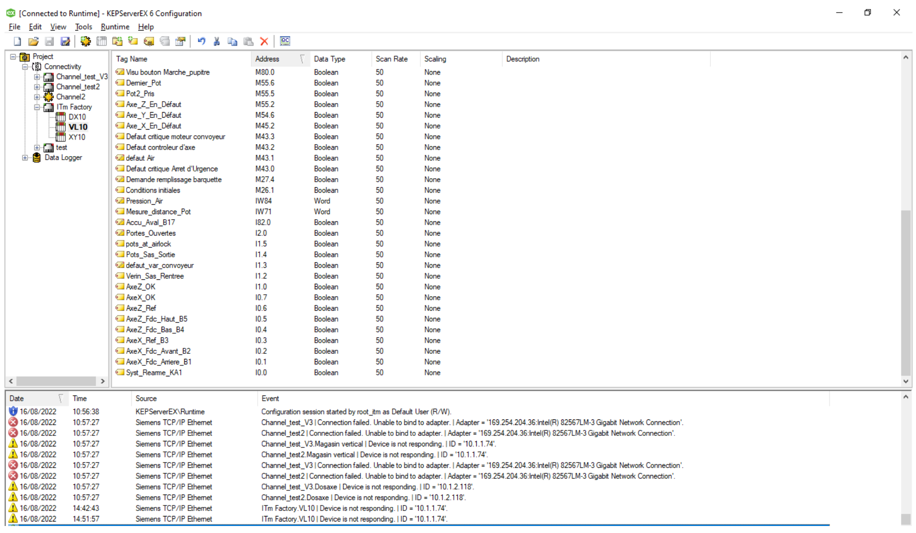
Figure 3: To read real time tag values from the PLC in TIA Portal

3.4 OPC-UA and Kep Server

Kep Server is a windows application that provides a way of bringing data and information from a wide range of industrial devices and systems into client applications on your windows PC. This platform design allows user to connect, manage, monitor, and control industrial automation devices and software applications. KEP Server provides data access for client applications and IoT via OPC protocols.

OPC stands for Open Platform Communications and is one of the most important communication standards for Industry 4.0 and the IoT.

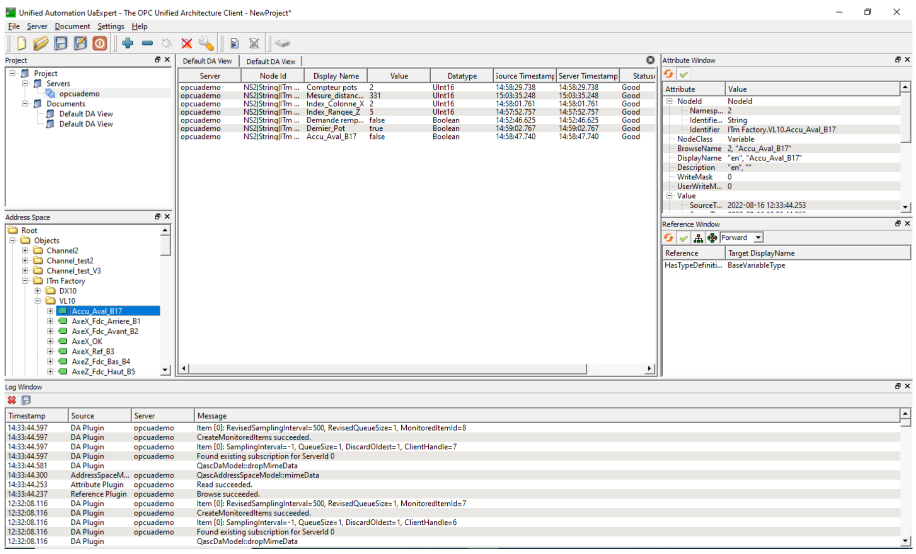
OPC is the leading standard for industrial automation connectivity. KEP Server supports the OPC Unified Architecture (OPC UA) specification and many of the OPC Classic specifications, including OPC Data Access (OPC DA), OPC Alarms and Events (OPC AE), and OPC Historical Data Access (OPC HDA).



Tag Name	Address	Data Type	Scan Rate	Scaling	Description
Visu bouton Marche_pupitre	M80.0	Boolean	50	None	
Dernier_Pot	M55.6	Boolean	50	None	
Pot2_Pris	M55.5	Boolean	50	None	
Axe_Z_En_Défaut	M55.2	Boolean	50	None	
Axe_Y_En_Défaut	M54.6	Boolean	50	None	
Axe_X_En_Défaut	M45.2	Boolean	50	None	
Defaut critique moteur convoyeur	M43.3	Boolean	50	None	
Defaut controleur d'axe	M43.2	Boolean	50	None	
defaut Air	M43.1	Boolean	50	None	
Defaut critique Arret d'Urgence	M43.0	Boolean	50	None	
Demande remplissage barquette	M27.4	Boolean	50	None	
Conditions initiales	M26.1	Boolean	50	None	
Pression_Air	IW84	Word	50	None	
Mesure_distance_Pot	IW71	Word	50	None	
Accu_Aval_B17	I82.0	Boolean	50	None	
Portes_Ouvertes	I2.0	Boolean	50	None	
pots_at_airlock	I1.5	Boolean	50	None	
Pots_Sas_Sortie	I1.4	Boolean	50	None	
defaut_var_convoyeur	I1.3	Boolean	50	None	
Verin_Sas_Rentree	I1.2	Boolean	50	None	
AxeZ_OK	I1.0	Boolean	50	None	
AxeX_OK	I0.7	Boolean	50	None	
AxeZ_Ref	I0.6	Boolean	50	None	
AxeZ_Fdc_Haut_B5	I0.5	Boolean	50	None	
AxeZ_Fdc_Bas_B4	I0.4	Boolean	50	None	
AxeX_Ref_B3	I0.3	Boolean	50	None	
AxeX_Fdc_Avant_B2	I0.2	Boolean	50	None	
AxeX_Fdc_Arriere_B1	I0.1	Boolean	50	None	
Syst_Rearme_KA1	I0.0	Boolean	50	None	

```
Date         Time      Source                   Event
16/08/2022   10:56:38  KEPServerEX\Runtime      Configuration session started by root_itm as Default User (R/W).
16/08/2022   10:57:27  Siemens TCP/IP Ethernet  Channel_test_V3 | Connection failed. Unable to bind to adapter. | Adapter = '169.254.204.36:Intel(R) 82567LM-3 Gigabit Network Connection'.
16/08/2022   10:57:27  Siemens TCP/IP Ethernet  Channel_test2 | Connection failed. Unable to bind to adapter. | Adapter = '169.254.204.36:Intel(R) 82567LM-3 Gigabit Network Connection'.
16/08/2022   10:57:27  Siemens TCP/IP Ethernet  Channel_test_V3.Magasin vertical | Device is not responding. | ID = '10.1.1.74'.
16/08/2022   10:57:27  Siemens TCP/IP Ethernet  Channel_test2.Magasin vertical | Device is not responding. | ID = '10.1.1.74'.
16/08/2022   10:57:27  Siemens TCP/IP Ethernet  Channel_test_V3 | Connection failed. Unable to bind to adapter. | Adapter = '169.254.204.36:Intel(R) 82567LM-3 Gigabit Network Connection'.
16/08/2022   10:57:27  Siemens TCP/IP Ethernet  Channel_test2 | Connection failed. Unable to bind to adapter. | Adapter = '169.254.204.36:Intel(R) 82567LM-3 Gigabit Network Connection'.
16/08/2022   10:57:27  Siemens TCP/IP Ethernet  Channel_test_V3.Dosaxe | Device is not responding. | ID = '10.1.2.118'.
16/08/2022   10:57:27  Siemens TCP/IP Ethernet  Channel_test2.Dosaxe | Device is not responding. | ID = '10.1.2.118'.
16/08/2022   14:42:43  Siemens TCP/IP Ethernet  ITm Factory.VL10 | Device is not responding. | ID = '10.1.1.74'.
16/08/2022   14:51:57  Siemens TCP/IP Ethernet  ITm Factory.VL10 | Device is not responding. | ID = '10.1.1.74'.
```

Figure 4: Tags in KEP Server



Server	Node Id	Display Name	Value	Datatype	Source Timestamp	Server Timestamp	Status
opcuademo	NS2|String|ITm ...	Compteur pots	2	UInt16	14:58:29.738	14:58:29.738	Good
opcuademo	NS2|String|ITm ...	Mesure_distanc...	331	UInt16	15:03:35.248	15:03:35.248	Good
opcuademo	NS2|String|ITm ...	Index_Colonne_X	2	UInt16	14:58:01.761	14:58:01.761	Good
opcuademo	NS2|String|ITm ...	Index_Rangee_Z	5	UInt16	14:57:52.757	14:57:52.757	Good
opcuademo	NS2|String|ITm ...	Demande remp...	false	Boolean	14:52:46.625	14:52:46.625	Good
opcuademo	NS2|String|ITm ...	Dernier_Pot	true	Boolean	14:59:02.767	14:59:02.767	Good
opcuademo	NS2|String|ITm ...	Accu_Aval_B17	false	Boolean	14:58:47.740	14:58:47.740	Good

Attribute	Value
NodeId	NodeId
Namesp...	2
Identifie...	String
Identifier	ITm Factory.VL10.Accu_Aval_B17
NodeClass	Variable
BrowseName	2, "Accu_Aval_B17"
DisplayName	"en", "Accu_Aval_B17"
Description	"en", ""
WriteMask	0
UserWriteM...	0
Value	
SourceT...	2022-08-16 12:33:44.253

Reference	Target DisplayName
HasTypeDefiniti...	BaseVariableType

```
Timestamp     Source             Server     Message
14:33:44.597  DA Plugin          opcuademo  Item [0]: RevisedSamplingInterval=500, RevisedQueueSize=1, MonitoredItemId=8
14:33:44.597  DA Plugin          opcuademo  CreateMonitoredItems succeeded.
14:33:44.597  DA Plugin          opcuademo  Item [0]: SamplingInterval=-1, QueueSize=1, DiscardOldest=1, ClientHandle=7
14:33:44.597  DA Plugin          opcuademo  Found existing subscription for ServerId 0
14:33:44.581  DA Plugin                     QascDaModel::dropMimeData
14:33:44.300  AddressSpaceM...   opcuademo  QascAddressSpaceModel::mimeData
14:33:44.253  Attribute Plugin   opcuademo  Read succeeded.
14:33:44.237  Reference Plugin   opcuademo  Browse succeeded.
12:32:08.116  DA Plugin          opcuademo  Item [0]: RevisedSamplingInterval=500, RevisedQueueSize=1, MonitoredItemId=7
12:32:08.116  DA Plugin          opcuademo  CreateMonitoredItems succeeded.
12:32:08.116  DA Plugin          opcuademo  Item [0]: SamplingInterval=-1, QueueSize=1, DiscardOldest=1, ClientHandle=6
12:32:08.116  DA Plugin          opcuademo  Found existing subscription for ServerId 0
12:32:08.116  DA Plugin                     QascDaModel::dropMimeData
```

Figure 5: Reading Real time values of the tags in OPC-UA Client

3.5 Semantic Web

The Semantic Web, the Web of Data, the Linked Data represents the next major evolution in connecting and representing information. The goal of the Semantic Web is to make internet data machine-readable. It enables data to be linked from a source to any other source and it to be understood by computers, so that they can perform increasingly advanced tasks. Semantic Web technologies enable people to create data stores on the Web, build vocabularies, and write rules for handling data. W3C offers different technologies to describe and define different forms of vocabularies in a standard format. These includes Resource Description Framework (RDF) and RDF Schemas, SPARQL (SPARQL Protocol and RDF Query Language) and Web Ontology Language (OWL). The choice among these different Semantic Web languages depends on the complexity and accuracy required by a specific application.

1. RDF - The data modelling language for the Semantic Web. All Semantic Web information is stored and represented in the RDF.

2. SPARQL – It is the semantic query language specifically designed to query data across various systems and databases, and to retrieve and process data stored in RDF format.

3. OWL - The schema language, or knowledge representation (KR) language, of the Semantic Web. OWL enables you to define concepts composably so that these concepts can be reused as much and as often as possible. Composability

means that each concept is carefully defined so that it can be selected and assembled in various combinations with other concepts as needed for many different applications and purposes.

3.6 Ontology

An ontology is a formal description of knowledge using machine processable specifications. These specifications have well defined meanings and contain known concepts and relationships[4]. The word Ontology is used for quite formal collection of terms. Ontologies build on description logic, which allows reasoning engines to check logical consistency and correctness. Such reasoning possibilities are advantageous in the Industry 4.0 setting to make implicit information explicit. Some applications may need more complex ontologies with complex reasoning procedures. It all depends on the requirements and the goals of the applications.

Ontologies provide a so-called vocabulary, which is a set of IRIs, that can be used in RDF graphs. For example we use the RDF vocabulary, by utilizing rdf:type to express that an entity is an instance of a class. Adding vocabularies is a common technique to already existing ontologies and makes integration of different semantic systems easier.

On the Semantic Web, Vocabularies define the concepts and relationships used to represent knowledge about things and the relationship between things. Vocabularies can be very complex or very simple depending upon the terms and concepts used to describe them on an area of concern. Vocabularies are the basic components for inference techniques on the Semantic Web. The role of vocabularies on the Semantic Web are to help data integration if uncertainty exist on the terms used in the different data sets, or when a bit of extra knowledge may need to the discovery of new relationships. It depends on the application how complex vocabularies they use. Some application may choose to use very simple vocabularies. We are using the below ontologies in our knowledge graphs to represent our source information.

1. The W3C WoT TD[5] for describing device capabilities and their interaction affordances
2. The W3C SSN[2] and SOSA ontology for describing sensor and observation
3. product type ontology
4. custom ontology for describing scenario specific vocabularies
5. Hypermedia Controls Ontology[3] for describing links and forms

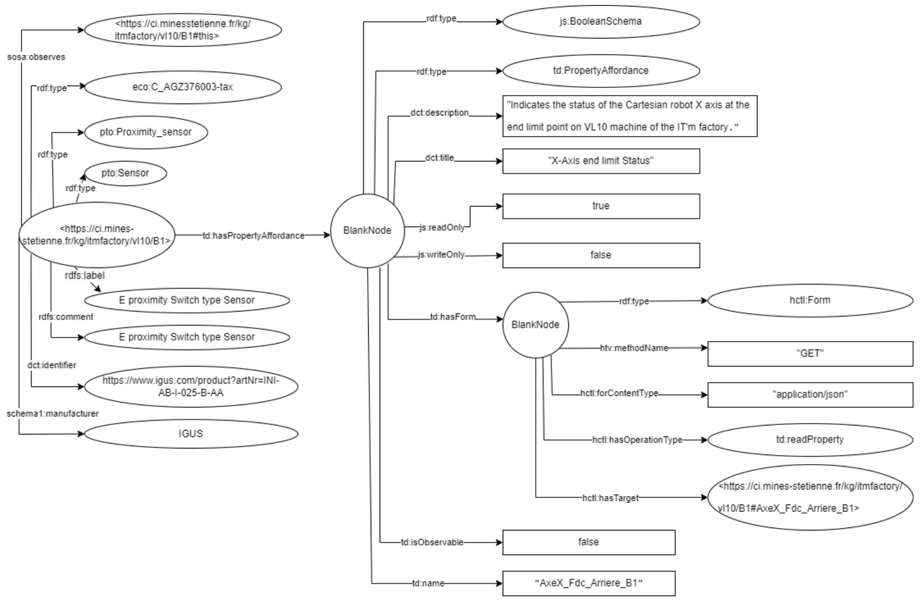


Figure 6: Graph to represent some entities in the ontologies

[2]https://www.w3.org/TR/vocab-ssn/
[3]https://www.w3.org/2019/wot/hypermedia

4 Architecture

The architecture represents the planning of work of creating knowledge graph and how the agents will use this knowledge graph to control industrial process.

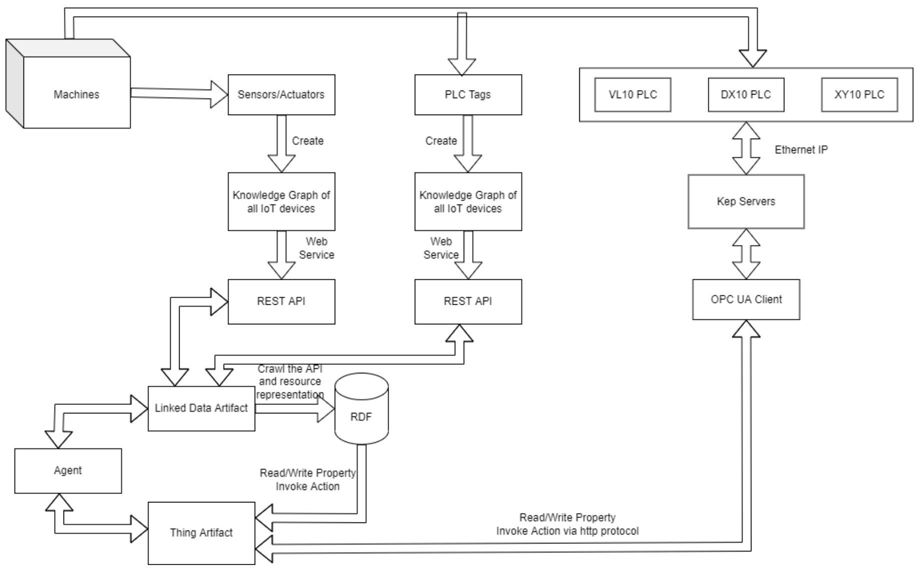

Machines
Sensors/Actuators
PLC Tags
VL10 PLC
DX10 PLC
XY10 PLC
Create
Create
Ethernet IP
Knowledge Graph of all IoT devices
Knowledge Graph of all IoT devices
Kep Servers
Web Service
Web Service
REST API
REST API
OPC UA Client
Crawl the API and resource representation
Linked Data Artifact
RDF
Agent
Read/Write Property Invoke Action
Thing Artifact
Read/Write Property Invoke Action via http protocol

Figure 7: Architecture diagram of the IT'm Factory's Production line

4.1 Conveying Workshop (VL10)

In this scenario, the Conveying Workshop is composed of a storage rack, a Cartesian robot on axes X/Z and a conveyor belt. The conveying workshop fills the storage rack with yogurt cups by an order for picks a cup from the shelves using a Cartesian robot on X/Z axes and it places the cup on a conveyor belt, finally the cup moves towards to the head of conveyor belt on the Filling Workshop.

It is used as follows: 1) an operator loads the storage rack with items, 2) an order to pick an item is placed, 3) the Cartesian robot moves to the input position, grasps the item and moves back to the conveyor belt where the item is released, 4) the item moves on to the next workshop via the conveyor belt.The reference name of the conveying workshop is VL10.

4.2 Filling Workshop (DX10)

The filling workshop is composed of a Cartesian robot on axis X only, a filler (with a magnetic valve) attached to the Cartesian robot and a conveyor belt. In Filling Workshop, when a cup is identified below the filler, the magnetic valve of the filler opens and the robot starts to moving towards X axis on the cup. While the cup is moving yogurt is poured into the cup. When the cup is filled with a certain amount of yogurt, the valve closes and the robot returns to its initial X coordinate. The yogurt filled cup keeps moving towards the end of the conveyor.

To do that, the Cartesian robot is set to have the same speed as the conveyor belt. The process is triggered as soon as a container is detected under the filler. It is used as follows: 1) the filler's valve opens and the Cartesian robot starts moving, 2) when a certain volume has finished pouring, the valve closes and the robot returns to its initial X coordinate at maximum speed while the container keeps moving towards the end of the conveyor. An optical sensor placed at the end of the conveyor belt indicates that a container has reached the end of the workshop and it is ready to be passed on to the next workshop. The reference name of the filling workshop is DX10.

4.3 Potting Workshop (Robotic Arm)

The Potting Workshop main goal is to grasp the yogurt filled cups from the end of the conveyor belt on Filling Workshop and release them to the next workshop. For this, we use a Bosch APAS robot which moves in six different directions and can control grasping/releasing actions via the built-in camera. In IT'm Factory, It is not controlled by mobile assistant robots.

4.4 Packaging Workshop (XY10)

The packaging workshop is composed of a Cartesian robot on axes X/Y, a package buffer (stack) and two conveyor belts. Intially, It pick up the empty package from the package buffer and keeps it on the conveyor belt and then it pickups the cups form the head of the conveyor belt and place them in an empty package. Once the package is filled with four cups, the Cartesian robot places the package on the next conveyor belt.

Its process is as follows: 1) if no package is available between the two conveyors, the robot fetches an empty package from the buffer and places it between the two conveyors, 2) when the two optical sensors on the first conveyor detect a container, the robot moves the pair of containers to free slots in the available package, 3) if there is no more slot, the robot moves the full package to the second conveyor. An optical sensor placed at the head of the second conveyor belt indicates that the full package is ready to be shipped. The reference name of the packaging workshop is XY10.

WAT[7] was developed as an extension to the AI for Industry 4.0 summer school[4]. The demonstrator consists of six autonomous agents which correspond (1) to the cup provider, (2) dairy product provider, (3) vl10 agent for controlling the conveying workshop, (4) dx10 agent for controlling the filling workshop, (5) apas agent for controlling the robotic arm, and (6) the xy10 agent for controlling the packaging workshop. The factory workspace contains seven artifacts, where six of them are modelled as ThingArtifact that the agents can observe and act on, and a LinkedData FuSpider representing the KnowledgeGraph Crawler by using Linked Data-Fu[6] to discover all related data in the knowledge graph using HTTP requests, starting from an entry point based on condition-action-rules and inference rules. The interference rules are predefined per thing to obtain all necessary information.

5 Implementation

We used “Hypermedea” to make Robust Multi-Agent Execution for Industry 4.0. Hypermedea is a programming framework for web agents based on JaCaMo. Hypermedea is a framework to develop autonomous agents[2] that are able to work toward their given goal and interact with their environment and other systems without the immediate help from humans. So, in our case we use Hypermedea[5] to develop autonomous agents for industrial control.

5.1 JaCaMo

JaCaMo[1] is a framework for Multi-Agent Programming that combines three separate technologies. It is a combination of

1. Jason - for programming autonomous agents
2. Cartago - for programming environment artifacts
3. Moise - for programming multi-agent organisations

It covers all levels of needs that are required for the development of knowledgeable multi-agent systems.

5.2 Hypermedea

Hypermedea follows the multi-agent-oriented programming (MAOP) approach, it provides several abstraction layers, from the agents in an environment to their coordination via agent organizations. Hypermedea provides utilities to program agents against a specific environment, that is Web. In MAOP, an agent perceives and acts on its environment via proxy entities called Artifacts. Hypermedea is composed of four artifacts.

1. Linked Data Artifact
2. Thing Artifact
3. Ontology Artifact
4. Planner Artifact

5.2.1 Linked Data Artifact

In web the first thing all web agents will do is Web Navigation. Hypermedea focuses on Linked Data navigation. The Linked Data artifact do an GET operation to dereference a URI and keeps its RDF representation in a knowledge base.

[4]https://ai4industry2021.sciencesconf.org/
[5]https://hypermedea.github.io/tutorial.html

The artifact also keeps track of visited resources and pending dereferencing operations. The Linked Data artifact performs GET operation on a known URI and prints the number of received RDF triples. The Linked Data artifact do GET operations asynchronously.

5.2.2 Thing Artifact

Any agent that targets a specific kind of thing is called a resource agent. Its main goal is to do particular plans to control a manufacturing resource. A resource agent normally has several alternative plans to do in the same manufacturing process, e.g. to autonomously recover from machine failures or to handle internal plans assigned to that. Now, I will explain one simple agent that is programmed on XY10 machine in our case. The main purpose of a packaging machine is to fill empty packaging buffers with product filled bottles. Its high-level process is as follows:

1. bring an empty package buffer to the conveyor.
2. Fill the product filled bottles in buffer.
3. move the filled buffer away from the conveyor

On the XY10 packaging machine, most of the process is executed locally by a Programmable Logic Controller (PLC). The agent may only perform two control actions: it can change the speed of the machine's belt conveyor and it can press an emergency stop that stops PLC execution. The agent may also read several state variables, e.g. number of bottles it packages upto now.

5.3 Thing Description (TD)

A Thing Description describes interfaces of Things. By default, Thing Descriptions are encoded in JSON Format. TD has four main components for describing about anything. Those are, Textual data to describe about the thing itself, Interaction model to describe about how the thing can be used, Schemas for the data exchanged with the thing and Web links to describe about any relation to other things if it has on the web. In Interaction model WoT defines three types of Interaction Affordances.

1. Property Affordance – It is used for sensing and getting the current value in a physical process. Information is provided, To indicate this property is accessible via the HTTP protocal with a GET method.
2. Action Affordance - It is a class in TD. It is used for getting invocation action in a physical process. POST method is used to indicate the status of the action in Action Affordance.
3. Event Affordance – It is used as a push model for asynchronous communication at receiver. It enables a mechanism for asynchronous messages sent by a thing.

In Schemas it has basic security scheme for accessing HTTP protocals. To give security first given name in securityDefinations and then it is activated by giving name in security section. It gives default access requirements for everyone to crawl the knowledge graph.

It also offers to add contextual definitions in some namespace. We can use different namespaces to declare the formal knowledge. Contextual information can also help for specify underlying communication protocols declared in the forms field. Each vocabulary used in the TD Information Model has its own namespace IRI. The Vocabularies are independent from each other.

1. For representing main things like Interaction Model with the Properties, Actions, and Events Interaction Affordances, It has https://www.w3.org/2019/wot/td.
2. For data schema, to the terms defined by JSON Schema, it has https://www.w3.org/2019/wot/json-schema.
3. For security schema, to identifying security mechanisms and requirements for their configuration, it has https://www.w3.org/2019/w
4. For Hypermedia Controls, to encoding the main principles of RESTful communication using Web links and forms, it has https://www.w3.org/2019/wot/hypermedia.

The TD model is a W3C standard for exposing so-called Things on the Web. On the Web of Things, connected devices acting as Web servers are modeled by means of their exposed properties (that clients can read and write), the actions they can perform on the physical world (that clients can invoke) and the events they can detect (that clients can subscribe to). A TD document being an RDF document, its URI can be dereferenced by the Linked Data artifact while crawling.

The XY10 machine's interface is exposed in a standard Thing Description (TD) document. For the XY10 machine, the agent is informed of the following interface:

Operation type	Name
readProperty	Optical Sensor Container1
readProperty	Package Buffer
readProperty, writeProperty	conveyorSpeed
invokeAction	pressEmergencyStop

All possible operations on the machine are called affordances and are described as Web forms (with additional schema information on the expected and returned data). To perform an action, the agent only has to fill in the form with valid input data. If the agent intends to start the conveyor belt, it must submit a single number according to the XY10 machine's TD (corresponding to a speed in m/s).

To keep action as simple as possible, a Thing artifact is created for every TD found in the knowledge base. This artifact exposes readProperty, writeProperty and invokeAction actions that turn an agent's input into a request to the server and that handles the server's response. The Thing artifact also performs input validation and deals with server authentication, if required.

5.3.1 Read Property

Here I represent a sample structure for proportyaffordance for readproporty to an agent on XY10 machine.

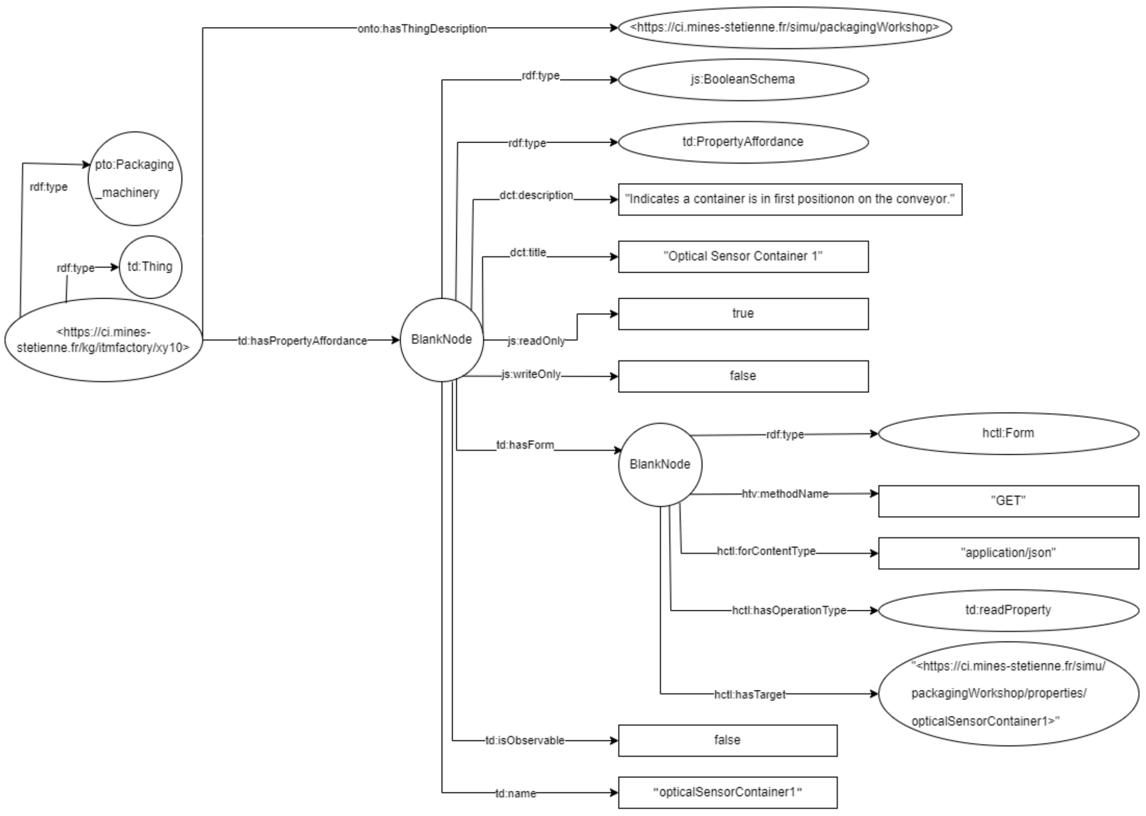


Figure 8: Read Property Architecture for one sample entity

NameSpace	Description
td:Thing	It indicates it is a virtual entity with the composition of one or more Things described by a WoT Thing Description.
pto:Packaging machinery	We use this product ontology to describe the object with a matching entry in the English Wikipedia. Here We use our XY10 machine as a packaging machine to pack product filled bottles.
rdf:type	It is a proporty in RDF used to state that resource is an instance of class.
onto:hasThingDescription	It used to indicate a URL to describe relation between our machine and the thing Description according to the W3C TD standard.
td: hasProportyAffordance	It is used to indicate all Property-based interaction affordance of the Thing.
Js: BooleanSchema	It is used to indicate Boolean value type is assigned in DataSchema instances types.
td:PropertyAffordance	It is one of the type in Interaction Affordances to describe how consumers interact with the thing.
dct:description	It is used to give some human-readable explanation about our resource.
dct:title	It is used to give a human-readable name to our resource.
js:readOnly	It is used to describe that whether our property interaction is read only or not. If ir is readonly we indicate with Boolean value as true other wise we indicate with Boolean value as false.
Js:writeOnly	It is used to describe that whether our property interaction is write only or. SO in readproporty it's defaultly false.
td:hasForm	Here we describe Set of form hypermedia controls to indicate how an operation can be performed.
hctl:Form	It is the object in our Thing Description to describe Thing itself for metadata interactions.
htv: methodName	It is used for indicate the HTTP method name used for the HTTP request.
hctl:forContentType	It is used for describe the content type. For Thing Description by default it takes JSON Format.
hctl: hasOperationType	It indicates the operation described by the form.
hctl: hasTarget	It is used for describe IRI link of the target.
td:isObservable	It is used for indicate either provide a Protocol Binding that supports the observeproperty operation for this Property by intermediaries.
td:name	It is a indexing property to store entity names in @index container when serializing them in a JSON-LD format.

5.3.2 Write Property

It represents the format for write property in Thing Description.

The agent has to read the value from the PLC and act regarding that. For example it has change the speed of the conveyor according to the speed we mention in PLC. The all properties are same as readpproperty expect that for both readOnly and writeOnly it has false and it don't have "GET" object. The TD don't has specific indication for GET.

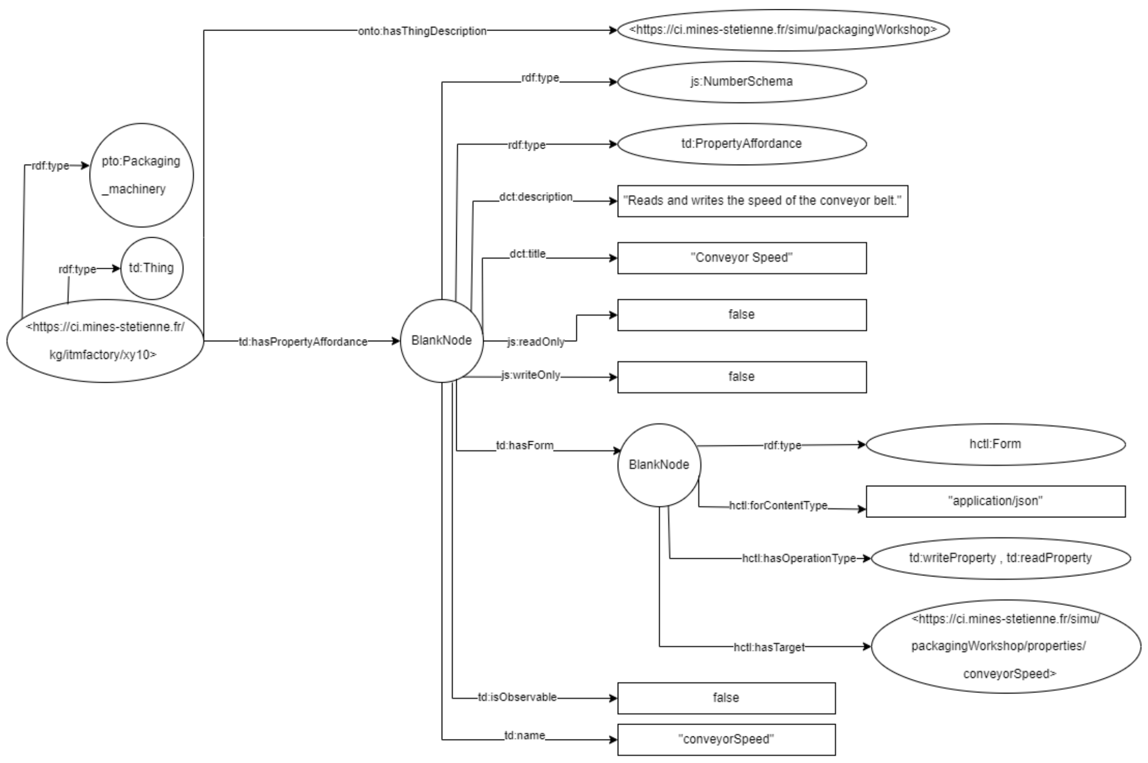


Figure 9: Write Property Architecture for one sample entity

5.3.3 Invoke Action

It represents the InvokeAction in Thing Description of the Thing.

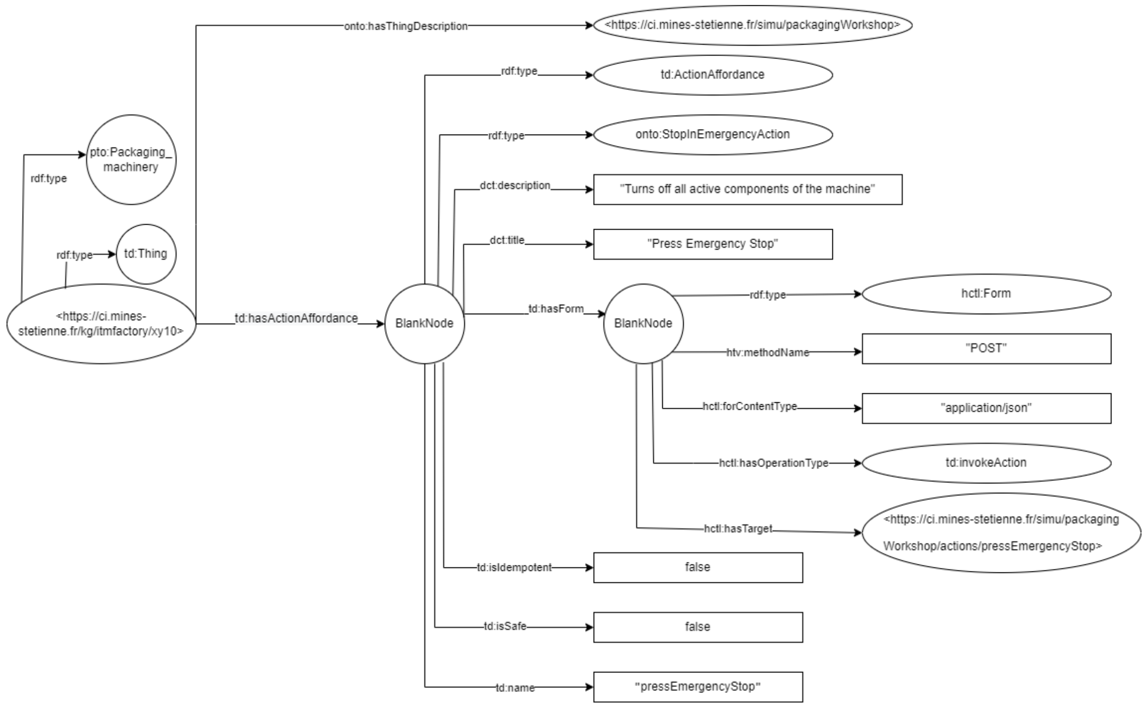


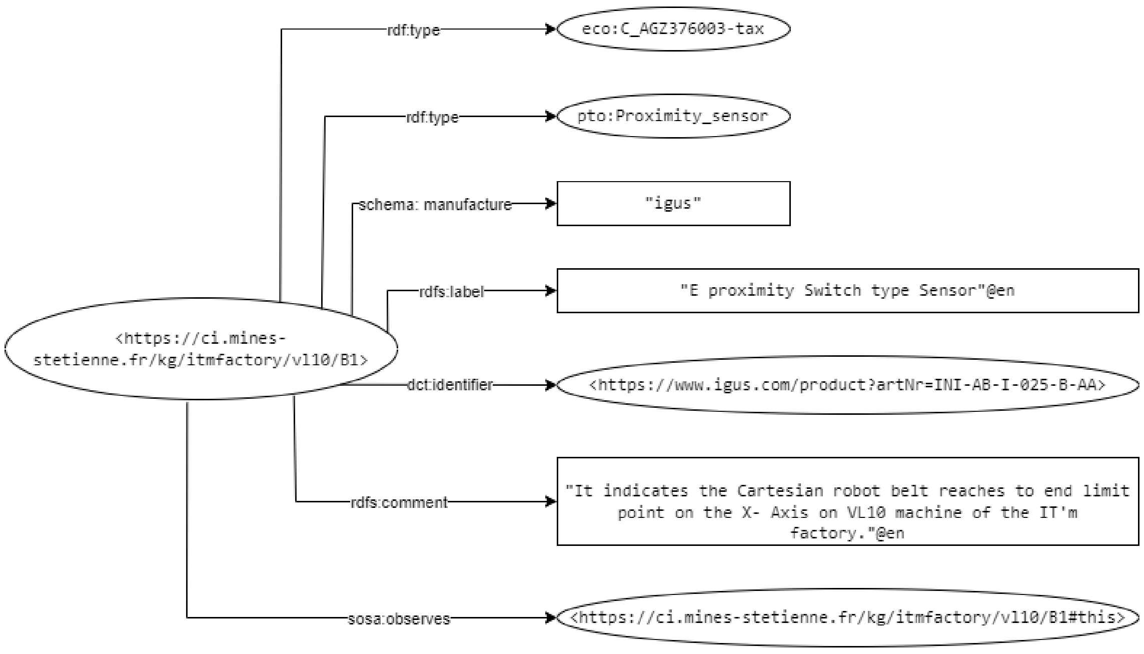
Figure 10: InvokeAction Architecture for one sample entity

It manipulates state or process on the thing. When a form instance is within an actionaffordance instance the value assigned to output must be invokeaction.

5.3.4 Mounted Equipment

To start representing knowledge graph for machines in production line I gathered information about all the mounted equipment on the machines as sensors, motors, actuators too. In knowledge graph we represent equipment names with labels by giving numbers to the equipment like B1, B2, B3 and so on.

Created turtle files for all the labels by giving source as label numbers. In turtle file we given below triples format.



<https://ci.mines-stetienne.fr/kg/itmfactory/vl10/B1>
rdf:type
eco:C_AGZ376003-tax
rdf:type
pto:Proximity_sensor
schema: manufacture
"igus"
rdfs:label
"E proximity Switch type Sensor"@en
dct:identifier
<https://www.igus.com/product?artNr=INI-AB-I-025-B-AA>
rdfs:comment
"It indicates the Cartesian robot belt reaches to end limit point on the X- Axis on VL10 machine of the IT'm factory."@en
sosa:observes
<https://ci.mines-stetienne.fr/kg/itmfactory/vl10/B1#this>

Figure 11: Knowledge graph for one sample label with manufacture information

6 Results

I uploaded all my work on gitlab repository in below mentioned path.

Path : https://gitlab.emse.fr/ext.21m2019/internshipm1/-/tree/master

I created Excel files for all the machines with all necessary coloums to create turtle files and I keep all those files in below mentioned path.

https://gitlab.emse.fr/ext.21m2019/internshipm1/-/tree/master/CSV Files and related PY Files/Excel Files

I am adding screenshots of VL10 Excel file to represent the data in excel files.

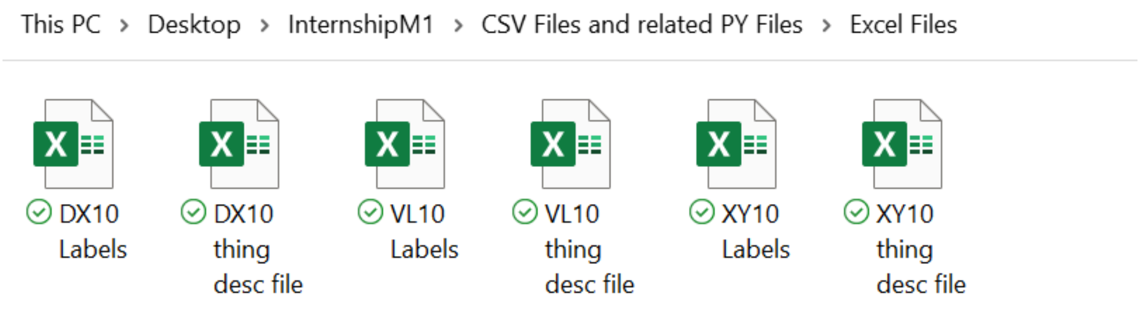
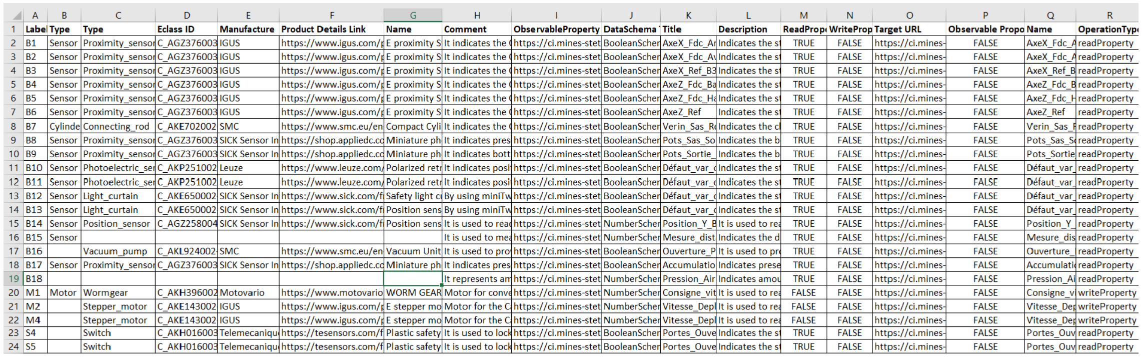
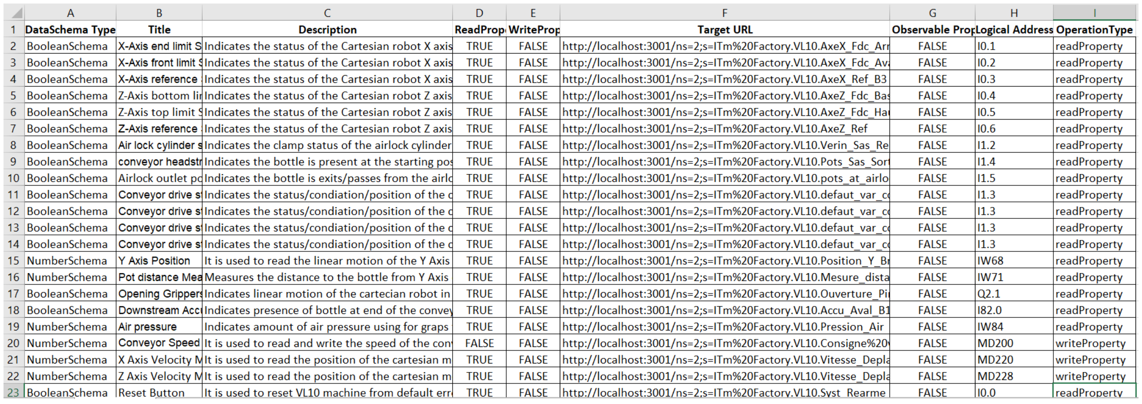


Figure 12: Excel Files



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	Label	Type	Type	Eclass ID	Manufacture	Product Details Link	Name	Comment	ObservableProperty	DataSchema	Title	Description	ReadPrope	WriteProp	Target URL	Observable Propo	Name	OperationType
2	B1	Sensor	Proximity_sensor	C_AGZ376003	IGUS	https://www.igus.com/p	E proximity S	It indicates the	https://ci.mines-stet	BooleanScher	AxeX_Fdc_Ar	Indicates the st	TRUE	FALSE	https://ci.mines-	FALSE	AxeX_Fdc_A	readProperty
3	B2	Sensor	Proximity_sensor	C_AGZ376003	IGUS	https://www.igus.com/p	E proximity S	It indicates the	https://ci.mines-stet	BooleanScher	AxeX_Fdc_Av	Indicates the st	TRUE	FALSE	https://ci.mines-	FALSE	AxeX_Fdc_A	readProperty
4	B3	Sensor	Proximity_sensor	C_AGZ376003	IGUS	https://www.igus.com/p	E proximity S	It indicates the	https://ci.mines-stet	BooleanScher	AxeX_Ref_B3	Indicates the st	TRUE	FALSE	https://ci.mines-	FALSE	AxeX_Ref_B	readProperty
5	B4	Sensor	Proximity_sensor	C_AGZ376003	IGUS	https://www.igus.com/p	E proximity S	It indicates the	https://ci.mines-stet	BooleanScher	AxeZ_Fdc_Ba	Indicates the st	TRUE	FALSE	https://ci.mines-	FALSE	AxeZ_Fdc_B	readProperty
6	B5	Sensor	Proximity_sensor	C_AGZ376003	IGUS	https://www.igus.com/p	E proximity S	It indicates the	https://ci.mines-stet	BooleanScher	AxeZ_Fdc_Ha	Indicates the st	TRUE	FALSE	https://ci.mines-	FALSE	AxeZ_Fdc_H	readProperty
7	B6	Sensor	Proximity_sensor	C_AGZ376003	IGUS	https://www.igus.com/p	E proximity S	It indicates the	https://ci.mines-stet	BooleanScher	AxeZ_Ref	Indicates the st	TRUE	FALSE	https://ci.mines-	FALSE	AxeZ_Ref	readProperty
8	B7	Cylinde	Connecting_rod	C_AKE702002	SMC	https://www.smc.eu/en	Compact Cyli	It indicates the	https://ci.mines-stet	BooleanScher	Verin_Sas_Re	Indicates the cl	TRUE	FALSE	https://ci.mines-	FALSE	Verin_Sas_R	readProperty
9	B8	Sensor	Proximity_sensor	C_AGZ376003	SICK Sensor In	https://shop.appliedc.co	Miniature ph	It indicates pres	https://ci.mines-stet	BooleanScher	Pots_Sas_So	Indicates the b	TRUE	FALSE	https://ci.mines-	FALSE	Pots_Sas_S	readProperty
10	B9	Sensor	Proximity_sensor	C_AGZ376003	SICK Sensor In	https://shop.appliedc.co	Miniature ph	It indicates bott	https://ci.mines-stet	BooleanScher	Pots_Sortie_	Indicates the b	TRUE	FALSE	https://ci.mines-	FALSE	Pots_Sortie	readProperty
11	B10	Sensor	Photoelectric_ser	C_AKP251002	Leuze	https://www.leuze.com/	Polarized retr	It indicates posi	https://ci.mines-stet	BooleanScher	Défaut_var_c	Indicates the st	TRUE	FALSE	https://ci.mines-	FALSE	Défaut_var_	readProperty
12	B11	Sensor	Photoelectric_ser	C_AKP251002	Leuze	https://www.leuze.com/	Polarized retr	It indicates posi	https://ci.mines-stet	BooleanScher	Défaut_var_c	Indicates the st	TRUE	FALSE	https://ci.mines-	FALSE	Défaut_var_	readProperty
13	B12	Sensor	Light_curtain	C_AKE650002	SICK Sensor In	https://www.sick.com/f	Safety light c	By using miniTw	https://ci.mines-stet	BooleanScher	Défaut_var_c	Indicates the st	TRUE	FALSE	https://ci.mines-	FALSE	Défaut_var_	readProperty
14	B13	Sensor	Light_curtain	C_AKE650002	SICK Sensor In	https://www.sick.com/f	Position sens	By using miniTw	https://ci.mines-stet	BooleanScher	Défaut_var_c	Indicates the st	TRUE	FALSE	https://ci.mines-	FALSE	Défaut_var_	readProperty
15	B14	Sensor	Position_sensor	C_AGZ258004	SICK Sensor In	https://www.sick.com/f	Position sens	It is used to read	https://ci.mines-stet	NumberScher	Position_Y_B	It is used to rea	TRUE	FALSE	https://ci.mines-	FALSE	Position_Y_	readProperty
16	B15	Sensor						It is used to mea	https://ci.mines-stet	NumberScher	Mesure_dist	Indicates the d	TRUE	FALSE	https://ci.mines-	FALSE	Mesure_dis	readProperty
17	B16		Vacuum_pump	C_AKL924002	SMC	https://www.smc.eu/en	Vacuum Unit	It is used to pro	https://ci.mines-stet	BooleanScher	Ouverture_P	It is used to pr	TRUE	FALSE	https://ci.mines-	FALSE	Ouverture_	readProperty
18	B17	Sensor	Proximity_sensor	C_AGZ376003	SICK Sensor In	https://shop.appliedc.co	Miniature ph	It indicates pres	https://ci.mines-stet	BooleanScher	Accumulatio	Indicates prese	TRUE	FALSE	https://ci.mines-	FALSE	Accumulati	readProperty
19	B18							It represents am	https://ci.mines-stet	NumberScher	Pression_Air	Indicates amou	TRUE	FALSE	https://ci.mines-	FALSE	Pression_Ai	readProperty
20	M1	Motor	Wormgear	C_AKH396002	Motovario	https://www.motovario	WORM GEAR	Motor for conve	https://ci.mines-stet	NumberScher	Consigne_vit	It is used to rea	FALSE	FALSE	https://ci.mines-	FALSE	Consigne_vi	writeProperty
21	M2		Stepper_motor	C_AKE143002	IGUS	https://www.igus.com/p	E stepper mo	Motor for the C	https://ci.mines-stet	NumberScher	Vitesse_Depl	It is used to rea	FALSE	FALSE	https://ci.mines-	FALSE	Vitesse_Dep	writeProperty
22	M4		Stepper_motor	C_AKE143002	IGUS	https://www.igus.com/p	E stepper mo	Motor for the C	https://ci.mines-stet	NumberScher	Vitesse_Depl	It is used to rea	FALSE	FALSE	https://ci.mines-	FALSE	Vitesse_Dep	writeProperty
23	S4		Switch	C_AKH016003	Telemecaniqu	https://tesensors.com/f	Plastic safety	It is used to lock	https://ci.mines-stet	BooleanScher	Portes_Ouve	Indicates the st	TRUE	FALSE	https://ci.mines-	FALSE	Portes_Ouv	readProperty
24	S5		Switch	C_AKH016003	Telemecaniqu	https://tesensors.com/f	Plastic safety	It is used to lock	https://ci.mines-stet	BooleanScher	Portes_Ouve	Indicates the st	TRUE	FALSE	https://ci.mines-	FALSE	Portes_Ouv	readProperty

Figure 13: Excel File content for labels description



	A	B	C	D	E	F	G	H	I
1	DataSchema Type	Title	Description	ReadPrope	WriteProp	Target URL	Observable Prop	Logical Address	OperationType
2	BooleanSchema	X-Axis end limit S	Indicates the status of the Cartesian robot X axis	TRUE	FALSE	http://localhost:3001/ns=2;s=ITm%20Factory.VL10.AxeX_Fdc_Arr	FALSE	I0.1	readProperty
3	BooleanSchema	X-Axis front limit S	Indicates the status of the Cartesian robot X axis	TRUE	FALSE	http://localhost:3001/ns=2;s=ITm%20Factory.VL10.AxeX_Fdc_Ava	FALSE	I0.2	readProperty
4	BooleanSchema	X-Axis reference	Indicates the status of the Cartesian robot X axis	TRUE	FALSE	http://localhost:3001/ns=2;s=ITm%20Factory.VL10.AxeX_Ref_B3	FALSE	I0.3	readProperty
5	BooleanSchema	Z-Axis bottom lir	Indicates the status of the Cartesian robot Z axis	TRUE	FALSE	http://localhost:3001/ns=2;s=ITm%20Factory.VL10.AxeZ_Fdc_Bas	FALSE	I0.4	readProperty
6	BooleanSchema	Z-Axis top limit S	Indicates the status of the Cartesian robot Z axis	TRUE	FALSE	http://localhost:3001/ns=2;s=ITm%20Factory.VL10.AxeZ_Fdc_Hau	FALSE	I0.5	readProperty
7	BooleanSchema	Z-Axis reference	Indicates the status of the Cartesian robot Z axis	TRUE	FALSE	http://localhost:3001/ns=2;s=ITm%20Factory.VL10.AxeZ_Ref	FALSE	I0.6	readProperty
8	BooleanSchema	Air lock cylinder s	Indicates the clamp status of the airlock cylinder	TRUE	FALSE	http://localhost:3001/ns=2;s=ITm%20Factory.VL10.Verin_Sas_Re	FALSE	I1.2	readProperty
9	BooleanSchema	conveyor headstr	Indicates the bottle is present at the starting pos	TRUE	FALSE	http://localhost:3001/ns=2;s=ITm%20Factory.VL10.Pots_Sas_Sort	FALSE	I1.4	readProperty
10	BooleanSchema	Airlock outlet pc	Indicates the bottle is exits/passes from the airlc	TRUE	FALSE	http://localhost:3001/ns=2;s=ITm%20Factory.VL10.pots_at_airlo	FALSE	I1.5	readProperty
11	BooleanSchema	Conveyor drive s	Indicates the status/condiation/position of the c	TRUE	FALSE	http://localhost:3001/ns=2;s=ITm%20Factory.VL10.defaut_var_co	FALSE	I1.3	readProperty
12	BooleanSchema	Conveyor drive s	Indicates the status/condiation/position of the c	TRUE	FALSE	http://localhost:3001/ns=2;s=ITm%20Factory.VL10.defaut_var_co	FALSE	I1.3	readProperty
13	BooleanSchema	Conveyor drive s	Indicates the status/condiation/position of the c	TRUE	FALSE	http://localhost:3001/ns=2;s=ITm%20Factory.VL10.defaut_var_co	FALSE	I1.3	readProperty
14	BooleanSchema	Conveyor drive s	Indicates the status/condiation/position of the c	TRUE	FALSE	http://localhost:3001/ns=2;s=ITm%20Factory.VL10.defaut_var_co	FALSE	I1.3	readProperty
15	NumberSchema	Y Axis Position	It is used to read the linear motion of the Y Axis	TRUE	FALSE	http://localhost:3001/ns=2;s=ITm%20Factory.VL10.Position_Y_Br	FALSE	IW68	readProperty
16	NumberSchema	Pot distance Mea	Measures the distance to the bottle from Y Axis	TRUE	FALSE	http://localhost:3001/ns=2;s=ITm%20Factory.VL10.Mesure_dista	FALSE	IW71	readProperty
17	BooleanSchema	Opening Grippers	Indicates linear motion of the cartecian robot in	TRUE	FALSE	http://localhost:3001/ns=2;s=ITm%20Factory.VL10.Ouverture_Pi	FALSE	Q2.1	readProperty
18	BooleanSchema	Downstream Acc	Indicates presence of bottle at end of the convey	TRUE	FALSE	http://localhost:3001/ns=2;s=ITm%20Factory.VL10.Accu_Aval_B1	FALSE	I82.0	readProperty
19	NumberSchema	Air pressure	Indicates amount of air pressure using for graps	TRUE	FALSE	http://localhost:3001/ns=2;s=ITm%20Factory.VL10.Pression_Air	FALSE	IW84	readProperty
20	NumberSchema	Conveyor Speed	It is used to read and write the speed of the con	FALSE	FALSE	http://localhost:3001/ns=2;s=ITm%20Factory.VL10.Consigne%20	FALSE	MD200	writeProperty
21	NumberSchema	X Axis Velocity M	It is used to read the position of the cartesian m	TRUE	FALSE	http://localhost:3001/ns=2;s=ITm%20Factory.VL10.Vitesse_Depla	FALSE	MD220	writeProperty
22	NumberSchema	Z Axis Velocity M	It is used to read the position of the cartesian m	TRUE	FALSE	http://localhost:3001/ns=2;s=ITm%20Factory.VL10.Vitesse_Depla	FALSE	MD228	writeProperty
23	BooleanSchema	Reset Button	It is used to reset VL10 machine from default err	TRUE	FALSE	http://localhost:3001/ns=2;s=ITm%20Factory.VL10.Syst_Rearme	FALSE	I0.0	readProperty

Figure 14: Excel File content for thing description

Converted all those Excel files into CSV files for use those in my Python program and uploaded those files in below path. https://gitlab.emse.fr/ext.21m2019/internshipm1/-/tree/master/CSV Files and related PY Files

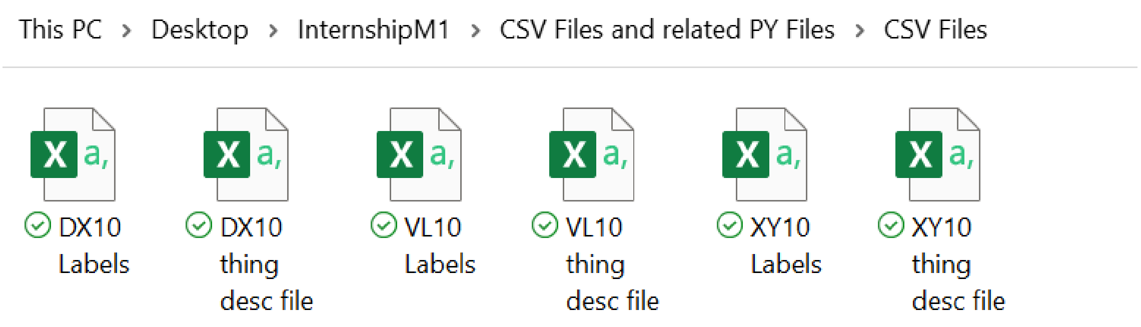


Figure 15: CSV Files

From those CSV Files, Created turtle files for all the mounted equipment on all the machines in below mentioned folders.

For VL10 Machine - https://gitlab.emse.fr/ext.21m2019/internshipm1/-/tree/master/CSV Files and related PY Files/VL10 Labels turtle files

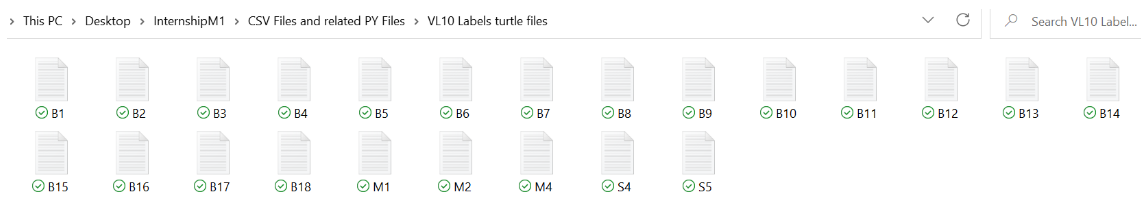

This PC › Desktop › InternshipM1 › CSV Files and related PY Files › VL10 Labels turtle files
Search VL10 Label...
B1
B2
B3
B4
B5
B6
B7
B8
B9
B10
B11
B12
B13
B14
B15
B16
B17
B18
M1
M2
M4
S4
S5

Figure 16: vl10 turtle Files

For DX10 Machine - https://gitlab.emse.fr/ext.21m2019/internshipm1/-/tree/master/CSV Files and related PY Files/DX10 Labels turtle files

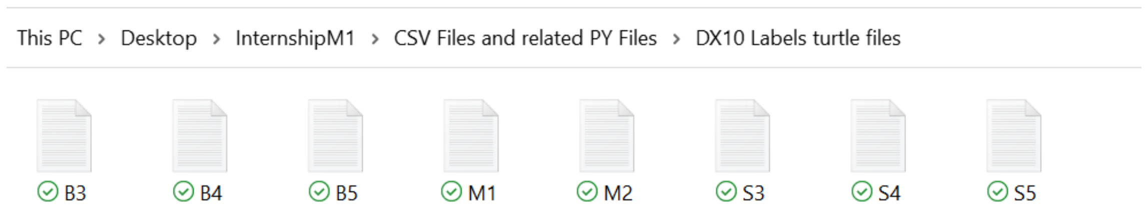

This PC › Desktop › InternshipM1 › CSV Files and related PY Files › DX10 Labels turtle files
B3
B4
B5
M1
M2
S3
S4
S5

Figure 17: dx10 turtle Files

For XY10 Machine - https://gitlab.emse.fr/ext.21m2019/internshipm1/-/tree/master/CSV Files and related PY Files/XY10 Labels turtle files

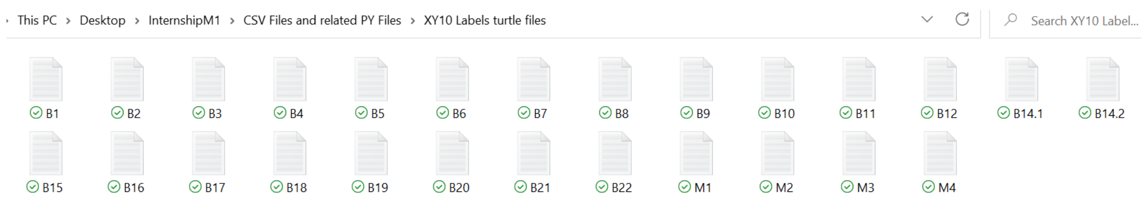
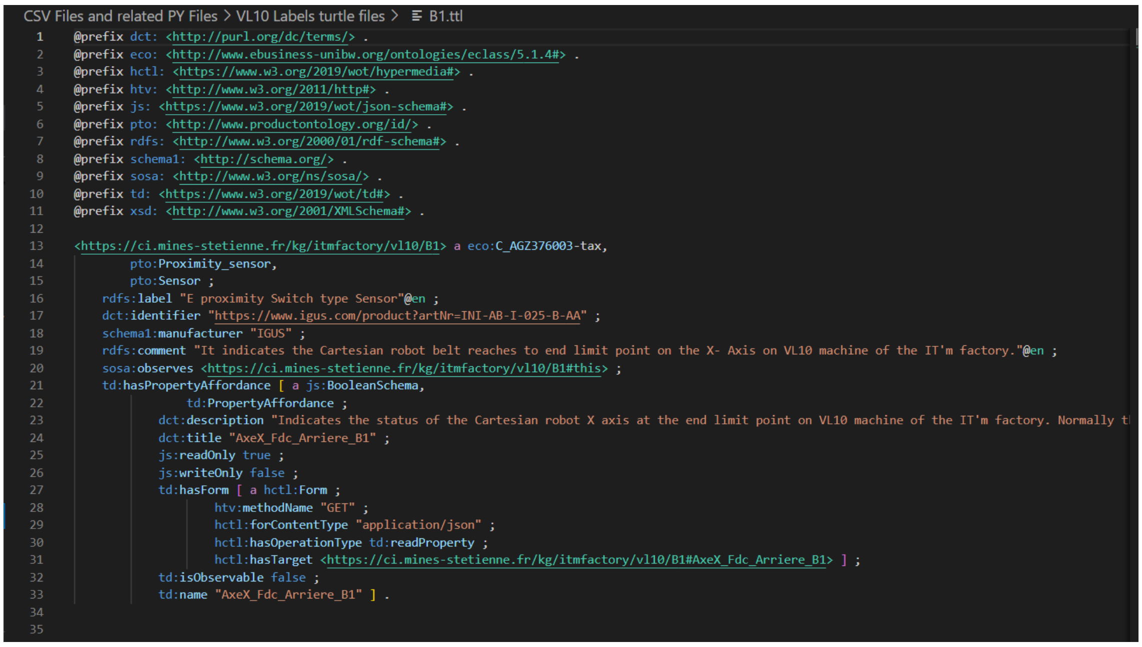

This PC › Desktop › InternshipM1 › CSV Files and related PY Files › XY10 Labels turtle files
Search XY10 Label...
B1
B2
B3
B4
B5
B6
B7
B8
B9
B10
B11
B12
B14.1
B14.2
B15
B16
B17
B18
B19
B20
B21
B22
M1
M2
M3
M4

Figure 18: xy10 turtle Files

I am adding sample screenshot to represent the data in one turtle file.



CSV Files and related PY Files > VL10 Labels turtle files > B1.ttl

```
@prefix dct: <http://purl.org/dc/terms/> .
@prefix eco: <http://www.ebusiness-unibw.org/ontologies/eclass/5.1.4#> .
@prefix hctl: <https://www.w3.org/2019/wot/hypermedia#> .
@prefix htv: <http://www.w3.org/2011/http#> .
@prefix js: <https://www.w3.org/2019/wot/json-schema#> .
@prefix pto: <http://www.productontology.org/id/> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix schema1: <http://schema.org/> .
@prefix sosa: <http://www.w3.org/ns/sosa/> .
@prefix td: <https://www.w3.org/2019/wot/td#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<https://ci.mines-stetienne.fr/kg/itmfactory/vl10/B1> a eco:C_AGZ376003-tax,
        pto:Proximity_sensor,
        pto:Sensor ;
    rdfs:label "E proximity Switch type Sensor"@en ;
    dct:identifier "https://www.igus.com/product?artNr=INI-AB-I-025-B-AA" ;
    schema1:manufacturer "IGUS" ;
    rdfs:comment "It indicates the Cartesian robot belt reaches to end limit point on the X- Axis on VL10 machine of the IT'm factory."@en ;
    sosa:observes <https://ci.mines-stetienne.fr/kg/itmfactory/vl10/B1#this> ;
    td:hasPropertyAffordance [ a js:BooleanSchema,
                td:PropertyAffordance ;
            dct:description "Indicates the status of the Cartesian robot X axis at the end limit point on VL10 machine of the IT'm factory. Normally t
            dct:title "AxeX_Fdc_Arriere_B1" ;
            js:readOnly true ;
            js:writeOnly false ;
            td:hasForm [ a hctl:Form ;
                    htv:methodName "GET" ;
                    hctl:forContentType "application/json" ;
                    hctl:hasOperationType td:readProperty ;
                    hctl:hasTarget <https://ci.mines-stetienne.fr/kg/itmfactory/vl10/B1#AxeX_Fdc_Arriere_B1> ] ;
            td:isObservable false ;
            td:name "AxeX_Fdc_Arriere_B1" ] .
```

Figure 19: turtle File for label

Created Thing Description for all the three machines in turtle format in below folder.

Thing Description turtle files path - https://gitlab.emse.fr/ext.21m2019/internshipm1/-/tree/master/CSV Files and related PY Files/thing desc files

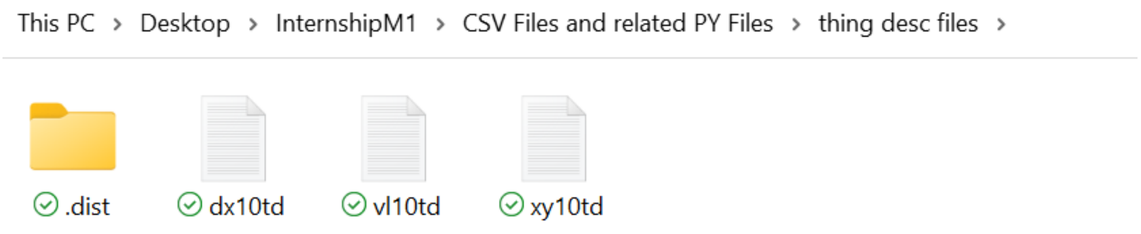
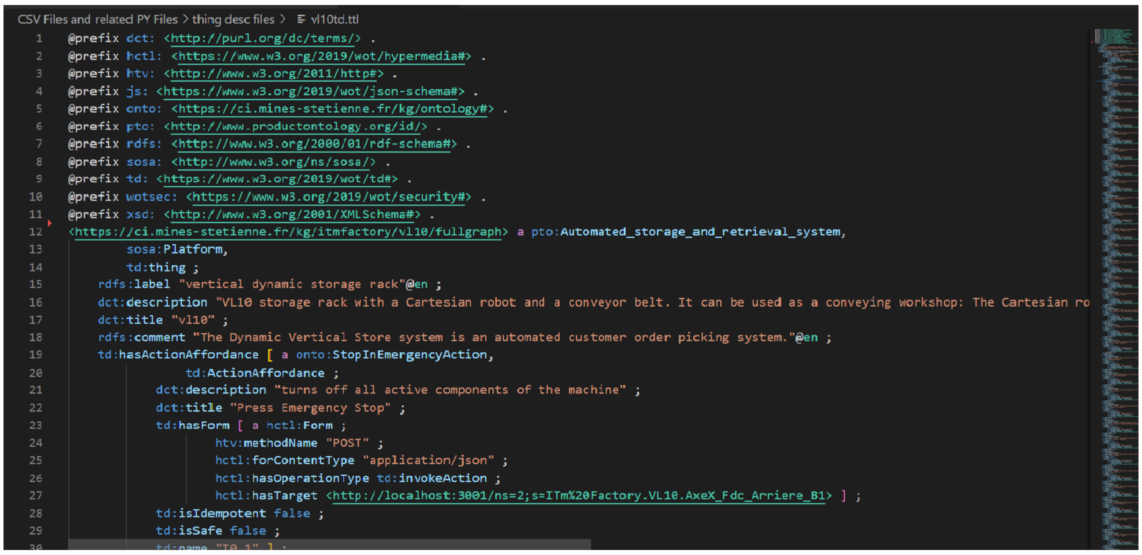

This PC > Desktop > InternshipM1 > CSV Files and related PY Files > thing desc files >
.dist
dx10td
vl10td
xy10td

Figure 20: thing description turtle Files

I am adding screenshot to represent the data format in thing description turtle file.


CSV Files and related PY Files > thing desc files > vl10td.ttl

```
@prefix dct: <http://purl.org/dc/terms/> .
@prefix hctl: <https://www.w3.org/2019/wot/hypermedia#> .
@prefix htv: <http://www.w3.org/2011/http#> .
@prefix js: <https://www.w3.org/2019/wot/json-schema#> .
@prefix onto: <https://ci.mines-stetienne.fr/kg/ontology#> .
@prefix pto: <http://www.productontology.org/id/> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix sosa: <http://www.w3.org/ns/sosa/> .
@prefix td: <https://www.w3.org/2019/wot/td#> .
@prefix wotsec: <https://www.w3.org/2019/wot/security#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
<https://ci.mines-stetienne.fr/kg/itmfactory/vl10/fullgraph> a pto:Automated_storage_and_retrieval_system,
        sosa:Platform,
        td:thing ;
    rdfs:label "vertical dynamic storage rack"@en ;
    dct:description "VL10 storage rack with a Cartesian robot and a conveyor belt. It can be used as a conveying workshop: The Cartesian ro
    dct:title "vl10" ;
    rdfs:comment "The Dynamic Vertical Store system is an automated customer order picking system."@en ;
    td:hasActionAffordance [ a onto:StopInEmergencyAction,
                td:ActionAffordance ;
            dct:description "turns off all active components of the machine" ;
            dct:title "Press Emergency Stop" ;
            td:hasForm [ a hctl:Form ;
                    htv:methodName "POST" ;
                    hctl:forContentType "application/json" ;
                    hctl:hasOperationType td:invokeAction ;
                    hctl:hasTarget <http://localhost:3001/ns=2;s=ITm%20Factory.VL10.AxeX_Fdc_Arriere_B1> ] ;
            td:isIdempotent false ;
            td:isSafe false ;
            td:name "I0_1" ] ;
```

Figure 21: thing output file

In the below table, I am representing some entities added in thing description Knowledge Graph for SIRAM. The Knowledge Graph has more entities and to represent all those in table it takes more pages. I created web URL's to run individual Knowledge Graphs for individual machines.

1. For VL10 Machine -http://127.0.0.1:5000/vl10
2. For DX10 Machine -http://127.0.0.1:5000/dx10
3. For XY10 Machine -http://127.0.0.1:5000/xy10

Address	Title	Tag Description	DataSchematype	Target URL
I0.1	X-Axis end limit Status	Indicates the status of the Cartesian robot X axis at the end limit point on VL10 machine of the IT'm factory. Normally the switch is in close and the status is true, if the Cartesian robot belt reaches to end point then the switch is opened and the status is false.	BooleanSchema	http://localhost:3001/ns=2;s=ITm%20Factory.VL10.AxeX_Fdc_Arriere_B1
I0.2	X-Axis front limit Status	Indicates the status of the Cartesian robot X axis at the front limit point on VL10 machine of the IT'm factory. Normally the switch is in close and the status is true, if the Cartesian robot belt reaches to front limit point then the switch is opened and the status is false.	BooleanSchema	http://localhost:3001/ns=2;s=ITm%20Factory.VL10.AxeX_Fdc_Avant_B2
I0.3	X-Axis reference Status	Indicates the status of the Cartesian robot X axis reference on VL10 machine of the IT'm factory. Normally the switch is in open and the status is false, if belt reaches to intial/ideal position on the X-Axis then the switch is closed and the status is true.	BooleanSchema	http://localhost:3001/ns=2;s=ITm%20Factory.VL10.AxeX_Ref_B3
I0.4	Z-Axis bottom limit Status	Indicates the status of the Cartesian robot Z axis at the bottom limit point on VL10 machine of the IT'm factory. Normally the switch is in close and the status is true, if belt reaches to bottom limit point then the switch is opened and the status is false.	BooleanSchema	http://localhost:3001/ns=2;s=ITm%20Factory.VL10.AxeZ_Fdc_Bas_B4
I0.5	Z-Axis top limit Status	Indicates the status of the Cartesian robot Z axis at the top limit point on VL10 machine of the IT'm factory. Normally the switch is in close and the status is true, if belt reaches to top limit point then the switch is opened and the status is false.	BooleanSchema	http://localhost:3001/ns=2;s=ITm%20Factory.VL10.AxeZ_Fdc_Haut_B5
I0.6	Z-Axis reference Status	Indicates the status of the Cartesian robot Z axis reference on VL10 machine of the IT'm factory. Normally the switch is in open and the status is false, if belt reaches to intial/ideal position on the Z-Axis then the switch is closed and the status is true.	BooleanSchema	http://localhost:3001/ns=2;s=ITm%20Factory.VL10.AxeZ_Ref
I1.2	Air lock cylinder status	Indicates the clamp status of the airlock cylinder on the conveyor belt on VL10 machine of the IT'm factory. Normally the airlock is in off and the clamp status is true, if the bottle reaches to airlock place and at that time one more bottle is present at the downstream of the conveyor then air lock is on and the clamp status is false.	BooleanSchema	http://localhost:3001/ns=2;s=ITm%20Factory.VL10.Verin_Sas_Rentree
I1.4	conveyor headstream accumulation status	Indicates the bottle is present at the starting position of the conveyor belt on the VL10 Machine of the IT'm factory.(upstream accumulation).	BooleanSchema	http://localhost:3001/ns=2;s=ITm%20Factory.VL10.Pots_Sas_Sortie
I1.5	Airlock outlet pots	Indicates the bottle is exits/passes from the airlock position of the conveyor belt on the VL10 Machine of the IT'm factory.	BooleanSchema	http://localhost:3001/ns=2;s=ITm%20Factory.VL10.pots_at_airlock
I1.3	Conveyor drive status	Indicates the status/condiation/position of the conveyor belt drive on the VL10 Machine of the IT'm factory.	BooleanSchema	http://localhost:3001/ns=2;s=ITm%20Factory.VL10.defaut_var_convoyeur
IW68	Y Axis Position	It is used to read the linear motion of the Y Axis on the VL10 Machine of the IT'm factory.	NumberSchema	http://localhost:3001/ns=2;s=ITm%20Factory.VL10.Position_Y_Brut

Address	Title	Tag Description	DataSchematype	Target URL
Q2.1	Opening Grippers	Indicates linear motion of the cartecian robot in Y-Axis direction to graps the bottle on the VL10 Machine of the IT'm factory.	BooleanSchema	http://localhost:3001/ns=2;s=ITm%20Factory.VL10.Ouverture_Pinces
I82.0	Downstream Accumulation	Indicates presence of bottle at end of the conveyor belt on the VL10 of the IT'm factory. (downstream accumulation)	BooleanSchema	http://localhost:3001/ns=2;s=ITm%20Factory.VL10.Accu_Aval_B17
IW84	Air pressure	Indicates amount of air pressure using for graps the bottle on the VL10 of the IT'm factory.	NumberSchema	http://localhost:3001/ns=2;s=ITm%20Factory.VL10.Pression_Air
MD200	Conveyor Speed	It is used to read and write the speed of the conveyor belt on the VL10 Machine of the IT'm factory.	NumberSchema	http://localhost:3001/ns=2;s=ITm%20Factory.VL10.Consigne%20vitesse%20convoyeur
IW71	Pot distance Measurement	Measures the distance to the bottle from Y Axis on the VL10 Machine of the IT'm factory.	NumberSchema	http://localhost:3001/ns=2;s=ITm%20Factory.VL10.Mesure_distance_Pot
I0.0	Reset Button	It is used to reset VL10 machine from default errors of the IT'm factory.	BooleanSchema	http://localhost:3001/ns=2;s=ITm%20Factory.VL10.Syst_Rearme_KA1
MD264	Y Axis Current Position	It reads the current position of the Cartesian motor on Y Axis on VL10 machine of the IT'm factory.	NumberSchema	http://localhost:3001/ns=2;s=ITm%20Factory.VL10.Position_Actuelle_Y
MW276	X Axis Index Column	It reads the current value of the X Axis Index Column number on VL10 machine of the IT'm factory.	NumberSchema	http://localhost:3001/ns=2;s=ITm%20Factory.VL10.Index_Colonne_X
Q2.3	Y Axis Output	Detects the bottle in storage rack from the Y Axis to pick on the VL10 Machine of the IT'm factory.	BooleanSchema	http://localhost:3001/ns=2;s=ITm%20Factory.VL10.Sortir_AxeY
M55.5	Taken Pot2	Indicates the Cartesian motor took the pot in the second position on tray on VL10 machine of the IT'm factory.	BooleanSchema	http://localhost:3001/ns=2;s=ITm%20Factory.VL10.Pot2_Pris
M80.3	Visualization of the console Initialisation button	Indicates Initialisation button is able to visualize on VL10 machine of the IT'm factory.	BooleanSchema	http://localhost:3001/ns=2;s=ITm%20Factory.VL10.Visu%20bouton%20Initialisation_pupitre

7 UML Diagrams

7.1 State Diagrams

State diagram is a type of UML diagram used to represent the behavior of systems. Below I am representing some state diagrams for machines. I used PlantUML[6] to draw these diagrams.

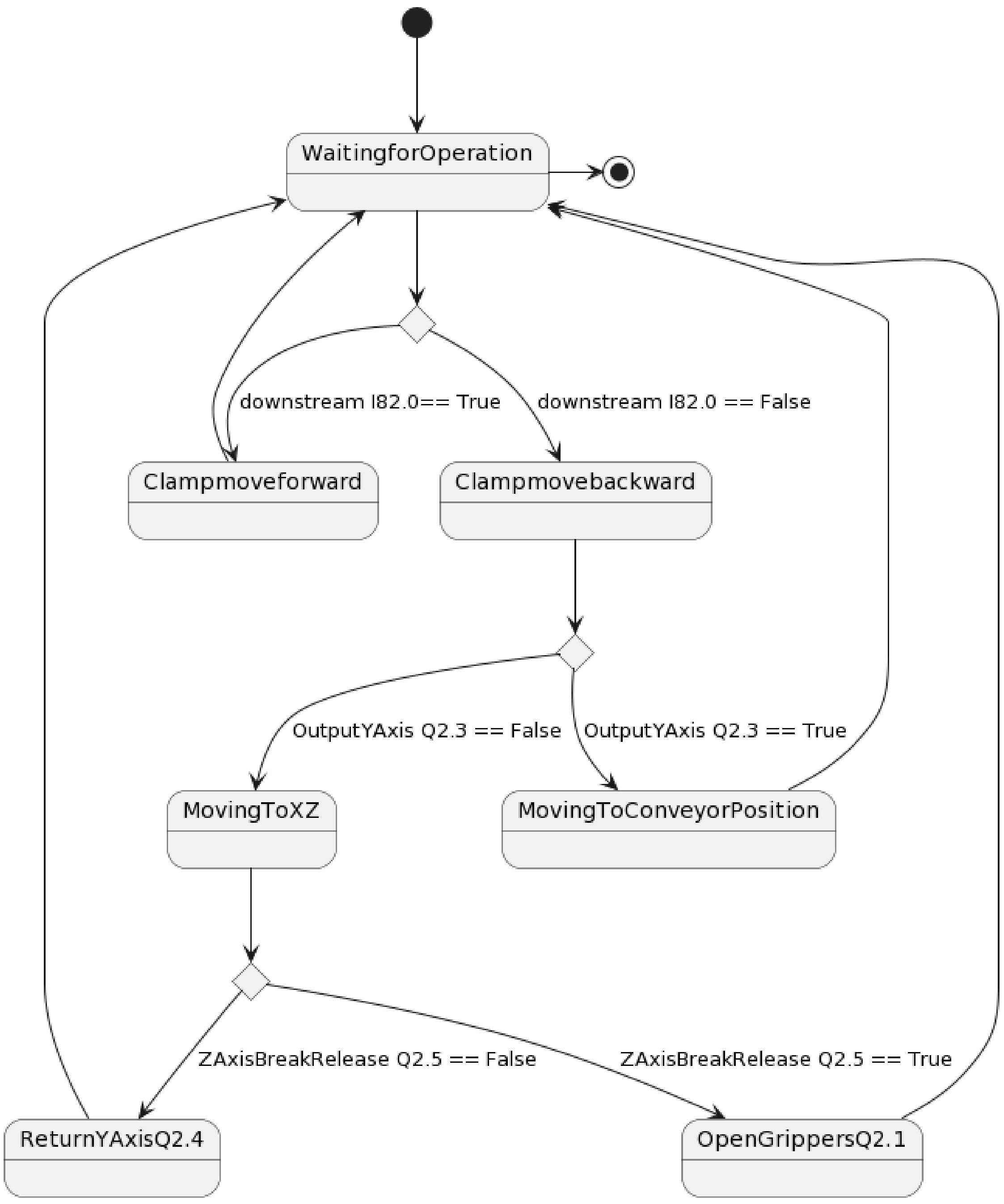


Figure 22: State diagram for VL10 machine

[6]https://plantuml.com/

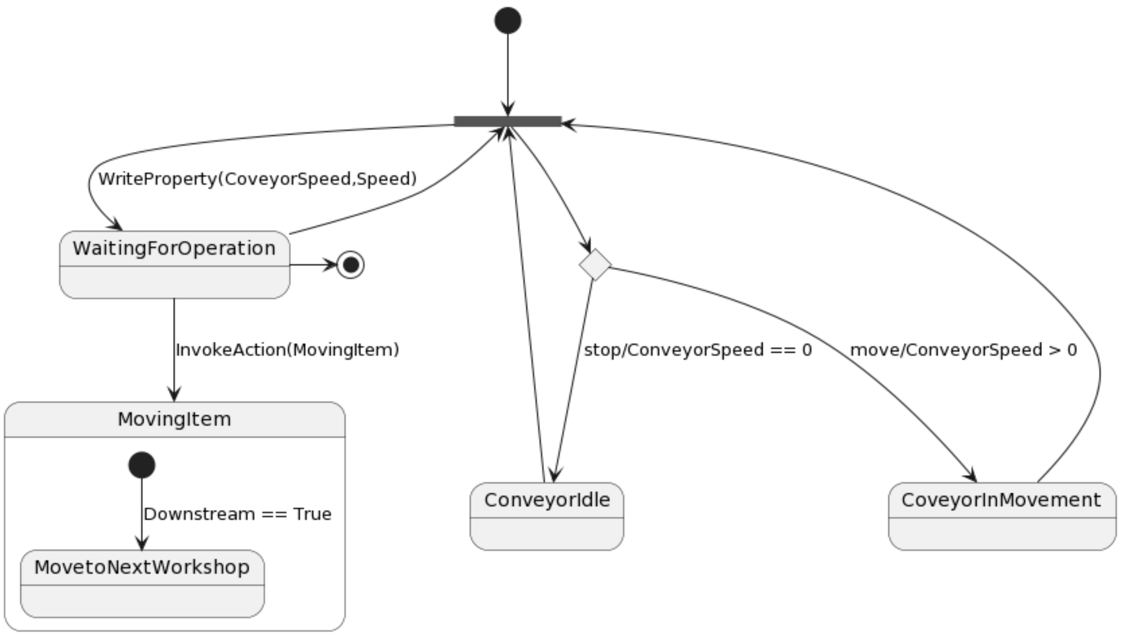


Figure 23: State diagram for VL10 machine

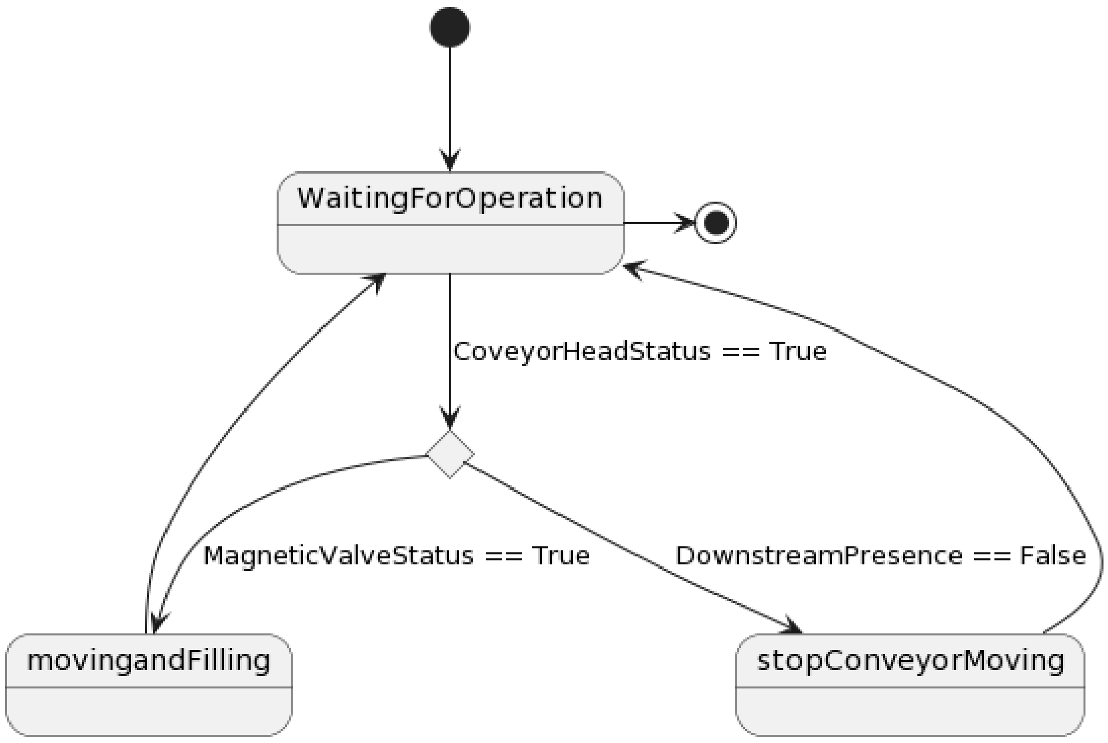

WaitingForOperation
CoveyorHeadStatus == True
MagneticValveStatus == True
DownstreamPresence == False
movingandFilling
stopConveyorMoving

Figure 24: State diagram for DX10 machine

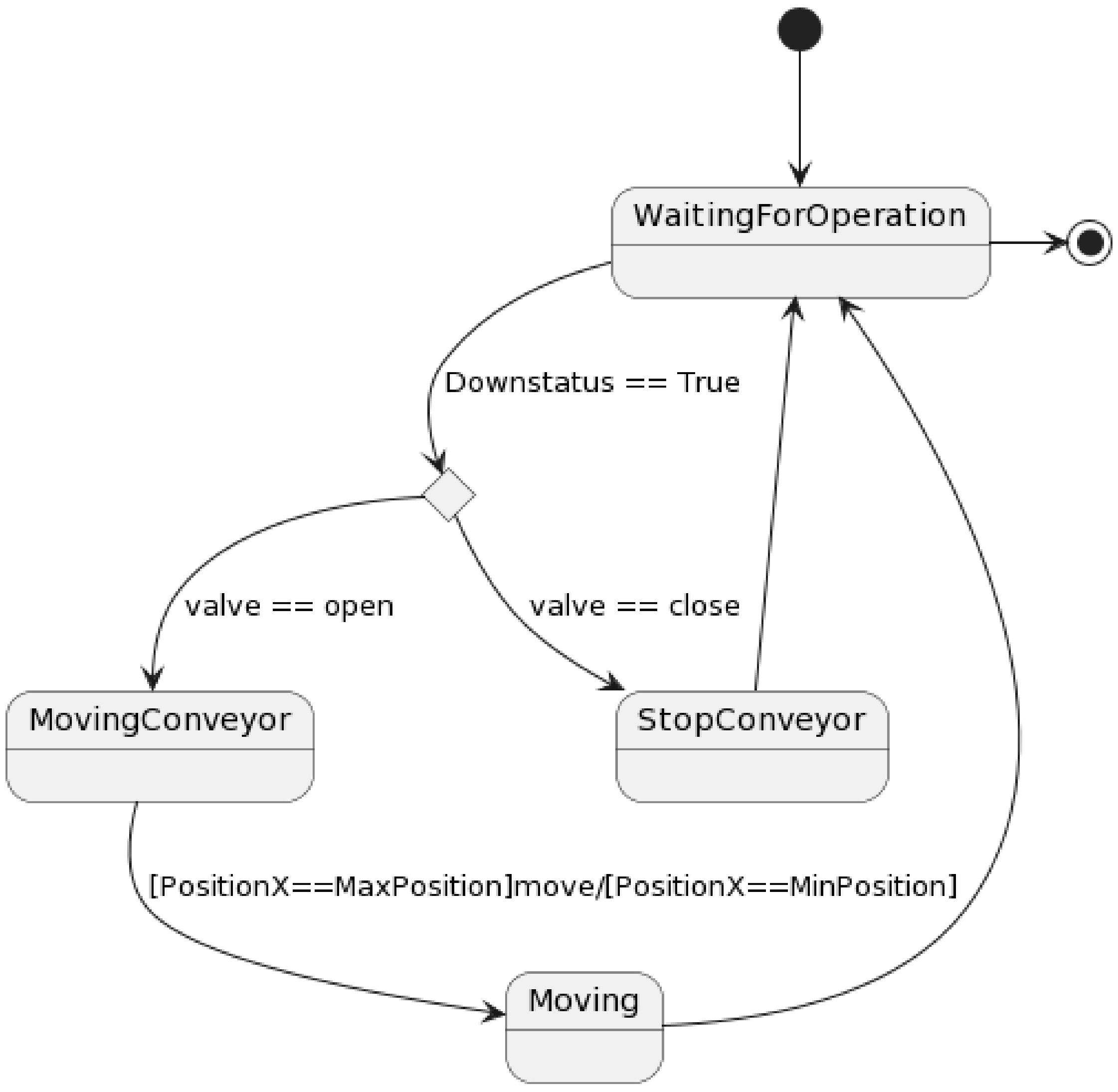


Figure 25: State diagram for DX10 machine

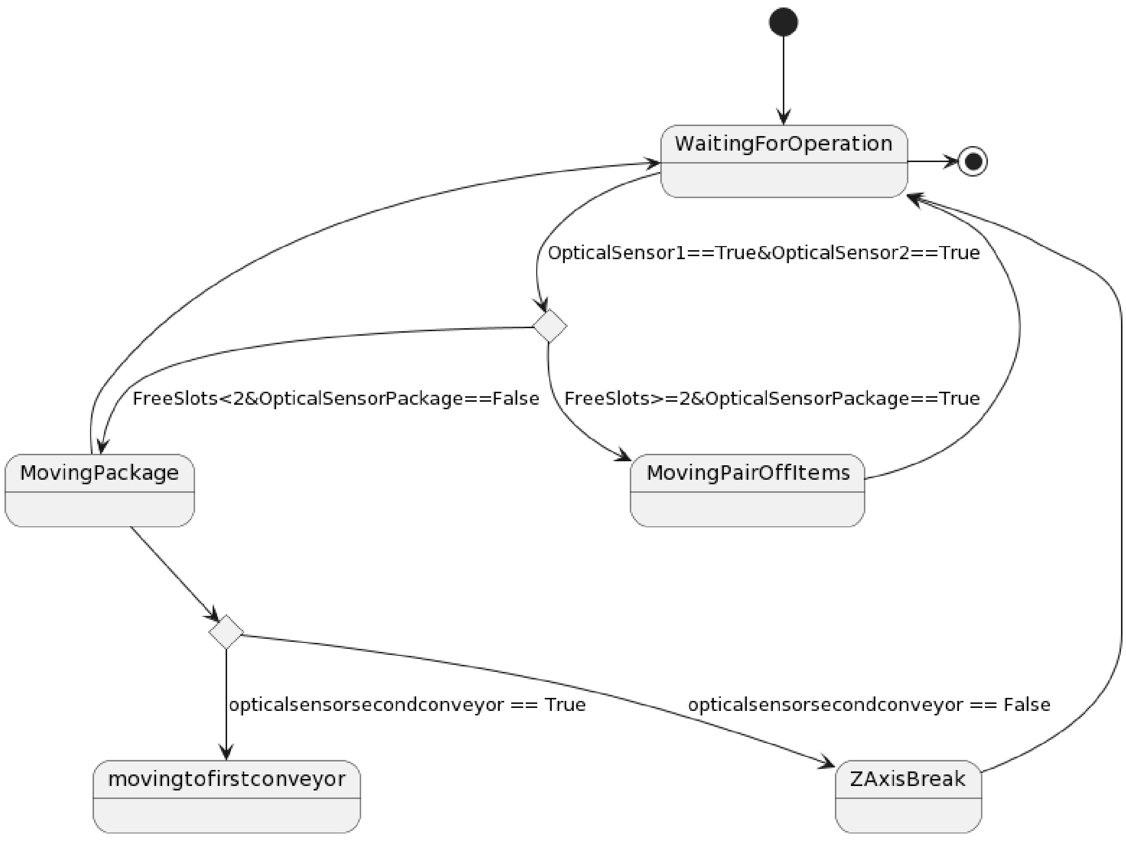

WaitingForOperation
OpticalSensor1==True&OpticalSensor2==True
FreeSlots<2&OpticalSensorPackage==False
FreeSlots>=2&OpticalSensorPackage==True
MovingPackage
MovingPairOffItems
opticalsensorsecondconveyor == True
opticalsensorsecondconveyor == False
movingtofirstconveyor
ZAxisBreak

Figure 26: State diagram for XY10 machine

8 Conclusion

The implemented Knowledge Graph helped to agents to control machines in industrial process by reading values and by knowing machine status from the entities. It is described by different vocabularies such as SSN/SOSA, Pto etc., The production rules and inference rules developed for crawling the knowledge graph are independent of the specific production line and the devices involved. An update on the production line does only require to revise the knowledge graph. The Graph is developed on top of WoT, which means that it can be easily extended with any Thing that exposes it's capabilities with the WoT TD. The overall system is developed with different components for the agents and the different types of artifacts. This means that new agents and artifacts can be easily added or replaced without affecting the overall system.

References

[1] Olivier Boissier, Rafael H Bordini, Jomi F Hübner, Alessandro Ricci, and Andrea Santi. Multi-agent oriented programming with jacamo. *Science of Computer Programming*, 78(6):747–761, 2013.

[2] Olivier Boissier, Andrei Ciortea, Andreas Harth, and Alessandro Ricci. Autonomous agents on the web. In *Dagstuhl-Seminar 21072: Autonomous Agents on the Web*, page 100p, 2021.

[3] Stefan Elmer, Foued Jrad, Thorsten Liebig, Anees ul Mehdi, Michael Opitz, Thomas Stauß, and Dirk Weidig. Ontologies and reasoning to capture product complexity in automation industry. In *ISWC (Posters, Demos & Industry Tracks)*, 2017.

[4] Pascal Hitzler, Markus Krotzsch, and Sebastian Rudolph. *Foundations of semantic web technologies*. Chapman and Hall/CRC, 2009.

[5] Sebastian Kaebisch and Takuki Kamiya. Web of things (wot) thing description w3c working draft 5 april 2018. *W3C Working Draft, W3C*, 2018.

[6] Tobias Käfer and Andreas Harth. Rule-based programming of user agents for linked data. In *LDOW@ WWW*, 2018.

[7] Mahda Noura, Valentin Siegert, and Martin Gaedke. Wat: Autonomous hypermedia-driven web agents for web of things devices. In *ATAC@ ISWC*, pages 38–43, 2021.

[8] Martin Ringsquandl, Evgeny Kharlamov, Daria Stepanova, Steffen Lamparter, Raffaello Lepratti, Ian Horrocks, and Peer Kröger. On event-driven knowledge graph completion in digital factories. In *2017 IEEE International Conference on Big Data (Big Data)*, pages 1676–1681. IEEE, 2017.

[9] Martin G Skjæveland, Anders Gjerver, Christian M Hansen, Johan Wilhelm Klüwer, Morten R Strand, Arild Waaler, and Per Øyvind Øverli. Semantic material master data management at aibel. In *International Semantic Web Conference (P&D/Industry/BlueSky)*, 2018.

[10] Baifan Zhou, Yulia Svetashova, Seongsu Byeon, Tim Pychynski, Ralf Mikut, and Evgeny Kharlamov. Predicting quality of automated welding with machine learning and semantics: a bosch case study. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 2933–2940, 2020.