Developing a reusable ontology for intelligent production facilities using Functional Object-Oriented Network

Scientific work to obtain the degree
Bachelor of Science (B.Sc.)

at the Human-centered Assistive Robotics
Technical University of Munich

Submitted by Rafik Ayari
on 26. 10. 2021

First supervisor: Univ.-Prof. Dr.-Ing. Dongheui Lee

Second supervisor: Matteo Pantano, David Paulius Ramos

TECHNISCHE UNIVERSITÄT MÜNCHEN
Human-centered Assistive Robotics
UNIV.-PROF. DR.-ING. DONGHEUI LEE

October 20, 2021

B A C H E L O R   T H E S I S

Developing a reusable ontology for intelligent production facilities using Functional Object-Oriented Network

Problem description:

With the a-priori defined 4th Industrial revolution (I4.0) the concept of connected factories is de-facto standard for competitive advantages of European manufacturing companies. However, the realization of this vision is still far from reaching companies. One of the pain points in the integration of this vision is the data representation due to the lack of a common data representation [1]. One of the application fields related to this problem is the one of human robot collaboration (HRC) as long it would benefit the collaboration and monitoring [2]. Some examples are present in the research field and one example is the functional object-oriented network (FOON) [3]. Therefore, in this bachelor thesis you will try to establishing a simple data structure for representing information in the field of HRC using the FOON. More specifically, you will focus in the use case of Learning from Demonstration (LfD) in an industrial task. Such operation should be represented by your data structure in an automatic way. For doing so you will select an object recognition algorithm and then you will try to propose a methodology for creating the data structure. Once the structure will be prepared you will focus on defining an ontology that can be shared so others can use your work. Your objectives can be summarized as follows:

Tasks:

- Literature research on object and action recognition
- Choice of a ontology for representation of information using the object recognition
- Proposal of a FOON extension with the ontology
- Implementation of the proposed framework on a factory alike infrastructure

Bibliography:

[1] Legat C., Seitz C., Lamparter S. and Feldmann S., "Semantics to the Shop Floor: Towards Ontology Modularization and Reuse in the Automation Domain", IFAC Proceedings Volumes, Volume 47, 2014, pp. 3444-3449, ISSN 14746670.

[2] Kaiser L., Schlotzhauer A. and Brandstötter M., "Safety-Related Risks and Opportunities of Key Design-Aspects for Industrial Human-Robot Collaboration", Springer International Publishing, 2018, pp. 95-104, ISBN 978-3-319-99582-3.

[3] Paulius D., Huang Y., Milton R., Buchanan W., Sam J. and Sun Y., "Functional object-oriented network for manipulation learning", 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2016, pp. 2655-2662.

Supervisor:	Prof. Dr. Dongheui Lee, M. Eng. Matteo Pantano, Ph.D. David Paulius Ramos
Start:	14/06/2021
Intermediate Report:	21/09/2021
Delivery:	26/10/2021

(D. Lee)
Univ.-Professor

Abstract

A decade ago, the work on the fourth industrial revolution started with one of its goals to make the production facilities autonomous and cooperative.
To achieve that, communication between industrial components needs to be established. One of the approaches considered for establishing the connection between the components are ontologies. Ontologies provide shared vocabulary to model a certain domain. Their drawback however is their lack of reusability, which brings us to the goal of our work to develop a reusable Ontology.
In the development of the ontology, we will use FOON, which is short to *Functional Object-Oriented Network*, this framework represents tasks with knowledge graphs containing object and motion nodes that will be used to create a linked data model representing the ontology. During our work few implementations will be done on the FOON to make it more reusable by integrating neural network structures for processing visual input to directly extract object and motion nodes defined in a FOON.

Contents

Chapter 1

Introduction

Within the last decade, the term Industry 4.0 (an abbreviation for the "fourth industrial revolution") was introduced, where the aim of this initiative is to innovate the industrial landscape as we know it by making the production facilities autonomous, dynamic and cooperative [10].

The concept envisages that the Factories of the Future (FoF) will be able to share information regarding processes and products that will enable life-cycle tracking of development, production, and services Deutsches Institut für Normung [1]. In order to achieve these benefits, arbitrary components need to communicate between each other, This unfortunately fails because information sources are heterogeneous and communication standards focusing on data have not been established in the automation domain [3]. Automation will play an important role in the domain of FoF, and the National Institute of Standards and Technology (NIST) has outlined the main challenges in the manufacturing interoperability program of NIST [6]. Among those, the most important to our work is the misinterpretation of definitions or meaning of terms when exchanging information, which is mainly due to the lack of common data models that can be easily shared among factories [7]. A well-recognised approach for establishing the semantics of information sources to make content more understandable for both humans and robots are ontologies. Ontologies available in the automation domain are however very specific and have limited usability, and our approach to solve these issues and develop ontologies with maximum reusability is modularization [3].

In this work, we will develop a reusable ontology using various frameworks.First, we will discuss prior or state-of-the-art works in the second chapter "State of the art". Secondly, in the third chapter "Concept" we address how these frameworks will be used together to develop the ontology. Thirdly "Implementation" chapter, in which we describe the work done on these frameworks to develop the ontology. Finally, the "Evaluation" chapter where we use a new dataset and run it on our work to evaluate our work.

Chapter 2

State of the art

2.1 Industry 4.0

The concept of Industry 4.0 (also known as the fourth industrial revolution) became widely known in 2011 when a consortium known as Industrie 4.0 devised strategies with the stated intention of improving the effectiveness of the German manufacturing industry [10]. The realization of the potential implementation of Industry 4.0 has become more prevalent due to the development of the Industrial Internet of things (IIoT), cyber physical systems and smart manufacturing. These allow diverse devices and technologies, such as wireless sensor networks, cloud systems, embedded systems and autonomous robots, to be interconnected. The interconnection of these applications allows for the real-time collection of production data from low-level devices upwards to enterprise applications. Although the concept of Industry 4.0 has been formulated and expanded on since 2011, there is no agreed upon definition among industry experts; however, integration has been identified as one of the key requirements to realize the vision of Industry 4.0 [2].

2.2 Ontologies

Ontologies were introduced twenty years ago by Gruber et al. (1993) as an explicit "specification of a conceptualization". An ontology provides a shared vocabulary, which can be used to model a domain by defining the objects and concepts that exist and specifying the properties and relations between these objects [3].

The use of ontology in the software engineering domain was initiated by the Semantic Web and Web Services Initiatives [12]. The purpose of the Semantic Web was to extend the description of the content of a web page with machine-interpretable data so that software agents could better search and process the information on the web page. The core technologies of Semantic web are the Resource Description Format (RDF) and Web Ontology Language (OWL). OWL is a widely recognized semantic language for creating and sharing ontologies. OWL is an extension of RDF, providing the abstraction mechanism for creating classes (groups of resources with similar characteristics) and their properties and constraints. Although RDF and OWL were primarily designed for the web, they were found useful in other domains as well for describing and sharing semantically structured knowledge [4]. For example, one of the ontologies developed on OWL is SIARAS (Skill-based Inspection and Assembly for Reconfigurable Automation Systems) ontology, presented in Figure 2.1, in which manipulation and handling skills of robots are presented.

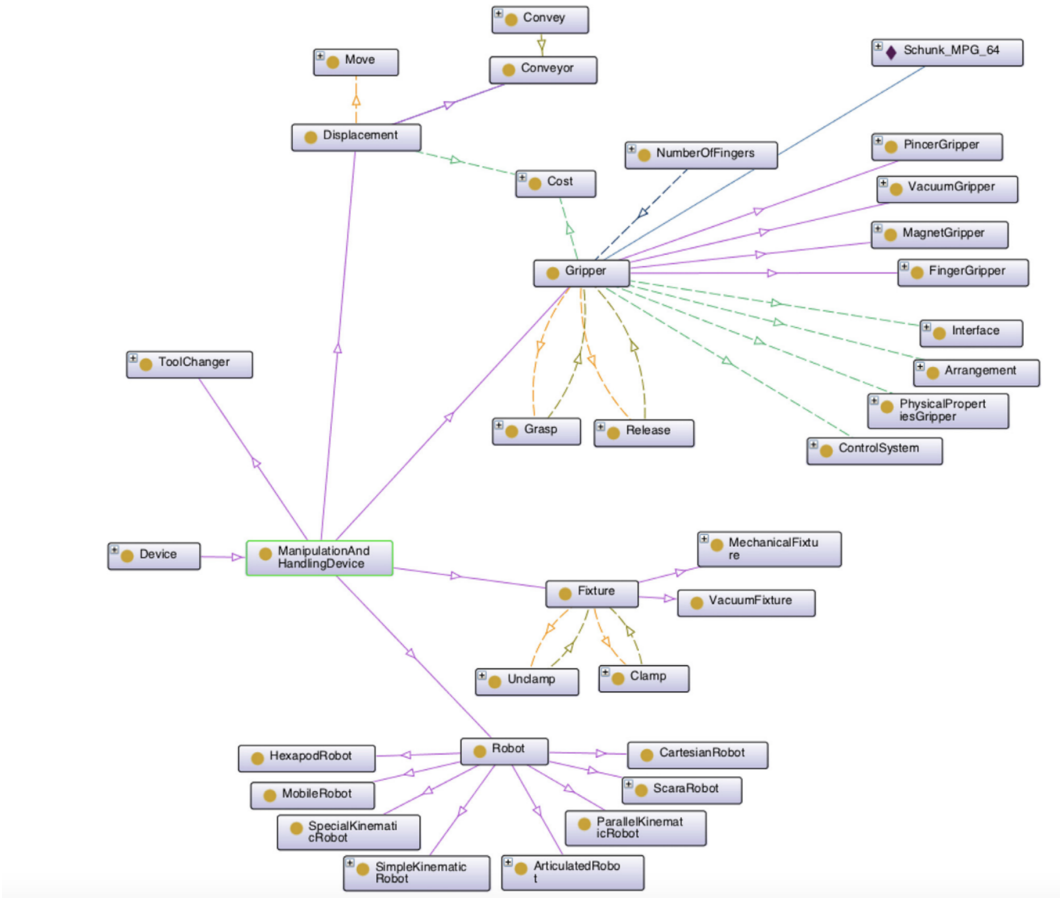


Figure 2.1: Manipulation and handling skills, as defined by SIARAS ontology [15].

Developing ontologies is nowadays considered as a standard activity in research projects dealing with semantics. Unfortunately, this is not a common result of applied projects, where the effort and knowledge required to develop an ontology from scratch is considered not sustainable, in respect of the expected benefits, which is a major drawback in the representation of the connection in automation domain. Hence, the goal of this thesis is to develop a reusable ontology.

2.3 FOON: Functional Object-Oriented Network

FOON is a framework that represents object state change in manipulation tasks and models the connectivity of functionally-related objects and their motions in the form of a subgraph. This subgraph is presented in the form of a bipartite network containing *objects and motion nodes*, as shown in Figure 2.2. In manipulation tasks, a FOON graph is learned by observing object state change and human manipulations upon the objects. A functional unit is considered as the minimum learning unit in a FOON. It represents the relationship between one or several objects and a single functional motion associated to the objects. In other words, each unit represents a single, atomic action that is part of an activity.The object nodes connected with the edges pointing to the functional motion node are called input object nodes, while the object nodes connected with the edges pointing from the functional motion node are called output object nodes [9].

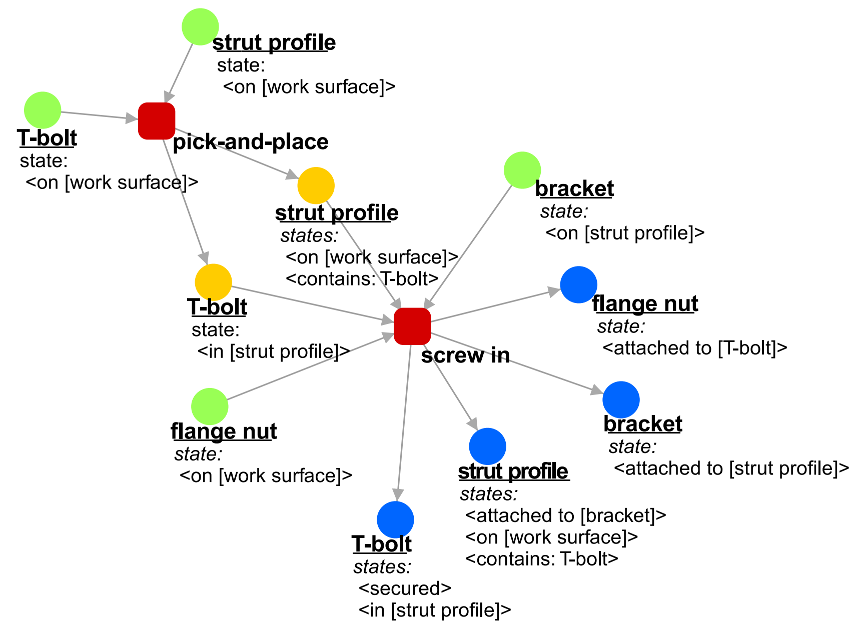


Figure 2.2: Two functional units, first for placing a bolt onto a strut profile and second screwing in the bolt onto the profile. The functional units have input nodes (green) and output nodes (blue) and are connected by two intermediary single motion nodes.

A FOON that comprises of the steps observed in a demonstration for a single activity is known as a *subgraph*; many subgraphs can be combined into a large source of information, which is known as a universal FOON. Knowledge graphs such as FOON encode contexts, expose connections and relations among entities and natively support inference and causation, and hence, they could be used for creating better representations of data [9].

2.4 Collaborative robotics

Collaborative robotics is a branch of robotics that focuses on the use of robots (cobots) together with humans, rather than robots working alone. These systems are believed to be an effective way to bring humans and robots together to improve safety by limiting the risks of injury due to human error or malfunctioning machinery [13].
However, current methods and tools for designing robotics applications are at best only able to help answer individual questions but do not allow support to study the system as a whole. As a result, it currently takes very long to develop and implement a simple application featuring collaborative robotics, there is a lot of uncertainty during that process, and there are no means for determining whether the design decisions made are the best. Figure 2.3 shows the overall flow of the design. There are feedback loops that check the requirements before moving on to the next step. In general, the design is more precise because the additional process, safety, and standard requirements are observed.
In industry, there are several field-specific tools for designing participatory robot applications. The standard procedure includes a project manager who works with mechanical and electrical engineers. Mechanical engineers work in Computer Aided Desing (CAD), typically with robot simulation environments or plugins. They show the general planning and the flow of materials and determine the type of cooperation with the human by determining his duties. They check whether the basic requirements are met and select the robot type based on several criteria such as load, access, and customer-specific preferences. The electrical system can be designed in an electronic Computer-Aided Design (ECAD) program. As a rule, there is no direct digital connection between the mechanical and the electrical field. Electrical components are physically formed and cable routes are designed by machine builders. Logical paths, communication, and electrical power can be modeled in ECAD. Once the system is ready, the programmer can start programming the robot. They rely on a variety of tools and frameworks and can sometimes use pre-built simulations for their own purposes.

2.5 Activity recognition

Human activity recognition (HAR) has been a very active research topic in the past two decades for its applications in various fields such as health, remote control, gaming, security, human-computer surveillance and interaction. Activity recognition can be defined as the ability to detect current activity based on information received from various sensors [5]. These sensors can be cameras, wearable sensors or sensors attached to everyday objects. With technological advances and reduced hardware costs, recording daily activities have become very popular and convenient. To capture these activities, different approaches have been used. These approaches can be broadly classified into vision-based and sensor-based [5] as shown in Figure

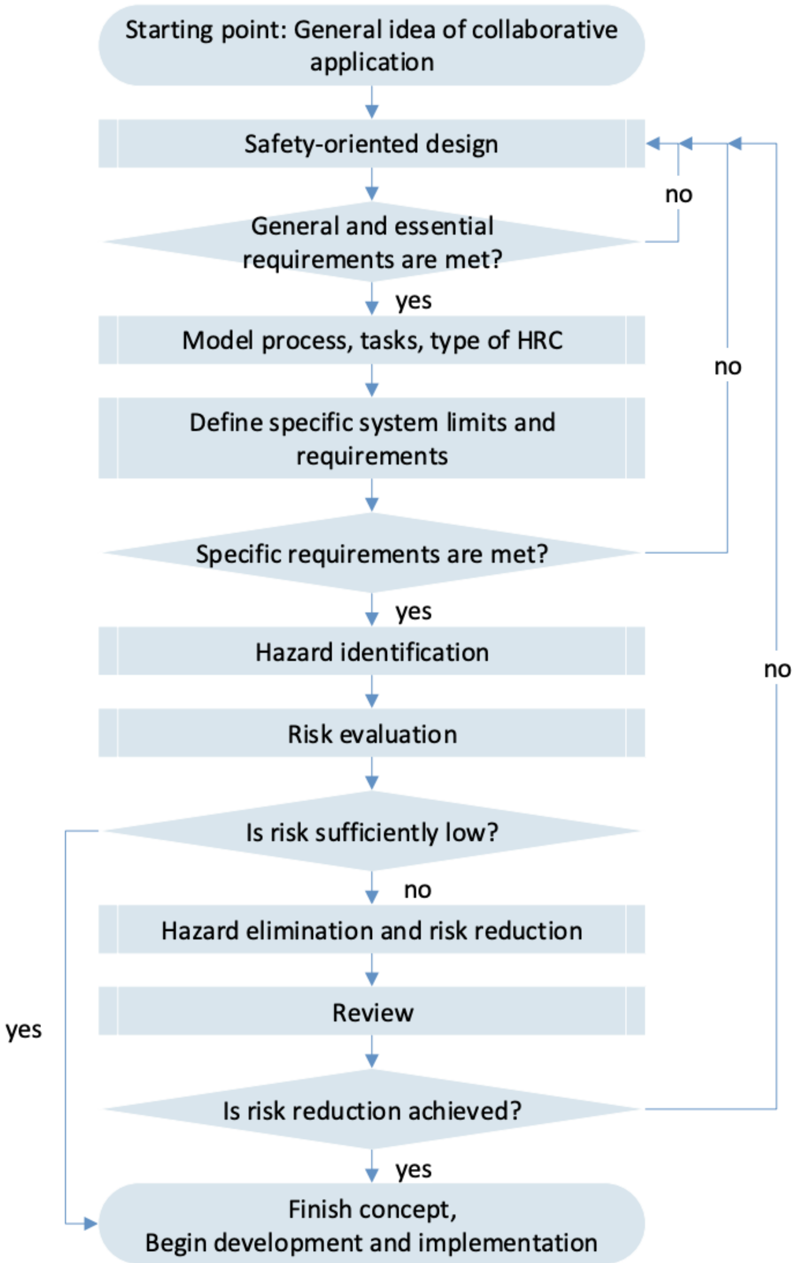


Figure 2.3: Flow model of different phases during concept (design) of a HRC (Human Robot Collaboration) application in manufacturing [13].

2.4.

In recent years, researchers have made extensive use of Convolutional Neural Networks (CNNs) for image classification problems. Given the success of CNNs in classifying the image and its content, the researchers used CNN to further classify videos of everyday activities. Classifying realistic videos into arbitrary free-form ac-

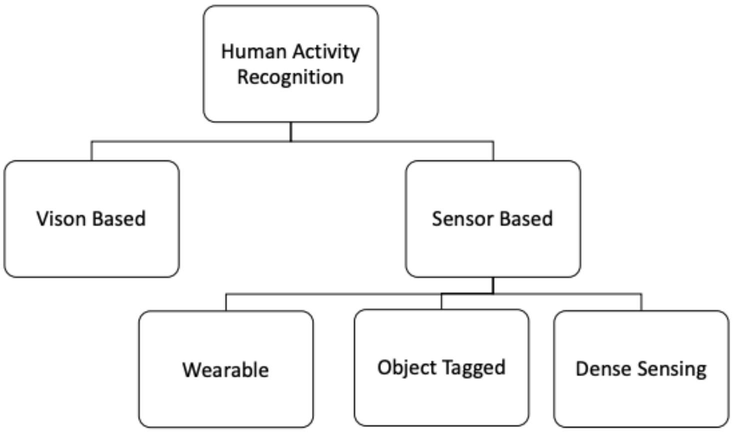


Figure 2.4: Classification of human activity recognition approaches [5]

tivities is a daunting task, mainly due to lighting conditions, congestion, background clutter, distortion, angle of view, size and contrast between layers. in this thesis we will focus on YOLO, one of various NNs available for object detection.

2.5.1 YOLO: You Only Look Once

YOLO is a single neural network that predicts bounding boxes and class probabilities directly from full images in one evaluation [12]. As shown in Figure 2.5, a single convolutional network simultaneously predicts multiple bounding boxes and class probabilities for those boxes.

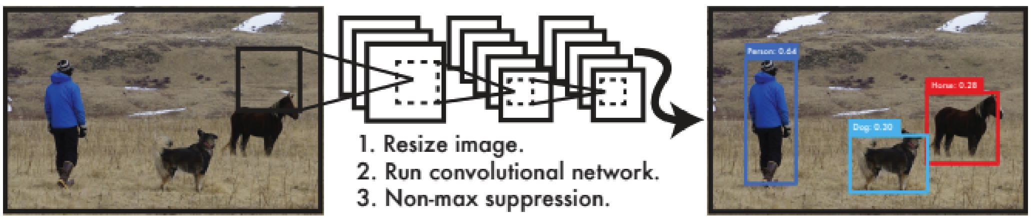


Figure 2.5: The YOLO Detection System [12]

YOLO trains on full images and directly optimizes detection performance. This unified model has several benefits over traditional methods of object detection.First YOLO is extremely fast as the base network runs at 45 frames per second (FPS) and the fast version (Fast YOLO) runs at more than 150 FPS.Second, YOLO reasons globally about the image when making predictions. Unlike sliding window and region proposal-based techniques, YOLO sees the entire image during training and test time, so it implicitly encodes contextual information about classes as well as their appearance.Third, YOLO learns generalizable representations of objects. When trained on natural images and tested on the artwork, YOLO outperforms top detection methods like DPM and R-CNN by a wide margin. Since YOLO is highly generalizable, it is less likely to break down when applied to new domains or unexpected inputs.

Chapter 3

Concept

Our approach to develop a reusable ontology, as shown in Figure 3.1, is integrating learning from demonstration into the FOON. Through the analysis of an I4.0 related industrial case, a FOON subgraph can be created afterwards, where the nodes of the subgraph represents the parts used in the industrial case. Their properties can be used to create a linked data model defining the ontology.

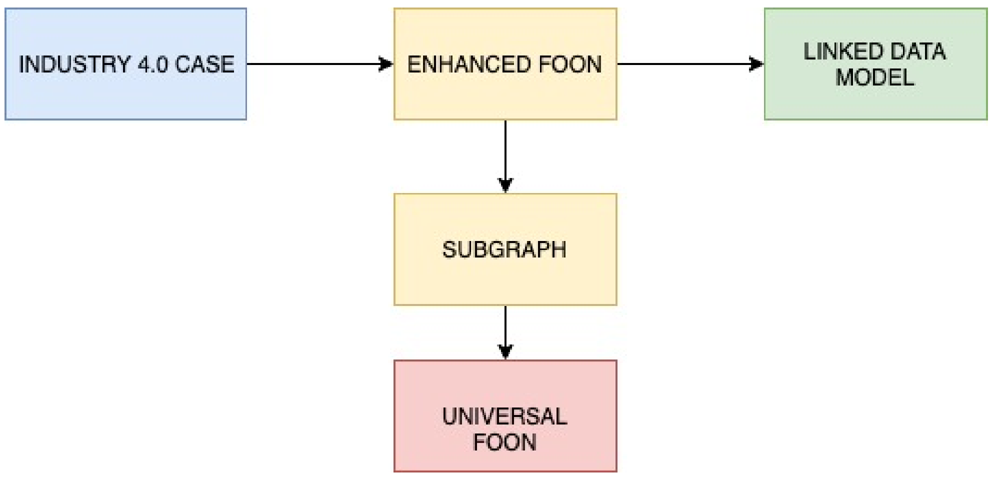


Figure 3.1: Overview of the Ontology

3.1 Enhancing the FOON

A FOON on its own is solely comprised of symbols and labels, which have no meaning on their own to a robotic system unless we give them one. The goal of the ontology is to encapsulate different features and properties together into a format that can be understood by a robot and also a smart system as a whole. Our approach is to enhance the use of the FOON with a system that will give meaning to visual input from YOLO and connect it to nodes in a FOON. One of our goals by enhancing the FOON is to solve problems like robot understanding of what humans do or understanding its own actions, and implementing action recognition will help the robot identify the actions performed by a human demonstrator or worker. YOLO and hands recognition will help us identify the objects that are being used and in which state(s) they may currently be in as it relates to the task.

3.2 Linked data model

In order to satisfy the need for an activity description aligned to an industrial-like infrastructure, semantically linked activity information model was employed. The ability to translate a semantically linked model to the OPC UA information model was a driving factor in this choice [11]. Considering that we are using the FOON comprised of several *functional units*, our modelling proposes an approach to build and connect *functional units*. For our modelling, we start from the *Task* data model[1] and *Resources* data models[2] created by the Smart Human Oriented Platform for Connected Factories (SHOP4CF) consortium [8]. These information models were selected because they are based on linked data and they represent manufacturing use cases; therefore, they satisfy the requirements for mapping into OPC UA. The *Task* data model was chosen due to its similar definition to a *motion node*: a *Task* is intended to be a manufacturing operation that may contain sub-steps, and it is associated to resources (i.e., persons, devices, materials, and assets) and locations. The *Resources* data models were also picked because of their notion of abstractly representing objects on the shop floor. As a result, the mapping between *functional unit*, *Task*, and *Resource* properties can be summarized as follows. The *Resource* data models are mapped to the input and output objects because they can reflect the state of the objects (i.e., state change). The *Task* data models are used to represent motion nodes because they may indicate which item is involved via the *involves* property and integrate the sequence of events via the *isDefinedBy* property. the object state is updated according to the perceived state through the attribute state and the Task with Resources are published on the FIWARE context broker. This approach is described in the following algorithm.

[1]`https://shop4cf.github.io/data-models/task.html`
[2]`https://shop4cf.github.io/data-models`

Algorithm 1 Connection of the FOON to the ontology

```
procedure FOON2ONT(PredictedUnit)
    Resources = []
    Task = []
    for each object obj in PredictedUnit do
        Resource[state] = obj[state]
        Resources.append(Resource)
        Task[involves] = obj[id]
    end for
    Task[isDefinedby] = PredictedUnit[id]
    publish(Task)
    publish(Resources)
end procedure
```

Chapter 4

Implementation

4.1 Training YOLO

In order for us to use YOLO to detect objects in industrial manipulation tasks, we trained the neural network using a dataset containing videos of manipulation tasks in industrial environment,Figure 4.1 bellow shows the objects used in our task.

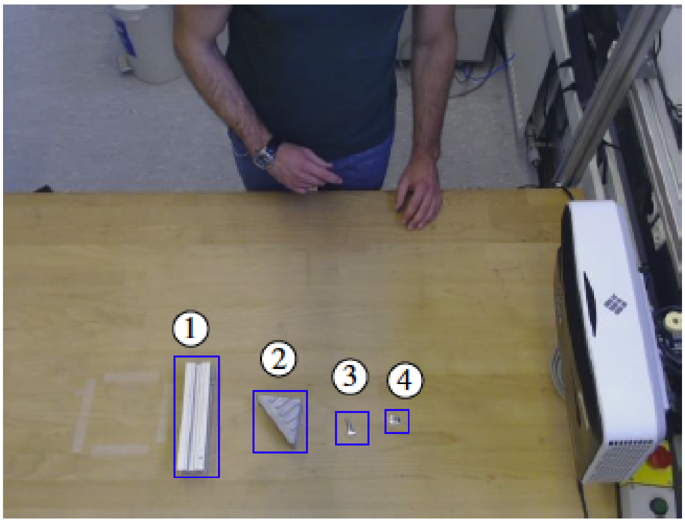


Figure 4.1: Image of the workbench with the operator and the parts used for the assembly process. Going from the left to the right, the parts are as follows: strut profile 1, bracket 2, T-bolt 3, and flange nut 4.

The videos were divided into frames; using VOTT [1], a web application used to label images or video frames and extract labeled data to local or cloud storage providers, the frames were labeled as shown in Figure 4.2 and a CSV file containing all the frames, the objects in them and the coordinates of the bounding boxes of said objects was created.

[1] `https://github.com/microsoft/VoTT`

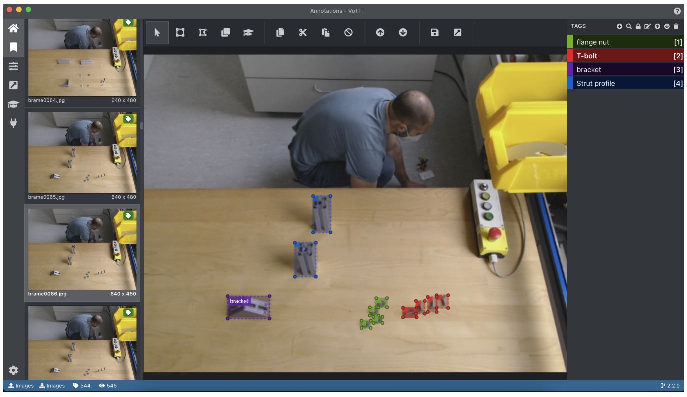
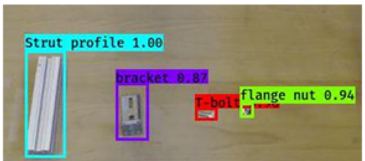


Figure 4.2: Labeling frames using VOTT web application

The CSV file and the frames were used to train the YOLO network; once training was achieved, YOLO was able to detect the objects on new frames and output the frames with bounding boxes drawn on them.A CSV file with the coordinates of the boxes as shown in Figure 4.3.



frame0026.jpg	/content/drive/MyDrive/TrainYourOwnYOLO/Data/Source_Images/Test_Images/frame0026.jpg	342	374	365	387	3	0.9619541168212891	640	480
frame0026.jpg	/content/drive/MyDrive/TrainYourOwnYOLO/Data/Source_Images/Test_Images/frame0026.jpg	389	369	403	384	2	0.9427311420440674	640	480
frame0026.jpg	/content/drive/MyDrive/TrainYourOwnYOLO/Data/Source_Images/Test_Images/frame0026.jpg	259	348	293	408	1	0.8685104250907898	640	480
frame0026.jpg	/content/drive/MyDrive/TrainYourOwnYOLO/Data/Source_Images/Test_Images/frame0026.jpg	164	314	206	426	0	0.9956373572349548	640	480

Figure 4.3: Frame with bounding boxes and the exported dataframe with the coordinates of the boxes.

4.2 Hand detection

Since the hands can change size and shape from the type of grasp, training YOLO manually with VOTT with the frames did not give optimal results, as it did not detect the hands if they were observed at different angles or in different grasp types; to have better results we decided to use a pre-trained YOLOv3 network trained to detect hands using CMU Hand DB dataset [2] with more than 14000 annotations, which returned good results when tested with our frames, as shown in Figure 4.4.

[2] http://domedb.perception.cs.cmu.edu/handdb.html

Figure 4.4: Detected hands using the pre-trained YOLO network

4.3 Bounding boxes and distance matrices

Once the YOLO neural network detects an object/hand in a frame, it adds its coordinates to a CSV file, from which we extract the coordinates (X,Y) of the center of the bounding box. This point is where almost exactly lays the object/hand in the frame, and by going through all the detected points in the frame, we can create an object-to-object and object-to-hand matrix as shown in Figure 4.5.

frame0039

	Hand_r	Hand_l	T-bolt	flange nut	bracket	srut profile
Hand_r		239.30106560565082	202.18061232472317	238.62313383240948	136.61991070118586	67.468511173732
Hand_l	239.30106560565082		207.63910999616618	201.20139164528658	231.36551169091732	265.58049627184596
T-bolt	202.18061232472317	207.63910999616618		42.190046219457976	78.02563681252464	167.35889578985635
flange nut	238.62313383240948	201.20139164528658	42.190046219457976		120.01666550941998	208.38665984174708
bracket	136.61991070118586	231.36551169091732	78.02563681252464	120.01666550941998		90.80198235721508
srut profile	67.468511173732	265.58049627184596	167.35889578985635	208.38665984174708	90.80198235721508	

Figure 4.5: Object to Object and Hand to object matrix

4.4 Recurrent neural network: LSTM

In order to train a long short-term memory (LSTM) to recognize the actions, on one hand, the distance matrix of the frames were used as input and on the other, output labels referring to the actions being done on the frame were used. These labels were handmade according to a FOON network representing the manipulation task. The input to the LSTM must be three-dimensional samples, time-steps and features. The distance matrices of the frames are put in a list and each list of a frame represents a sample, where each time-step contains 5 lists of 5 frames. The output of the LSTM is an array extracted from the label with the shape `1,number` `of labels`. Once the LSTM trained, the model was able to predict from a sequence of 5 frames, the array representing the action being done on the last frame. A second LSTM is trained with the same input to predict the objects being used in the action. The output of this LSTM is an array extracted from handmade labels with the shape `1, number of objects`. The frame in the Figure 4.6 bellow is an example of the output results.

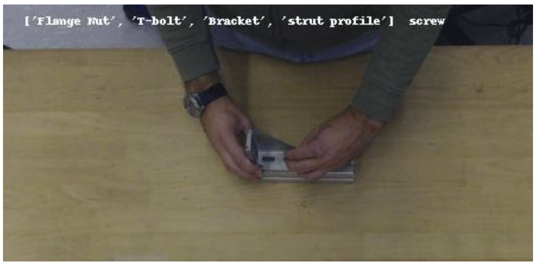


Figure 4.6: Labeled frame

4.5 Creating the Linked Data model

From a dataset containing frames of a manipulation task in an industrial environment, using the YOLO and the LSTMS, a CSV file is created that contains the frame number, the label number representing the action and the objects used in the action. For the creation of the linked data model we chose the JSON-LD (JavaScript Object Notation for Linking Data) format. JSON-LD[3]is a lightweight linked data format that is based on the already successful JSON format. This format has the advantage of being easy to read or write by human users while also being easy to process and parse by machines.

The functional units were extracted from the CSV file, which were used to created our JSON file as shown in the Figure 4.7 above.

[3] `https://github.com/json-ld/json-ld.org`

```
{
    "id": "urn:ngsi-ld:Task:tum:41347",
    "type": "Task",
    "isDefinedBy": {
        "type": "Relationship",
        "object": "urn:ngsi-ld:TaskDefinition:tum:1"
    },
    "involves": {
        "type": "Property",
        "value": [
            {
                "type": "Relationship",
                "object": "urn:ngsi-ld:Material:tum:[Strut profile]"
            }
        ]
    },
    "@context": [
        "https://smartdatamodels.org/context.jsonld",
        "https://raw.githubusercontent.com/shop4cf/data-models/master/docs/shop4cfcontext.jsonld"
    ]
}
```


Figure 4.7: Example of JSON-LD file

Chapter 5

Evaluation and discussion

5.1 Evaluation

For the evaluation of our work we will use a new dataset containing frames of a manipualtion task in an industrial environment and will run it through the different steps presented in the Figure 5.1 bellow.

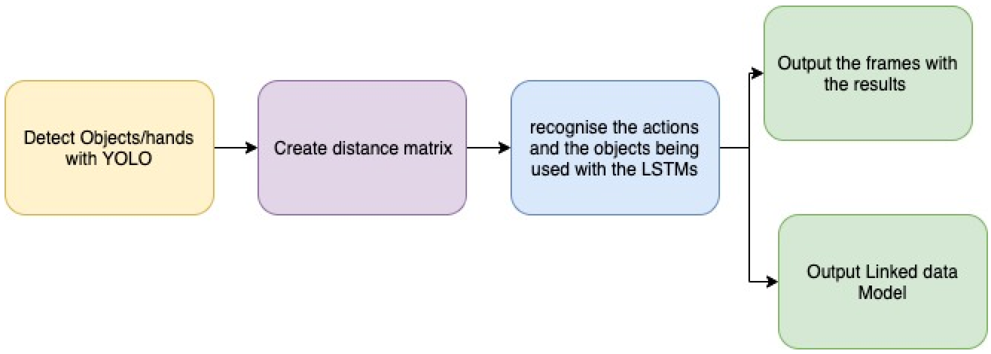


Figure 5.1: Diagram representing the different steps of the evaluation of our work

First, the frames were run through the YOLO networks to detect the objects and hands, which will be output in a csv file containing all the coordinates of the objects in the frames, from which the distance matrices will be created and will serve as input to the LSTMs to predict the actions and the objects used. These results were written on the frames so they can be evaluated by us and a linked data model was created with them so it can be run on FIWARE. From the CSV file containing the results of the YOLO neural networks, the objects and the hands were detected in 90 out of the 102 frames. In some cases, objects were not detected due to occlusion by the hands. Some of the more challenging objects to detect include "T-bolts" and "Flange nuts", mainly because of their small size.
The results of the activity recognition LSTM are shown in the table in Figure 5.2,

out of the 102 frames, the LSTM predicted the correct activity in 86 frames with an accuracy of 84%, the incorrect predictions are due to a delay in the prediction of 2-3 frames per new activity.

	Correct activity	Incorrect activity
Activity detected	84%	16%

Figure 5.2: Overview of the results of the activity recognition LSTM.

The results of the active objects prediction LSTM are shown in the confusion matrix bellow, it predicted the objects being used during the activity with an accuracy of 81%, like the activity recognition LSTM, this LSTMs incorrect predictions are due to a delay of 3-4 frames every time a detected object is not being used anymore. These delays can be reduced with more training of the LSTMs.

		True	
		Object being used	Object not being used
Predicted	Object being used	73%	13%
	Object not being used	6%	8%

Figure 5.3: Confusion matrix of the results of the activ objects LSTM.

5.2 Discussion and Future works

The results of the trained networks are satisfying but can be improved upon by increasing the dataset used for the training. Additionally the multiple neural networks make the creation of a linked data model a little time consuming, which can be improved by implementing a neural network to detect the grasp of the hands. This will provide us with not only the coordinates of the hands but also the used objects in this way, we can avoid using a second LSTM network and improve the results of the action recognition LSTM by adding additional information and will fix the previous issue of hands covering the objects from the camera. Other types of LSTM networks should also be tested, such as ConvLSTM [14] in which we can input video frames directly into the network in addition to the distance matrices, so it matches each matrix to its corresponding frame and thus the prediction will be faster, have more accurate results and provide labeled frames directly from the LSTM.

Chapter 6

Conclusion

In this work, we investigated with the purpose of making a reusable ontology, using the FOON, Functional Object-Oriented Network, that provides a promising manner of data representation with its knowledge graphs that encode contexts, expose connections and relations amongst entities. The FOON was enhanced with few implementations to make the data model reusable, which consist of object and action recognition, directly from frames of tasks in an industrial environment to give meaning to visual input and link it to nodes in the FOON. The object and motion nodes are used to create our linked data model that represents our ontology, under the JSON-LD format that is easy for operators to read and write.

The assessment of our work using a brand new set of frames of a similar industrial task showed accurate results. However for our work to be beneficial in an industrial environment the object and activity recognition networks may be improved so they can provide faster results. Improvements can be achieved by comparing results with other types of Recurrent Neural Networks and training the Convolutional Neural Networks with new objects and tasks to provide a wider range of reusability.

List of Figures

Acronyms and Notations

HRC Human-Robot Collaboration

HRI Human-Robot Interaction

FOON Functional Object-Oriented Network

YOLO You Only Look Once

VTT Visual Object Tagging Tool

LSTM Long Short Term Memory

NN Neural Network

JSON-LD JavaScript Object Notation for Linking Data

SIARAS Skill-based Inspection and Assembly for Reconfigurable Automation Systems

RDF Resource Description Format

OWL Web Ontology Language

fof Factories of the Future

NIST National Institute of Standards and Technology

IIot Industrial Internet of things

ECAD Electronic Computer-Aided design

FPS frames Per Second

Bibliography

[1] Deutsches Institut für Normung. Referenzarchitekturmodell Industrie 4.0 (RAMI4.0), 2016.04.

[2] T. B. J. C. F. Doyle. A Review of Interoperability Standards for Industry 4.0. *Procedia Manufacturing*, 38:646â–653, 2019.

[3] C. L. C. S. S. L. S. Feldmann. Semantics to the Shop Floor: Towards Ontology Modularization and Reuse in the Automation Domain. *The International Federation of Automatic Control*, pages 3444–3449, 2014.

[4] O. Harcuba and P. Vrba. Ontologies for flexible production systems. In *2015 IEEE 20th Conference on Emerging Technologies Factory Automation (ETFA)*, pages 1–8. IEEE, 2015.

[5] Z. Hussain, M. Sheng, and W. E. Zhang. Different approaches for human activity recognition: A survey. *CoRR*, abs/1906.05074, 2019.

[6] S. J. Kemmerer. *Manufacturing Interoperability Program, a Synopsis.* NIST Interagency/Internal Report (NISTIR), National Institute of Standards and Technology, Gaithersburg, MD, Gaithersburg, MD, 2009-02-24 2009.

[7] C. Legat, C. Seitz, S. Lamparter, and S. Feldmann. Semantics to the Shop Floor: Towards Ontology Modularization and Reuse in the Automation Domain. *IFAC Proceedings Volumes*, 47(3):3444–3449, 2014.

[8] Micha Zimniewicz. Deliverable 3.2 - SHOP4CF Architecture.

[9] D. Paulius, Y. Huang, R. Milton, W. D. Buchanan, J. Sam, and Y. Sun. Functional object-oriented network for manipulation learning. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2655–2662. IEEE, 2016.

[10] S. Pfeiffer. The vision of industrie 4.0 in the making a case of future told, tamed, and traded. *NanoEthics*, 11, 04 2017. doi: 10.1007/s11569-016-0280-3.

[11] M. Piattini, C. Calero, and F. Ruiz. *Ontologies for Software Engineering and Software Technology.* 01 2006. ISBN 3-540-34517-5. doi: 10.1007/3-540-34518-3_4.

[12] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[13] J. Saenz, N. Elkmann, O. Gibaru, and P. Neto. Survey of methods for design of collaborative robotics applications-why safety is a barrier to more widespread robotics uptake. In *Proceedings of the 2018 4th International Conference on Mechatronics and Robotics Engineering*, pages 95–101, 2018.

[14] X. Shi, Z. Chen, H. Wang, D. Yeung, W. Wong, and W. Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *CoRR*, 2015.

[15] M. Stenmark and J. Malec. Knowledge-based instruction of manipulation tasks for industrial robotics. *Robotics and Computer-Integrated Manufacturing*, 33: 56–67, 2015.

License