Analysis of metadata standards for the exchange of image datasets and algorithms in the agricultural domain

Authors | Daoud Urdu, Daan Goense, Johan Booij, Conny Graumans and Corné Kempenaar

WPR-OT-1019

WAGENINGEN
UNIVERSITY & RESEARCH

Analysis of metadata standards for the exchange of image datasets and algorithms in the agricultural domain

A metadata-oriented approach to identify minimum interoperability mechanisms for image data and deep learning algorithms that is used for vision-based applications in agriculture. Sprint Robotics Project PL4.0 WP7

Daoud Urdu[1], Daan Goense[2], Johan Booij[1], Conny Graumans[2] en Corné Kempenaar[1]
Reviewed by Ard Nieuwenhuizen[1]

1 Wageningen University & Research
2 AgroConnect

This study was carried out by the Wageningen Research Foundation (WR) and was commissioned and financed by PPS Precisielandbouw 4.0: op naar datagedreven landbouw voor rotatie management.

WR is part of Wageningen University & Research, the collaboration of Wageningen University and Wageningen Research Foundation.

Wageningen, April 2023

Report WPR-OT-1019

WAGENINGEN
UNIVERSITY & RESEARCH

Urdu, D; Goense, D.; Booij, J.A.; Graumans, C.; Kempenaar,C., 2023. *Analysis of metadata standards for the exchange of image datasets and AI in the agricultural domain.* Wageningen Research, Report WPR-OT-1019.

This report can be downloaded for free at https://doi.org/10.18174/632203

Summary

This report discusses the importance of precision agriculture in achieving sustainability goals and the need for a basis that considers different perspectives of a data space such as interoperability, scalability, security, transparency, and data ownership. The Towards Precision Agriculture 4.0 project aims to address these perspectives to provide better-informed management decisions for farmers and the ecosystem. The current study focuses on determining minimum interoperability mechanisms concerning the standardization of image data and deep learning algorithms for vision-based applications in weed management by robots. The study adopts a metadata-oriented approach to make data and algorithms semantically interoperable and reuses existing knowledge from the Reference Model Agro (rmAgro). The results indicate the need for a balance between established standardization and agile standardization for supporting semantic interoperability, and the interoperability of preferred standards like Robot Operating System (ROS) and Open Neural Network Exchange (ONNX) is insufficient. The study results are useful for professionals and academia who work in the design and development of software for the farming business.

Keywords: standardisation, Agro-food robotics, weed control, machine vision, interoperability

© 2023 Wageningen, Stichting Wageningen Research, Wageningen Plant Research, P.O. Box 16, 6700 AA Wageningen, The Netherlands; T +31 (0)317 48 07 00; www.wur.eu/plant-research

Chamber of Commerce no. 09098104 at Arnhem
VAT NL no. 8065.11.618.B01

Stichting Wageningen Research. All rights reserved. No part of this publication may be reproduced, stored in an automated database, or transmitted, in any form or by any means, whether electronically, mechanically, through photocopying, recording or otherwise, without the prior written consent of the Stichting Wageningen Research.

Stichting Wageningen Research is not liable for any adverse consequences resulting from the use of data from this publication.

Report WPR-OT-1019

Photo cover: The autonomous 5G robot solution for weed control

Contents

Summary

To support achieving sustainability goals with precision agriculture, there is an urgent need for a basis that takes different perspectives of a data space into account, such as interoperability, scalability, security, transparency, and data ownership. The project Towards Precision Agriculture 4.0 aims to address these perspectives to arrive at a basis that forms new knowledge to design data-sharing principles and to provide better-informed management decisions for the farmer and the ecosystem around the farmer. The main issue that is addressed within this study is to map all the data that is generated on the farm and design pathways that could potentially add value for the farmer with these data for his operational, tactical, and strategic decisions. The proposition is that data harmonization is an essential condition to achieve this.

The current study's purpose was to determine minimum interoperability mechanisms concerning the standardization of image data and deep learning algorithms for vision-based applications.
This study builds further upon earlier work in the same context, which suggests an architecture for data exchange within a FAIR data ecosystem. The architecture supports publishing interoperable algorithms and data and describes the playing level of actors in such an ecosystem (Booij, et al. 2022). In specific, this study further describes, analyses, and models the various data streams and relevant metadata that are part of the domain of weeding robots. Legal regulations, such as the General Data Protection Regulation (GDPR), are considered when performing these activities to safeguard the rights of a user, especially where data can be linked to a person, such as a farmer.

A metadata-oriented approach around the case is adopted in this study to make data and algorithms semantically interoperable and the approach resulted in relevant models using different modelling techniques such as process models with Business Process Modelling Notation (BPMN) and class diagrams with Unified Modelling Language (UML). For the modelling part, existing knowledge is reused from the Reference Model Agro (rmAgro), which is a well-known normative standard in the agricultural domain since the 1970s. Although the reference model is rich in explicit knowledge, this study suggests adopting approaches and standards for semantic interoperability following the Web 3.0 paradigm, using web technologies like Resource Definition Framework (RDF), Web Ontology Language (OWL) and SPARQL Protocol and Query Language (SPARQL).

The most obvious finding from this study is that the specification of metadata is insufficient for some standards for weed management by robots. The candidate processes, messages, dataflows, and classes that are identified and modelled in this study could facilitate data sharing between actors in the proposed ecosystem. The results indicate an increased potential between established standardization and agile standardization for supporting semantic interoperability.

This report is especially useful for professionals and academia who work in the design and development of software for the farming business. Several standards were analyzed, and the most important finding was that each standard covered part of the domain of interest and that the creators of these standards use different methods. In the domain of interest, the interoperability of preferred standards like Robot Operating System (ROS) and Open Neural Network Exchange (ONNX) is insufficient.
Our analysis and review of existing standards have resulted in four new process models and a sub-reference model containing multiple class diagrams that specify the domain of computer vision and robotics, considering rmAgro and preferred communication protocols like ROS and ISOBUS. The processes, messages, dataflows, and classes that are identified and modelled in this study indicate a first attempt of specifying the domain and could facilitate data sharing between actors in the proposed ecosystem.

1 Towards Precision Agriculture 4.0: data-driven agriculture

The project 'Towards Precision Agriculture 4.0' is a public-private partnership which focuses on a basis for large-scale, smart, secure, transparent and in-control use of data with which (1) new knowledge is obtained from the data-sharing principle, and (2) arrive at better-informed management decisions on primary production farms of field crops and secondary supply and processing chains, especially also on tactical and strategic issues. The project is co-funded by the Dutch Topsector Agri & Food.

Figure 1 shows a schematic overview of the data space of the farmer and the ecosystem connecting to the farmers' data. The figure shows an overview of connections between farmers in farmers' groups (green, left) and external connections with parties in the agri-food chain (right). In a transition to data-driven agriculture, the major bottleneck is bringing together all the data generated on a farm in an easy-to-use platform to convert the data into added value for the farmer. Underlying causes are a multitude of unstructured ICT tools and the lack of wide-supported architecture principles and standards in communication protocols in the data infrastructure.

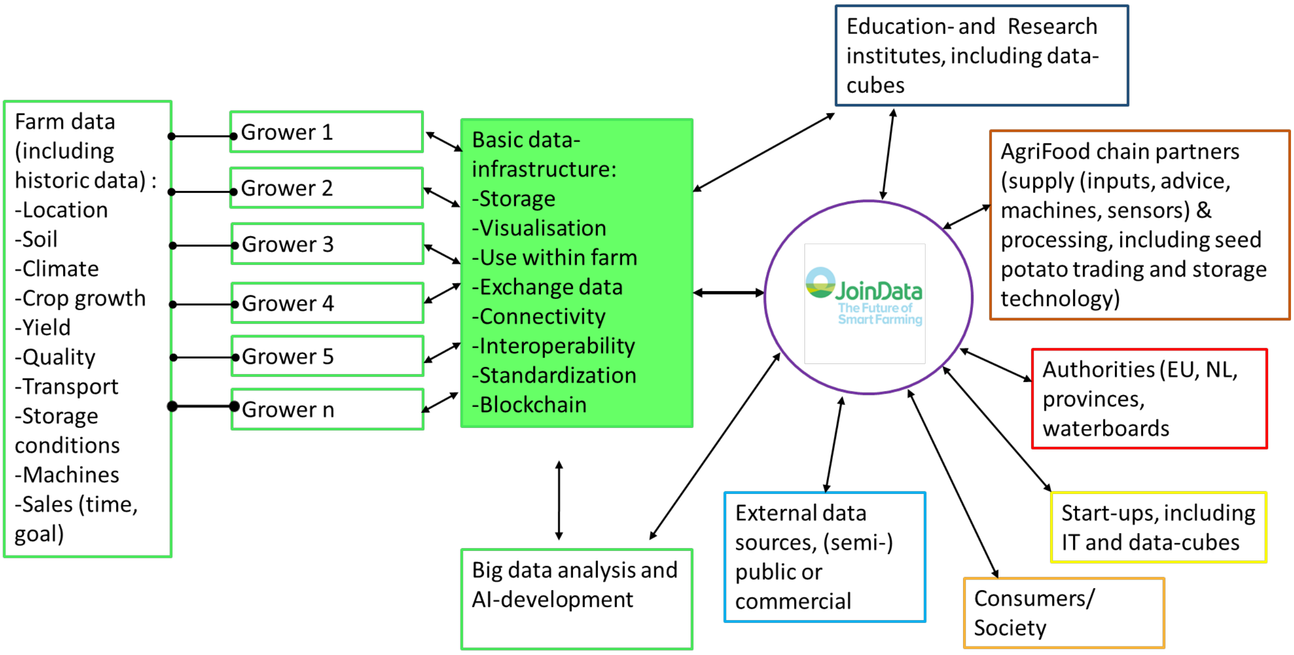


Figure 1 *Schematic overview data space PL 4.0.*

In the project the partners work on smart use of data in several use cases with a focus on developing architectural principles and achieving interoperability of data for field robots.

1.1 Scope and goal

Robot technology within the agricultural domain is making increasing developments in the last few years. Fresh players, such as start-ups, but also well-known machine manufacturers are required to cope with these developments and find a role for themselves. Recent technologies, such as computer vision, algorithms and robots bring many new possibilities. For example, computer vision promises to support users to understand and interpret the content of digital images and videos. This is done with computational methods, that use data coming from cameras and sensors[1]. Animal husbandry is a subsector within the agricultural domain dealing increasingly with computer vision, while other subsectors might potentially lack benefits, such as arable, dairy, fruit, and aquaculture.

Additionally, the need and use for standardization are becoming even more critical alongside the technological developments to achieve horizontal (cross-domain) and vertical (sub-sectoral) interoperability. Wageningen University & Research could play a significant role in this within the national and European context. However, the application of standards in data model development seems to be a non-easy task. This causes the lack of capabilities, capacity and efficient approaches to address the challenge of interoperable data and algorithms for images derived from agricultural robots. The case of autonomous weeding is selected to identify gaps in standardisation derived from functional requirements.

Following Bratton's Datastack, as mentioned in the feasibility study of PL4.0 (Kempenaar, et al. 2020), the main bottleneck in the adoption of field robots is the lack of ability to deploy in multiple contexts, meaning most current field robots focus on a few use cases or crops. The number of available algorithms is limited, due to the needed investments in time and costs to gather large data sets under a wide variety of conditions and in a wide range of crops. Exchanging image data could accelerate these developments, but a lack of data standardisations and communication protocols between farmers, robot service providers, robot manufacturers and algorithm providers hamper these developments. Also, the lack of infrastructure to process images in real-time with algorithms in the cloud (5G connectivity, GPU servers) and to store and exchange image datasets in a structured way are a bottleneck.

This report is an addition to an earlier study on the ecosystem of data exchange with agricultural robots for weed management (Booij, et al. 2022). The study aimed to identify the different actors for the use case of weed control with robots and map a preferred data ecosystem between those different actors. In the use case, the field equipment consists of a robot carrier, a camera (data acquisition system), an algorithm which detects crops and weeds on the images and a spot spray device which can spray individual plants. The actors are the farmer, the provider of the algorithm, the provider of the robot service and the manufacturer(s) of the equipment. Furthermore, there are different data sources like a FAIR Data Ecosystem itself, a Farm Management Information System, an OEM platform, the field equipment, additional weather data services, etc. In the study, the first step was made to identify necessary data streams between the actors, with a focus to provide enough context about image data and algorithms. Preferable context is for example where, how and under which circumstances images of crops and weeds are acquired. But also, which and how objects are annotated, which algorithm architecture is used, under which circumstances an algorithm is usable and ownership of data and algorithms. The study presented an initial list of definitions, data streams, business processes, description messages and a proposed architecture for federated data space. This data space should support principles according to Findable, Accessible, Interoperable and Reusable (FAIR) with published interoperable algorithms and annotated image data and support the workflow of image processing with neural networks.

The present study is a follow-up to this and aims to provide a basis for the standardization of data exchange of image datasets from agricultural field robots and a basis for the interoperability of algorithms. Therefore, we developed a sub-model containing multiple class models, considering rmAgro and preferred communication protocols like Robot Operating Systems (ROS) and ISOBUS. The focus is again on the use case of weed control. It is assumed that the autonomous weeding application should provide tangible results to generalize the results also to other autonomous applications.

[1] https://viso.ai/computer-vision/what-is-computer-vision/

Therefore, the starting point for this study is that the class model for the autonomous weeding application is merely one of the modules within rmAgro for the agricultural sector.

Reading guide:
Chapter 2 presents existing standards that are relevant for the use case of weed management by robots.
Chapter 3 describes the general approach of the study.
Chapter 4 describes the processes between the actors in detail and is modelled in Business Process Model and Notation (BPMN).
Chapter 5 presents a sub model containing multiple class models that are identified in the processes and existing standards. The class models are modelled in Unified Modelling Language (UML) and rmAgro is reused as much as possible.
Chapter 6 discusses the outcomes and gives recommendations for future work.

1.2 Relevant initiatives

In this paragraph, relevant work is presented in the domain of agriculture, standardisation, robotics, image data and algorithms. Since the primary focus of the domain is agriculture and the case involves robotic applications, the most relevant standards, such as ISO XML and ROS 2, are analysed thoroughly about computer vision and standards for agricultural machines to plan and execute tasks.

Within the IoF2020 project, an analysis is conducted on information models with their impact on algorithms and IoT (Internet of Things) (Cantera 2019). Different generic requirements derived from this study are listed below.

Table 1 *Generic requirements for information models supporting algorithms (adopted from Cantera 2019).*

Requirement	Description
Acting on confidence levels	When receiving predictions from an algorithm, a device may decide whether to act on them or not based on up-front knowledge of the quality of the algorithms or individual predictions.
Evaluation in the field	A device may store the predictions of an object and the actual measurement to facilitate model validation and evaluation of model performance.
Training and (local) retraining	A device may use a static, pre-trained model for daily operations, deployed as part of the solution. This solution may continue training on the job, so that it adapts to changes and local circumstances.

In the case of weed control, the algorithm YOLOv5 [2], is based on detecting objects in images. It predicts the classes of objects in the images with certain confidence levels. To train model, thousands of images with (hand)labelled objects are required to train, test, and validate the model. Moreover, the performance of a weed robot is not only validated by the accuracy of the model, but also the by accuracy of its application (removing weeds). We see a trend that a second sensor is installed behind the implement to gather evaluation data and processes it with another algorithm, increasing the number of images gathered by a single robot. It shows that image data can be used to develop new algorithms to improve existing algorithms by retraining them with local data and to validate the application and algorithm itself.

Furthermore, the project agROBOfood aims to build an ecosystem for the effective adoption of robotic technologies in the agricultural and food sector, while this should support the sector in becoming more effective and competitive [3]. One of the deliverables details the overview of the existing standardization landscape in agricultural and food robotics by considering six different areas of standardization for the analysis. These areas are communication protocols, robots, robot safety, food safety, security, and energy management. It is indicated that standardization in robotics is sparse in most of these areas and most of the standards derive from large agricultural machinery or industrial automation. Finally, consortium experts are identified on different standards who are available for support in interpreting and complying with these standards (Christoph Hellmann Santos 2020).

The Dutch government publishes base registries for crop fields [4]. The publication contains metadata including a reference ID to the dataset itself but also information on the publisher, data reuse, licences, and restrictions. The resource metadata set is also available in Extensible Markup Language (XML) / Resource Definition Framework (RDF), which makes it easier for computers to interpret data.
From an international perspective, collaborative effort is put into the “Data Sharing Coalition” which is an open and growing initiative to unlock the value of cross-domain data sharing (Data Sharing Coalition Expert Group 2021). In this work the authors present an overview of data standards, semantics, structure, and format of data that is to be exchanged. A small limitation of the study could be the distinguishment between semantics and structure on the one side and the format on the other side, since semantics and structure can

[2] https://github.com/ultralytics/YOLOv5
[3] https://agrobofood.eu/project/
[4] https://data.overheid.nl/dataset/10674-basisregistratie-gewaspercelen--brp-

be serialized following different formats. It is detected that there exist many standards among and within domains. A data service therefore must specify which standards are used in its Service Discovery (chapter 9.3.1) and requires a machine-readable way of specifying metadata (chapter 14). Moreover, the authors suggest a standard agnostic Trust Framework that will facilitate the possibility to harmonize the semantics of data standards across domains.

1.3 General Data Protection Regulation (GDPR)

The General Data Protection Regulation (GDPR) has been in effect for the European Union since 25 May 2018. GDPR gives protection to data subjects in case of misuse of personal data. In the table below, you see a summary of the GDPR requirements translated from table 6-1 from the feasibility study of PL4.0 (Kempenaar, et al. 2020).

Some, but not all farm data is personal data. However, in 2023, the EU (European Union) Data Act is likely to become effective, protecting the position of the enterprise where the data is generated.

The Netherlands and EU have translated GDPR and other regulations in Code of Conducts for agricultural data use. The CoCs state that third parties cannot use farm data unless they have consent (data sovereignty), digital services should be compliant with software standards in Agrifood (interoperability) and users of tools should be able to take their data to other providers (portability). This all means that the farm is in control of data generated on the farm. It is up to the farm owner with whom to share the data. These principles will become the bases in the future, and it is highly recommended to consider this in building new digital infrastructures.

Table 2 *GDPR requirements for handling personal data*[5]

The GDPR prescribes rules that data subjects can use to defend their rights if their personal data is misused. So this is specifically about personal data, which is directly identifiable.

- Informed consent

In order for personal data to be used, the person (the data subject) about whom that data is concerned must give consent. Consent is required for the processing of personal data, unless there is a legal obligation to share certain data.

- Right to access data

According to Article 15 of the GDPR, a data subject may have access to their personal data and the following information: the purposes for which the data are processed, the categories of personal data used, the people with whom that data has been shared, and how long the data will be stored

- Right to data portability.

Article 20 of the GDPR is closely related to the right to access data, but also differs from it. It states that data subjects have the right to receive personal data they have provided to a data controller, in a structured, common and machine-readable form. He also has the right to transfer that data to another data controller, without being interfered with by the controller to whom the personal data had been provided.

- Right to be forgotten (right to forget)

The right to be forgotten (Article 17) is the right to demand that personal data be deleted and prevent further dissemination. A person's personal data must be forgotten if:

- Personal data are no longer needed for the purposes for which they were originally collected or processed;
- The data subject withdraws his or her consent to the organization processing the data;
- If the data subject disagrees with the data processing organization;
- If the organization processing the data does not comply with the law;
- If the period in which the data was allowed to be stored has expired.

- Right to rectification

People have the right to have inaccurate personal data corrected, and incomplete personal data completed, if they request this in writing or verbally.

- Right to restriction of processing

People have the right to request a restriction on the processing of their personal data. This right is only applicable if the processing of data was unlawful, where personal data has been processed improperly or when people challenge the legal basis of the processing of the data

- Right to object to automatic decision-making, including profiling

People have this right if the decision-making has legal effects, or otherwise affects them. Therefore, data controllers, data subjects must provide short, transparent, clear and easily accessible information about how their personal data is processed. To prevent organizations from collecting more personal data than they actually need, data controllers must ensure that they comply with the "data minimization principle," and the requirements set forth by the principles of "purpose limitation" and "storage limitation.

- Right to refuse data processing

According to Article 21 of the GDPR, people have the right to refuse processing of their personal data.

[5] https://gdpr-info.eu/issues/personal-data/

2 Interoperable data and algorithms: a landscape of standards

An overview of relevant standards is presented in this chapter to give insights into reusable candidate concepts and classes. This understanding supports the respective process and data modelling phases as proposed in the following chapters of this paper.

In this study, the rmAgro standard is used to model the domain. It is a well-known normative standard in the agricultural domain and despite the rich explicit knowledge that resides in the model, there is more reason to adopt standards that are based on the Web 3.0 paradigm. There is a growing body of knowledge that recognizes the importance of Web 3.0 for publishing data on the web to make lives easier for humans and data understandable for machines (Hendler 2009; Lassila and Hendler 2007; Rudman and Bruwer 2016). These benefits could range from autonomous data integration (a world wide web data warehouse) to the enablement of AI technologies with the support of the Web Ontology Language (OWL) and its extension with Description Logic (DL) (Sirin, et al. 2007). Whereas Web 1.0 is characterized by static information retrieval, Web 2.0 is more interactive, and Web 3.0 supports the integration of data and applications with semantic web technologies, including the use of RDF and SPARQL Protocol and RDF Query Language (SPARQL). Despite promising advantages of the Web 3.0, there are disadvantages that should be considered, such as development of harmful scripts and languages, autonomous initiation of actions, unauthorised access, and manipulation of data.

This chapter is structured according to relevancy of the standards that were identified during the analysis. An overview of all standards is presented in Annex 2 Table of standards for data and algorithms. Commonly, it is found that there are useful standards that generically specify the exchange of interoperable robotics algorithms, such as Open Neural Network Exchange Format (ONNX), Robot Operating System (ROS2), SensorML and Common objects in Context (COCO). Alternatively, there are multiple initiatives that map existing standards that specify data objects for the agricultural domain, such as ISO standards. There are no specific standards for the agricultural domain that harmonise these both two domains, agricultural and robotics. In chapter 4 we present some proposals with harmonisation efforts. Interestingly, there is increasingly more standards that are based on the Web 3.0 paradigm, such as the Agricultural Information Model (AIM), Saref4Agri and FoodOn.

2.1 Reference Model Agro (rmAgro)

The rmAgro model is used in the Netherlands as a reference for the definition of data which is exchanged between parties in the agricultural sector. It is initiated and maintained by Wageningen University & Research by following as much as design principles in information technology and continuously adapting to new and changing functional requirements. The main purpose of the domain reference model part, rmAgro/drmAgro, is a clear description and definition of classes of objects recognised in agriculture and how they can be characterised. One of the ways to populate the domain model part is by mapping drmAgro to other models used for data exchange in agriculture like ISO11783 and UNCEFACT. The model is used as a reference when standard messages are formulated by the Dutch standardisation organisation for agriculture data, AgroConnect.

The reference model specifies the whole agricultural production domain, with a focus on crop production. This involves for example parties, fields, activities on the farm, data processing, crop recording, sampling and analyses, handling of products, machinery, etc. rmAgro comprises also specifications for the subdomaisns greenhouse production, animal husbandry and aqua culture (Goense 2021). The very first steps in the reference model dates already from 1984 when the Dutch government stimulated the development of information models to improve the use of information technology in the agriculture and horticulture (Aarts

1987a; Aarts 1987b; Goense 2017a; Goense 2017b). Following the UML language and Platform Independent Modelling (PIM) paradigm, the model is built with the reuse of knowledge of existing standards such as ISO 11783-10, AgroXML, Edaplos, Inspire and ADAPT. More recently, the reference model is elaborated as part of research activities on the domains of animal husbandry and plant protection products (Breemer 2021; Cantera 2019; Urdu, et al. 2022). Since most of the standards are focused on a specific data exchange process, rmAgro aims to cover a wider scope. For example, ISO11783-10 is focused in specific for data exchange between implements and Farm Management Information Systems (FMIS). With the introduction of tracking and tracing, IoT, cloud, etc., the lack of reference model for a wider scope is becoming more evident. Therefore, this study has selected the rmAgro to further model the domain of metadata standards for robot images.

2.2 Open Neural Network Exchange Format (ONNX)

The ONNX format aims to enable AI tools to work together by allowing them to share AI models[6]. The format is designed to represent any type of Machine Learning and Deep Learning model and therefore the need of interoperability is fulfilled to exchange algorithms and their parameters, assumingly also for weed detection. ONNX format is developed and supported by a community of partners, such as, Alibaba Group, Microsoft, AMD, Intel, etc., who have adopted the format in their frameworks and tools. Although a generic specification of the ONNX could not be found in a generic format such as UML, XML, RDF, there are many scripts for different platforms that are accessible through the GitHub ecosystem of repositories[7]. An appropriate example is the Proto buffers. For the YOLO algorithm, as it is commonly used for agricultural robots' imagery and object detection, further analysis is needed to indicate the effort that is needed to make the algorithm ONNX compliant.

2.3 Robot Operating System (ROS)

In the domain of robotics, ROS is a well-known software development kit that supports building robot applications. The standard is currently published as open source and maintained by Open Robotics[8]. Although ROS contains protocols and communication messages, it is lacking on recommendations from a domain modelling perspective. However, the implementation of ROS 2 contains an updated version of the standard that comprises the concept of the so-called Data Distribution Service (DDS) for allowing messages to be structured.

DDS is a middleware protocol and standard for API in which data connectivity has a significant role. By integrating the components of a system together, it aims to improve connectivity of low-latency data, reliability, and architectural scalability for IoT applications[9]. For the proposed data space, this could imply that the DDS function between the weed robot (operating system) and FMIS (application). The mechanism to share date between these nodes (either human or machines) are elaborated in the Data-CentricPublish-Subscribe (DCPS). The DCPS[10] defines the functionality used by applications which needs to publish and subscribe the values of data objects. The Object Management Group (OMG) develops enterprise integrations standards. An example is the Interface Definition Language (OMG IDL), which is a descriptive language used to define data types and interfaces. The specification provides a description into two sub parts: a Platform Independent Model (PIM) and a Platform Specific Model (PSM). In Figure 2 a conceptual model is presented for the DCPS.

[6] https://onnx.ai/get-started.html
[7] https://github.com/onnx/onnx/tree/main/onnx
[8] https://www.openrobotics.org/
[9] https://www.dds-foundation.org/what-is-dds-3/
[10] https://www.omg.org/spec/DDS/1.4/PDF

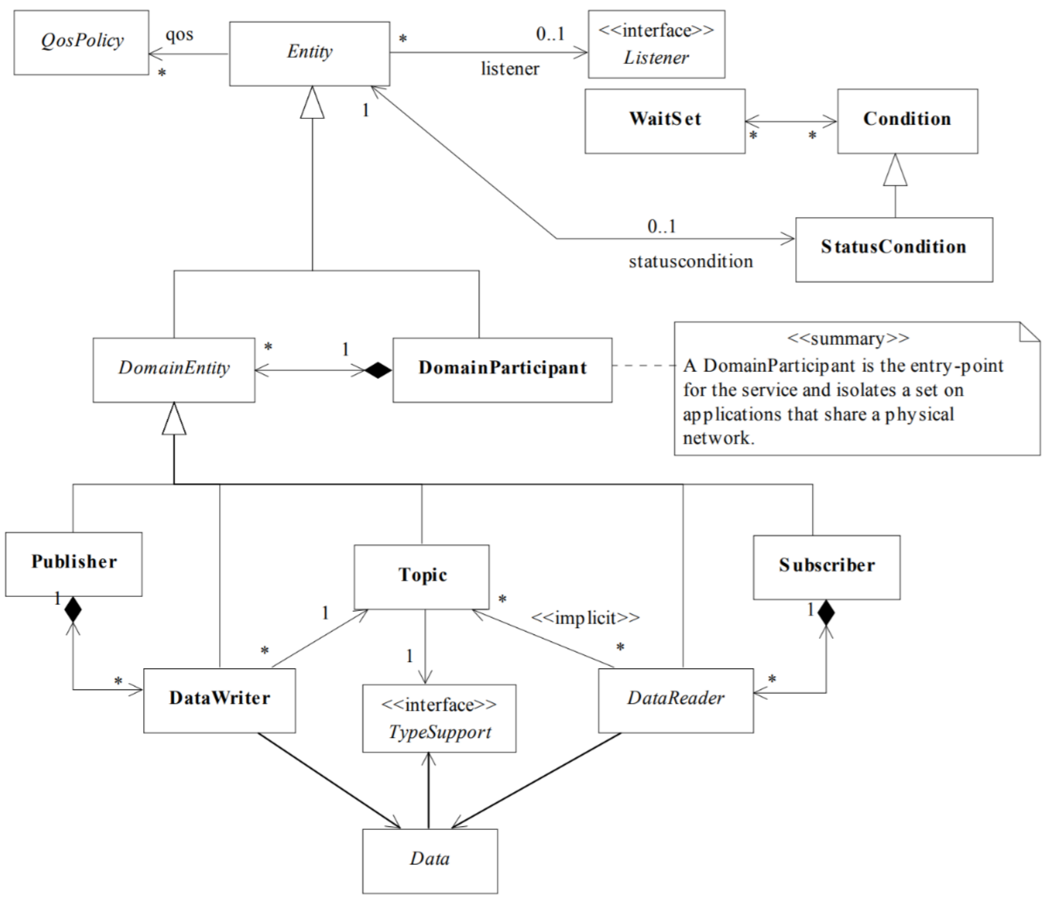


Figure 2 *Conceptual Data Centric Publish and Subscribe adopted from OMG DDS standard.*

In this study we found many resources for the use of DDS in ROS[11] and possibilities for common and custom messages [12] [13] [14]. Summarized, DDS is a middleware that provide relevant features to the ROS system, such as distributed discovery and possibilities to control options for transportation of Quality of Service (QoS)[15] [16] [17]. Knowledge on diverse types of implementations can be found elsewhere [18] [19] [20] [21].
Recent developments show the possibilities to enrich the existing ROS standard with a multi-layer localisation and mapping procedure for agricultural sites, adding semantic maps as part of the VineSLAM (dos Santos, et al. 2016). This initiative proposes a certain ROS structure with definition of messages[22], such as vision messages[23] and sensor messages [24].

2.4 Common Objects in Context (COCO)

The COCO dataset format is a large-scale object detection, segmentation, and captioning dataset with the goal to advance state-of-the-art object recognition by broadening the context of image recognition to scene

[11] https://design.ros2.org/articles/ros_on_dds.html
[12] https://docs.ros.org/en/foxy/Tutorials/Custom-ROS2-Interfaces.html
[13] https://roboticsbackend.com/ros2-create-custom-message/
[14] https://github.com/ros2/common_interfaces
[15] https://docs.ros.org/en/ros2_documentation/galactic/Concepts/About-Different-Middleware-Vendors.html
[16] https://www.dds-foundation.org/omg-dds-standard/
[17] https://www.omg.org/spec/DDS/
[18] https://opendds.org/
[19] https://cyclonedds.io/index.html
[20] https://zenoh.io/
[21] https://www.eprosima.com/
[22] https://gitlab.inesctec.pt/agrob/vineslam_stack/vineslam/-/blob/master/docs/interfaces.md
[23] http://docs.ros.org/en/api/vision_msgs/html/msg/Detection2DArray.html
[24] http://docs.ros.org/en/melodic/api/sensor_msgs/html/msg/PointCloud2.html

understanding (Lin, et al. 2014). The large dataset is publicly available to use for training of algorithms and contain richly annotated images that are translate from complex everyday scenes.
The dataset has a well described format which can be considered as basis for a standard to exchange images and annotated images. The format covers: info, images, annotations, categories, and licenses. Info describes the information of a dataset, which contains the images and the annotations for the images. Image describes the main characteristics of an image, and under which license it is made available. An annotation describes objects distinguished within an image and belongs to a particular category. The COCO dataset includes a list of 90 categories and is seen as fundamental for metadata for images. A class model of the COCO dataset format is given in Figure 3.

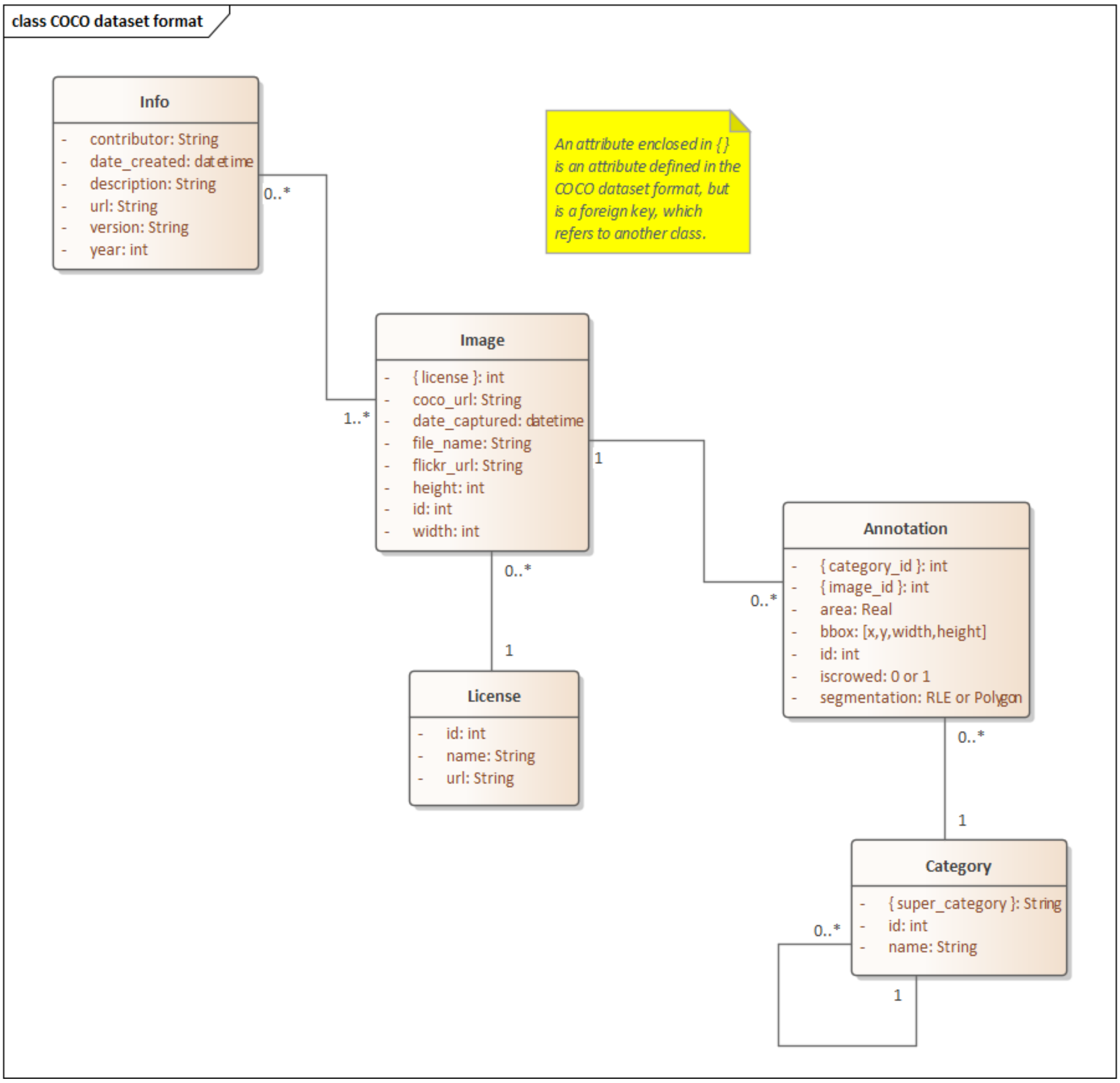


Figure 3 *Class model of the COCO dataset format.*

2.5 Minimal Information About a Plant Phenotyping Experiment (MIAPPE)

Looking from the plant phenotyping domain, the Minimal Information About a Plant Phenotyping Experiment (MIAPPE) has developed a data model to allow conducting field experiments[25]. It describes the minimal information needed to give context for a field experiment. The data model describes classes, attributes and

[25] https://www.miappe.org/

coding lists that are relevant to specify conditions for learning and testing images during the execution of field operations. In the same line of direction, the Netherlands Plant Eco-phenotyping Centre (NPEC) emphasize data harmonization with a pilot in the EMPHASIS project [26]. The data model provides context for researchers to interpret the data from the experiment. Some classes, attributes and coding lists could be useful for our study; however, the scope of our study is on data from farming practice and to give context of image data and algorithms to the developers, providers, and users of algorithms. Furthermore, our data-model should be more generic to cover other agricultural domains potentially also.

2.6 Data Expression, Exchange, and Processing in Smart Agriculture IEEE standard P2992

More recently, attention has focused on the provision of recommended practices for designing smart agriculture data, specifically on the data format, data tags, their naming rules, and data transfers to propose unified practices of data sharing, processing, and expression in the IEEE standard P2992[27]. Limitations of this development may be the unclarity of addressing the control of data for robots, collection of images, classification of objects in those images and description and development of algorithms. Moreover, it is still imprecise what the exact relation is of this standard with other agricultural standards.

2.7 International Organization for Standards (ISO) based standards

2.7.1 ISOXML / ISO 11873 - 10

The well-known standard ISOXML is often used as the expression for ISO11783 part 10, which includes the use of DDI's defined in part 11. ISO11783 stands for "Tractors and machinery for agriculture and forestry—Serial control and communications data network". It covers all communication layers for data exchange on farm machinery using the CAN communication protocol, and the communication between task controllers and FMIS by means of XML files.

The use of robots in agriculture can be seen as a specific type of agricultural machinery and that makes it relevant to consider ISO XML as a standard for data exchange between FMIS and robots. However, ISO11783-10&11 does not cover images and as it is still based on the use of XML files, it is also not suited for real time communication that is necessary for alarms generated by a robot. ISO11783-10&11 also have characteristics inherited from the technical limitations that were present during its development in the 1980s of the last centuries.

The Agricultural Industry Electronic Foundation (AEF) takes the initiative for additions to ISO11783 which will allow for real time communication, and these developments should be monitored.

ISO11783-10 describes the underlying data model as an entity relationship diagram which is further specified as elements of XML schemas. This data model covers parts of the agricultural domain relevant for the execution of farm work through mobile machinery.

2.7.2 Infrastructure for Spatial Information in Europe (INSPIRE)

The INSPIRE Directive addresses 34 spatial data themes with an impact on the environment and need for environmental applications. The Directive seeks to create a European Union Spatial Data Infrastructure which should (1) enable public sector organizations to share environmental spatial information among each other, (2) facilitate public access to spatial information and (3) support policy making[28].

---

[26] https://emphasis.plant-phenotyping.eu/services/emphasis-pilots/harmonisation-pilot#:~:text=The%20harmonisation%20pilot%20service%20will,of-the-art%20procedures

[27] https://standards.ieee.org/ieee/2992/10614/

[28] https://inspire.ec.europa.eu/about-inspire/563

This encoding of the INSPIRE metadata in this technical specification is based on the ISO Standards ISO 19115[29][30], ISO 19119[31][32] and ISO 19139[33]. The abstract standards 19115 and 19119 provide a structural model and specify the content of the set of metadata elements used in this specification, but they do not specify the encodings of those elements. The ISO 19139 specifies an XML encoding of ISO 19115 elements, but not for the service-specific metadata elements contained in ISO 19119. To provide an XML encoding also for the INSPIRE service metadata, XML Schemas implementing the ISO 19119 model have been published by the Open Geospatial Consortium (OGC). These XML Schemas, though not officially endorsed by ISO, are widely used within the metadata community, and have been chosen to be used also in INSPIRE since version 1.0 of this specification.

2.7.3 Training Samples Markup Language (TSML)

For geospatial and remote sensing data, the diversity of formats makes it difficult to share data. This is especially the case when there is a desire to design advanced applications, such as knowledge discovery, pattern recognition, data analysis and data integration. The TSML effort proposes a structure based on XML to store training data sets for specifically supervised classification algorithms (Soares, et al. 2011). The main advantage is the ability to share examples among classifiers from different applications to analyse and compare results. This characteristic aligns with the initial goal of this study.

[29] https://www.iso.org/standard/53798.html
[30] https://inspire.ec.europa.eu/documents/Metadata/INSPIRE_MD_IR_and_ISO_v1_2_20100616.pdf
[31] https://www.iso.org/standard/59221.html
[32] https://docs.geostandaarden.nl/md/mdprofiel-iso19119/
[33] https://www.iso.org/standard/67253.html

2.8 Landscape of standards

The most important standards are presented in previous paragraph, while there are several other standards that specifies the domain of agricultural robot vision from different perspectives. As seen in Figure 4, all these standards are graphically illustrated into five Venn-diagrams. Each diagram represents a specific topic and consists of the standards that are found within this study, respectively, generic standards, machine communications, agricultural domain, standards based on Web 3.0 and vision & algorithms standards. The standards that are part of the diagram "Generic standards" are further sliced into three main standardization organisations, namely ISO, OGC and GS1. There might be also other relevant standardisation organisations, such as UNCEFACT, that are not part of the scope of this landscape. These generic standards are usually applied in rmAgro. For example, the Geography Markup Language (GML) is used to specify fields and field boundaries that is usually derived from an XML-based dataset.

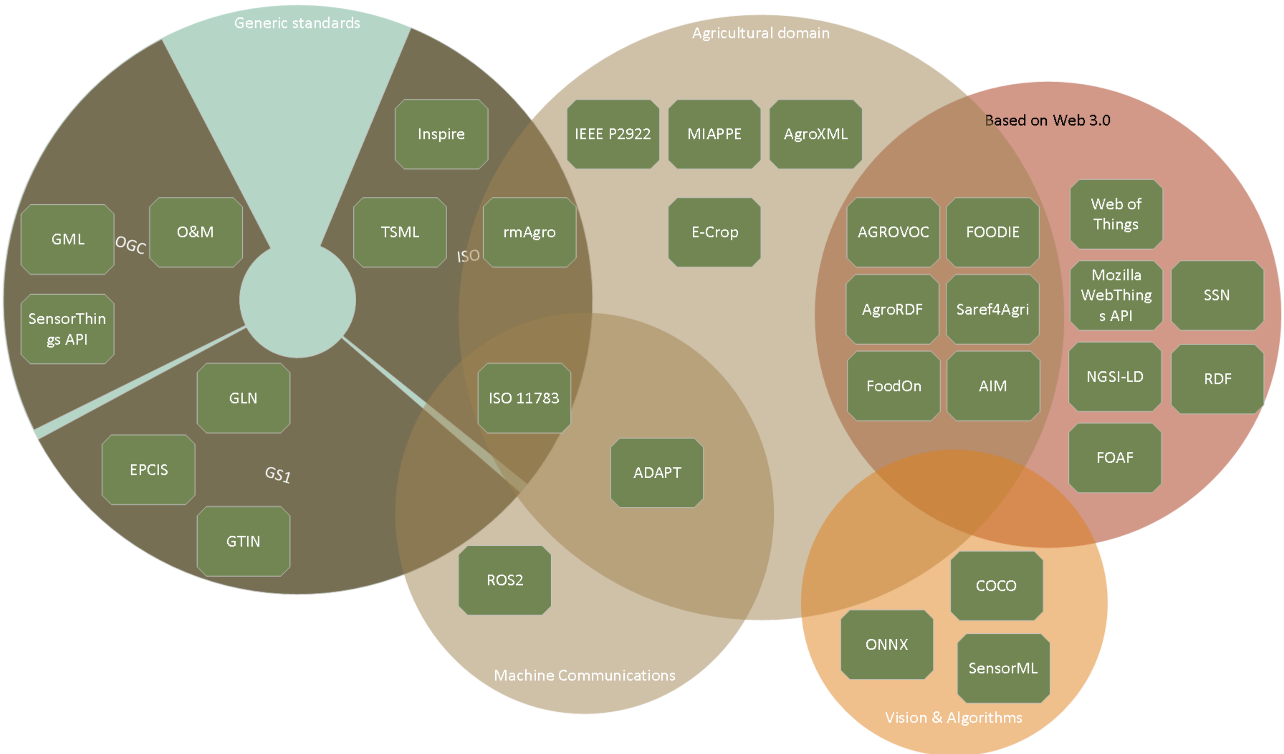


Figure 4 *Landscape of standards for interoperable robot vision data. The standards are potted in diagrams, which represent a specific domain of interest as starting point.*

3 Methodology

3.1 Use case Weeding Robot: FAIR Data Ecosystem – Published Interoperable Algorithms and Data

As mentioned, the previous study of Booij, et al. (2022) presented an initial list of definitions, data streams, business processes, description messages and the proposed architecture for a federated data space that should support FAIR principles. The aim to publish interoperable algorithms and annotated image data required this study to develop a class model with the definitions of the reference model rmAgro. Such a class model can be used to generate specification and sample code for different communication protocols like XML or JSON messages. After a thorough desk and literature study on existing standards, the followings steps were taken carefully:

1. Describe different business processes in the data space of weed management with robots and the development of algorithms from annotated images, using Business Process Model and Notation (BPMN);
2. Develop Unified Modelling Language (UML) based class diagrams for the different message flows and data associations that are presented in the business process models;
3. Use as much as possible already existing classes from the domain reference model Agro (drmAgro) and extend it with data elements identified in the BPMN message flows and data associations not already present in drmAgro;
4. Provide an example of XML code for one of the message flows.

A BPMN model describes the executions of process thinking and activities from a customer's perspective. The process modelling paradigm is especially useful to visualise the different actors in a use case and their relationship and communication between them. BPMN makes a clear difference between message flows and data associations. Message flows is the data exchanged between partners, represented by a pool in the BPMN diagram. Following the FAIR based architecture, they need a common semantic and format, a standard, to make interchangeability possible. The data associations represent internal data communication of the partners and do not, in principle, need no standard. Though a standard can be extremely helpful, a clear example is the ISO11783 standard for data exchange between implements of a partner. The data associations are in this study used to identify data that is required to realize the business process.
The UML Class Diagram is a graphical notation used to visualize object-oriented systems and which describe the structure of a system using the systems classes, their attributes, operations, and the relationships among objects. The classes identified for a message flow are for a common semantic for communication between partners through a FAIRDataEcosystem. The drmAgro reference model is, as described below, a platform independent data model. From this independent data model, platform specific models can be generated which have a common semantic, but different format like for example XML, JSON, or RDF.
For this study, a sub model "RobotVision" is generated which holds only those classes, attributes and associations which are relevant for robot and vision-based weeding and development of the algorithms for weed detection. Throughout the document, as a modelling convention, camelCase is used to denote certain model objects, such as processes, messages, classes, attributes, or class diagrams.

3.2 Processes to elicit relevant messages and data flows

As mentioned in the introduction, the process models in this study should support the FAIRDataEcosystem. The system described in the BPMN's does not store the data itself. The service part of the eco system is responsible for authentication of the partners and it is the platform where can be specified who has access to which data, the authorization. These two sub processes are shown as one activity "check_Authorisation" in chapter 4 Business processes. There is the possibility to add more resources in terms of CPU and data storage in case there are partners and users that have this wish.

The FAIRDataEcosystem could also have an APP_Store functionality, where applications, software, algorithms can be placed and downloaded, and which also keeps track of the financial aspects of the use of those apps. This can also include licenses for data services like for example weather data.
Several details required for secure data transfer between organizations, are not specified in the BPMN's and the messages themselves. Message transfer is for now specified in the BPMN's as a push by the platform, while a mechanism to inform a receiving partner where data is available for download and providing that partner with a token which can be checked by the sending partner.
The most important aspect of the FAIRDataEcosystem as facilitator for interoperability is that it specifies the data semantics and the data formats for well-defined messages.

3.3 Schema generation and reuse of standards

3.3.1 Sub model RobotImages

The first step is selecting the classes, attributes and associations which are relevant for the domain of robot vision. This is done by using the Schema Generator in Enterprise Architect. This is described in **rmAgroDocumentation.docx[34],** chapter 5.

3.3.2 Platform specific models

The second step is creating a platform specific model. This is described in chapter 6 of the documentation of rmAgro. DrmAgro itself is subdivided in packages as described in chapter 3.6.1 of the documentation. When the schema generator is used to create the sub model, all classes, datatypes, and enumerations are placed in one single package. On itself that is not a problem for the use case specific model, but as the intention is to use as much as possible existing standards, like for example GML and UNCEFACT, those classes and datatypes which are specified by those standards should be placed in a package representing these standards, with their appropriate name spaces. This becomes relevant when generating (xml) schemas.

3.3.3 XML schema generation

As explained in the documentation of rmAgro, chapter 7, it is required for GML based geometries to make a GML sub package with target namespace and prefix gml [35].
There is no need to generate the schemas for those standards, but by giving those packages, the sub model specific schema will include and refer to the correct standard schemas and namespaces.
In this study on robot images only an XML model is generated from the sub model RobotVision, which can be found on the website of AgroConnect[36].

3.3.4 Standards to model the domain

The model rmAgro is a model used in the Netherland as a reference for the definition of data which is exchanged between parties in the agricultural sector. It is initiated and maintained by Wageningen University by following as much as design principles in information technology and continuously adapting to new and changing functional requirements. The main purpose of the platform independent domain data model part, rmAgro/drmAgro, is a clear description and definition of classes of objects recognised in agriculture and how they can be characterised. One of the ways to populate the domain model part is by mapping rmAgro/drmAgro to other models used for data exchange in agriculture like ISO11783 and UNCEFACT. rmAgro is used as a reference when standard messages are formulated by the Dutch standardisation organisation for agriculture data, AgroConnect.

[34] https://www.agroconnect.nl/Portals/10/documenten/rmAgro/rmAgroDocumentation_oct_2020.docx
[35] http://www.opengis.net/gml/3.2
[36] https://www.agroconnect.nl/Portals/10/documenten/RobotImages/rmAgro_RobotImages_XMLandXSDs.zip

3.3.1 Domain reference model Agro (drmAgro)

There are a number of standards for data exchange in Agriculture and domains used in agriculture. A characteristic of most of those standards is that they are defined in a technology which was most common at the time of development. A clear example is ISO11783-10 based on an XML schema. Another characteristic of most of those standards is that they cover a limited domain within agriculture.

The reference model rmAgro/drmAgro tries to cover all use cases encountered by Wageningen UR in its own projects and in projects with partners like those from AgroConnect and the EU. It follows naming conventions and other design principles established at its start and adapted during further development.

The consequence is that expressions used in other standards are not always followed. The change in technology over the last decades led to the decision to make drmAgro a platform independent model. As it covers all encountered use cases it has the advantage that standards based on this model can reuse components developed for different use cases. A disadvantage is that covering many use cases leads to more complex data structures then is required when dealing with only one use-case or a limited domain. Another disadvantage is that the reference model is not hindered by upward compatibility and can therefore deviate from already existing messages. The reference model can be seen as a model like it ideally should be, as far as it ever can, and used as a basis when new versions of standard messages are developed.

4 Business processes

The business processes for this study are modelled with BPMN language. One business process is already modelled as a BPMN in Booij, et al. (2022) and includes the development and improvement of algorithms; the first mentioned process as presented in this chapter.
One of the basic principles of a business process is that an organisation can initiate and interfere with business processes of other organizations. The development or improvement of algorithms could be initiated by an algorithm service provider, while the use of algorithms is part of executing operations which are initiated by farmers. As the initiative comes from two different organizations, we should see them as two different business processes.

There are four different business processes to distinguish in respect of robot images:

1. The *development and improvement of algorithms*;
2. *Ordering Robot weed control*. The selection of an appropriate algorithm for vision-based applications and ordering an operation. In this use case selecting weed detection and ordering weed control;
3. *Robot Weed Control Execution*. The use of an algorithm during the execution of field operations;
4. *Retrain an Algorithm from Images made during Weeding*.

The processes are published as part of this study in HTML and are publicly accessible in a more readable and comprehensive way [37]. The transfer of data between organizations, and sometimes between entities within an organisation such as a ManMachineSystem and a Farm- or ContractorManagementInformationSystem, is following the proposed architecture, controlled by a service part of the FAIRDataEcosystem. This is not shown in the BPMN train_CollectedImages and in the other BPMN's this is left away for some of the messages. The reason is that focus is on the data content of the messages and specifying the mechanisms of this communication architecture will result in complex BPMN's. See also the architectural representation of the data ecosystem in Booij, et al. (2022).

4.1 Process "Developing and improving algorithms"

An algorithm service company can decide at a certain stage to develop or improve an algorithm to detect weeds or diseases in crops, which are based on images made in the field. This requires that one or more fields are selected and agreed on by the farmer to collect images and to classify objects in those images as identified weeds or diseases. It is assumed in this BPMN that the latter will be done manually in the field by a person. An alternative for easy to recognize objects this can be done manually at a desk on hand of the images themselves. This scenario is described in the BPMN train_CollectedImages and an HTML resource is accessible through an URL[38].
During image collection relevant properties of field, crop, weeds, weather, and eventual other properties that can influence the image should be captured. Also, relevant properties of the camera system must be known. When classified images are available a "learning" process can be started to train algorithms, in most cases NeuralNetwork's, with their parameters. A trained algorithm will be tested and validated with independent sub datasets. When trained algorithms have sufficient accuracy, they can be published for use by others.

1. The business process is started by an algorithm service provider when there is a need for a new or improved algorithm. The service provider will collect a list of ParticipatingFarms and chooses a farm for a request. (It is now specified such that the algorithm service provider has such a list of participating farms, but such a list could also be available in the FAIRDataEcosystem).
2. An investigation is made for suitable fields for which a request is made to the farm. This request implies the request for field data.

[37] https://www.agroconnect.nl/Portals/10/EnterpriseArchitect/BPMNRobotImages/index.html
[38] https://www.agroconnect.nl/Portals/10/EnterpriseArchitect/BPMNRobotImages/EARoot/EA7.html

    a. The request “TrainingFieldRequest” contains at least the PlantSpecies and period of planting or a GrowthStage. It might also contain a list of “required” WeedSpecies.
3. The farmer accepts or rejects the request by a TrainingFieldResponse message and in case of acceptance sends required field data.
    a. TrainingFieldData will contain at least: The CropField with its identifier, Designator of the field and its PlotSurface, such that area and location are known. It will specify the planting Operation such that planting date, row distance and plant distance in the row is known. When available, also quantitative data on weeds, pests and/or diseases can be provided. Eventually also the driving pattern during planting can be provided, such that a robot that will be used for image collection can generate its driving pattern.
4. The algorithm server provider evaluates TrainingFieldData and when suited, this will be stored. In case a request is rejected, or TrainingFieldData is not suited, a new request will be done.
5. A next process is started when the RobotServiceProvider will start image collection. This could be the farmer, contractor, other party, or the algorithm service provider itself. It will transfer the Georeferenced, AnnotatedImages, and Conditions to the algorithm service provider. Conditions include FieldConditions and EnvironmentalConditions. Eventually this can be done through the platform of the device manufacturer and the FAIRDataEcosystem, which is not specified in this BPMN. It is assumed here that collecting images for training purposes is an activity without a weeding or crop protection operation. Using images collected during such operations is described in the BPMN retrain_WeedingImages.
6. When arrived in the field, the ManMachineSystem will start image collection. Based on the FieldSurface (and eventually the planting pattern) a Zone will be selected for image collection.
7. An image will be taken in the selected zone and based on the PositionAndOrientation of the device and CameraPositionAndOrientation of the camera on the device, the image will be georeferenced and stored as GeoReferencedImage.
8. During image collection also relevant conditions like EnvironmentalConditions, CropConditions and FieldConditions can be collected and stored. Environmental conditions will include weather conditions like solar radiation, cloudiness, and wind speed. FieldConditions will, as far as not already part of the TrainingFieldData, include soil surface wetness, visible objects like shelves, stones, and leaves on the soil. CropConditions will include canopy wetness and growth stage when the latter differs from the data provided by the farmer or differs in the field.
9. The zone will be inspected for weeds, diseases or pests and its specific species will be classified. In the Image, the classified objects will be indicated either by drafting polygons (like bounding boxes), lines, or points. This can be either done by drafting the relevant objects on the image manually and classify them, but it is also possible that an algorithm is used to identify objects, which still must be checked. This results in an Image with an Annotation an Image is a result of an Observation which can also result in one or more PropertyValue’s describing other characteristics observed during that observation. These PropertyValues describe the EnvironmentalConditions, CropConditions and FieldConditions. Observation is part of an Operation which refers to the CropField on which it is carried out.
10. When the number of required images is reached, the collected images, Annotations, georeference data and Conditions will be send as Georeferenced, AnnotatedImages, AssociatedData and these will be stored by the service provider.
11. When sufficient images are available for training, a selection (subset) will be made from the available classified images that will be used for training. Another independent subset will be reserved for test or validation.
12. The ClassifiedImages will be synchronized with OtherData, which includes the FieldConditions, CropConditions and the EnvironmentalConditions. Furthermore, specifications of the used camera and other sensors can be attached. This results in TrainingImagesWithData
13. An algorithm for weed/disease/pest recognition will be selected and this will be trained on the selected ClassifiedImages. The result is the AlgorithmAndParameters.
14. When training is finished, the algorithm with the derived parameters will be validated on the images selected for validation, which will be ValidatingImagesWithData.
15. When validation is successful, the AlgorithmPlusSpecification will be published.

4.2 Process “Ordering robot weed control with algorithms”

This business process starts when there is a need for weed (or disease) control and assumes that a choice is made on the forehand for robotic weeding. When modelling this BPMN, the assumption is made that there is a platform (FAIRDataEcosystem) available which manages data exchange between organisations, eventually a data store and an app store can forward requests to service providers. [39]

1. The BPMN starts when there is a need for weeds control.
2. The farmer asks for weather data by a WeatherDataRequest to a platform.
3. The platform checks whether the farmer has a data license for weather data, DataLicenses, and when that is the case, the farmer forwards the request to the data provider.
4. The data provider collects the requested weather data and sends it as a WeatherDataResponse.
5. The farmer asks for available algorithms by an AlgorithmsRequest and gets a AlgorithmsResponse with available algorithms. Based on WeatherData, CropData and FieldData, suited Algorithms are selected.
6. With an AvailableRobotServicesRequest is investigated who is able to deliver robot services that uses the selected Algorithms. The response, AvailableRobotServicesResponse**,** is a list of robot services providing the suited Algorithms.
7. One of the RobotServices is selected and a ServiceOrder is placed at the service part of the FAIRDataEcosystem. After a check whether the farmer is authorised to place orders, this is forwarded to the RobotServiceProvider.
8. When a OrderReject message is received, a new robot service will be selected, and a new order will be placed. When there is an OrderResponse which indicates acceptance. This AcceptedOrder will be stored as a planned operation.

Execution of the placed, and accepted order is described in the BPMN execute_RobotWeeding.

4.3 Process “Executing robot weed control”

This BPMN model describes the execution of weed control with a robot by a robot service provider[40]. There is already an order placed for the weed control operation and as soon as the operation is planned for a particular farm, the farm is informed. It is assumed that the robot is travelled to the field by means of another vehicle. This requires that during travel time other equipment and men are available. After finishing the operation, the farm is informed.

1. At regular time intervals, one or two times a day, a request for weather data is placed by the robot service provider, e.g., a contractor operating weeding robots on behalf of farmers which is the message WeatherDataRequest. This request is placed at the FAIRDataEcosystem, which checks whether the contractor is authorised to request for weather data.
2. The weather service provider collects the weather data from a forecast for the requested location.
3. When the WeatherDataResponse is received by the robot service provider all planned Operations are collected as well as all Resources (men and machinery) with their status (available or not) during periods. With this data Jobs and Tasks are scheduled.
4. The PlannedJobsAndTasks are distributed to the machinery of the service provider. In this case, that means one Task to the ManMachineSystem which is responsible for the travel and another Task for the weeding to the robot itself. As both Tasks depend on each other they form together a Job. The ManMachineSystem responsible for travel knows by the information in the Job which robot to travel. Data exchange is managed by the FAIRDataEcosystem. Also, the farmer is informed of the planned Task for the weeding operation. Jobs and Tasks for the farm machinery (Robot and tractor pulling a trailer) are send to the original end manufacturer’s, OEM, platform, which takes care that the data is send to the machinery itself.

[39] https://www.agroconnect.nl/Portals/10/EnterpriseArchitect/BPMNRobotImages/EARoot/EA3.html
[40] https://www.agroconnect.nl/Portals/10/EnterpriseArchitect/BPMNRobotImages/EARoot/EA1.html

5. At the time a Job starts, the ManMachineSystem collects the data of the Job and the TaskData that belongs to that Job, called JobAndTaskData. The ManMachineSystem for travel is proposed as the JobController.
6. When arrived on the field, the robot will be unloaded and the RobotTaskData collected to start three main activities: navigation, weeding and real time evaluation of the weeding.
7. For weeding the AlgorithmAndParameters must be loaded. A camera takes a picture, the AlgorithmAndParameters detects the objects on the picture, then the ObjectPosition on the image is translated to real world positions and an actuator is controlled to remove or kill te weed.
8. To evaluate if the system of detection and actuator has done its job, a second camera and another AlgorithmAndParameters is used which generate WeedingEvaluationData .
9. The WeedingEvaluationData will also be used to control the weeding process. When the evaluation data don't pass the WeedingCriteria, which must be part of the specification of the Operation, a WeedingAlarm is generated and send to the server providers home base by using the robot OEM platform and the FAIRDataEcosystem. When the WeedingAlarm is received at the home base, it will be evaluated and when seen as appropriate, a message to abort the task, called TaskAbortMsg, will be send to the robot. When this message is received, the task will be stopped.
10. The task will also be stopped when the navigation function of the robot has covered the whole field surface. A message will be sent by the robot to the ManMachineSystem (MMS) responsible for travel of the robot and the TaskData will be send to the RobotServiceProvider.
11. The travel MMS will travel with the robot to the next field or to the home base and send also TaskData to the robot server provider.
12. When the OEM platform receives TaskData from the machinery it will inform the contractor (or farmer) that new Taskdata is available. The information management system, IMS, of the contractor can collect this TaskData from the OEM platform and send the relevant data of the executed (or aborted) Task to the farmer.

4.4 Process "Retraining algorithm from images during weeding"

This business process starts with the start of a robot weeding operation shown in the lowest pool in the BPMN model[41]. It stops after publishing a new algorithm with parameters on the FAIRDataEcosystem.
This Business process is not complete. The focus is on the data flows which are relevant for training algorithms from images made during a weeding operation, so some details shown in other BPMN's are not shown in this one.
This business process shows two pools FAIR_DataEcosystem and FAIR_DataEcosystem2. They represent the same pool, but it is split in two to prevent too much message lines crossing other pools.

1. The BPMN starts with the start of the weeding operation in the field. The navigation function and the weeding function are specified here. (For the real time evaluation function see the BPMN execute_RobotWeeding).
2. During object detection with the chosen algorithm from the images, also a ProbabilityPercentage is calculated and stored.
3. At the end of the Task, which is described more in detail in BPMN execute_RobotWeeding, the home base will be informed that the Task is ready and new TaskData is available. This TaskData also includes the images itself and all associated data which includes the ProbabilityPercentage, proposed as TaskWithImages&AssociatedData.
4. The Task, with images and its associated data, will be collected by the RobotServiceProvider.
5. After evaluation of the data a warning will be sent to the provider of the Algorithm when the probability percentages during object detection are (too) low.

[41] https://www.agroconnect.nl/Portals/10/EnterpriseArchitect/BPMNRobotImages/EARoot/EA5.html

5 Messages and dataflows

The business processes show which physical parts and actors in a federated data space for weed detection interact with each other. The dataflows between the parts and actors could be defined in messages. Many elements to build up these messages are available in the reference model rmAgro. However, the domain of vision techniques and deep learning with neural networks is new and not described in rmAgro yet. This chapter focusses on the exchange of algorithms and image datasets following from the business processes and provides a mapping from the minimal list of metadata described in Booij, et al. (2022) to rmAgro. Most of the class diagrams described in this chapter are published and accessible by AgroConnect [42].

5.1 Algorithms

5.1.1 Algorithms and parameters

Algorithm is an existing class in drmAgro and was introduced for crop growth simulation models. It is proved to be useful for model interpolation of spatial data based on parameters derived from semi-variograms. Another example on the use of the algorithm class is for sensors to calculate for example weight from voltage level. An algorithm has one or more variables as input and one or more variables as output. Voltage level is, in the example of the sensor, the input, weight is the output. A neural network, which is often used for object detection from images, can also be seen as an algorithm.

In this use case convolutional neural networks are used to detect objects (weeds) on images. Much of the existing literature on neural network plays particular attention to object detection and image recognition (Bianco, et al. 2018; Ren and Wang 2022) . A common way to exchange these deep learning models is with the use of the standard ONNX [43]. This standard is a serialized representation of the model in a protocol buffer and is designed to allow framework interoperability between various tool stacks like PyTorch, CNTK, MXNet, Caffe2, TensorFlow, CoreML, etc.

The model description of a Neural Network is incorporated in the class model AlgorithmAndParameters[44] and is based on the description of ONNX. A neural network is an algorithm with fitted parameters and its specification in respect of requirements, validity and where it is derived from. Some limitations of the adoption of the ONNX standard could be the ambiguity of the term's parameters, that is interchangeably used when variables are intended.
A neural network is a special type of algorithm which is described as a graph with nodes. The Weights and Bias Parameters in a NeuralNetwork are fitted during the learning process. These ParameterValues form a ParameterSet, which is a Resource. To exchange these parameters an URL could be used, defined in ParameterSet ➔ Resource.ResourceLocator.
The same applies for an URL reference to a Github repository of a neural network architecture Algorithm ➔ Resource.ResourceLocator.

The accuracy is an output of the training and validation of a neural network. During a ParameterFit, which includes the learning of a NeuralNetwork, various categories of error estimates can be calculated[45]. ParameterFit.CrosEntropyError is the most appropriate for classification algorithms.
An Algorithm can also be validated. ClassificationValidation gives several errors which can be specified.
For the case of weeding robot, input variables for a neural network are the pixel values in an image and other characteristics like physical soil variables. Output is the predicted plant species and its location.

42 https://www.agroconnect.nl/Portals/10/EnterpriseArchitect/rmAgro_SubModelRobotImages/index.html
43 https://github.com/onnx/onnx/blob/main/docs/IR.md
44 https://www.agroconnect.nl/Portals/10/EnterpriseArchitect/rmAgro_SubModelRobotImages/EARoot/EA1.html
45 https://www.neuraldesigner.com/learning/tutorials/testing-analysis

Objects, in this case weeds, should be labelled and specified by their BotanicalName. When Weeds are only detected as weed, and not on species level, then a reference to PlantGroup should be made.
Based on the class diagram of Algorithm and Parameters, in Table 3 an overview is presented of the minimum metadata that should be included with the exchange of an algorithm.

Table 3 *Minimal metadata fields that should specify an algorithm.*

What	class rmAgro	attributes	Example
Title of algorithm	Algorithm	AlgorithmDesignator (String)	*Volunteer potato detection in sugar beet*
ID of algorithm	Algorithm Resource	AlgorithmIdentifier (IdentifierType) ResourceIdentifier (IdentifierType)	global identifier global identifier
Description of algorithm	Algorithm	AlgorithmDescription (String)	*The algorithm detects volunteer potatoes in sugar beets on clay and peat soils in the North-East Netherlands.*
Version of algorithm	Algorithm	Version (String)	*v5.2*
Classification categories	NeuralNetwork ➔ PropertyVariable Plant ➔ PlantSpecies	ValueEnumerator BotanicalName	0, 1 Solanum tuberosum, Beta vulgaris Altissima Group
DatasetID's used for training	Dataset	DataSetIdentifier (IdentifierType)	global identifier
Name architecture	NeuralNetwork	Architecture (String)	*YOLOv3*
URL of model	Algorithm ➔ Resource	ResourceLocatorURL (anyURL)	*https://github.com/ultralytics/YOLOv5*
URL of weights and biases	ParameterSet ➔ Resource	ResourceLocatorURL (anyURL)	
Settings algorithm	ObjectDetectionAlgorithm	ConfidenceThreshold (Real) MaximumObjects (Integer) MaximumObjectSize (Integer) MinimumObjectSize (Integer)	
Accuracy	Algorithm ➔ ParameterFit ➔ *ClassificationValidation*	 CrosEntropyError (Real) ClassificationAccuracy (Real) ConfusionMatrixValues (integer) ErrorRate (Real) ...	
Ownership	Party	Designator (String PartyIdentifier (IdentifierType) ...	*Name company* *KVK nr*
Licensing	License	LicenseDesignator (String) LicenseIdentifier (Identifiertype) LicenseURL: (anyURL) ValidFrom (Datetype) ValidTo (Datetype)	

5.1.2 Licensing and ownership of resources

As presented in Table 3, during the exchange of algorithms ownership and licensing of digital resources become important. The published class diagram[46] shows a class diagram that represents this importance. Based on the Inspire standard, it is decided that DataSet and Algorithm are a subclass of Resource. Resource has a responsible party as defined by Inspire and a PartyRole. To eliminate redundancy there is only an association to PartyRole, which on its turn is associated with the Party.

5.1.3 Specifications of an algorithm

The algorithm architecture plus its weights makes the algorithm useful. However, the trained neural network will only be valid for certain conditions under which the training data are collected. For the case of weed detection in crops, it will only be valid for certain plant species occurring in the dataset. Crops and weeds can look different (phenotype) between fields and regions due to variation in field and crop characteristics, like row width, plant distances, and environmental conditions. Furthermore, the camera setup, such as type of camera and lens, angle and field of view will also determine how objects are projected in the images. So, to exchange suitable fitted (trained) algorithms for a farmer's local situation and available equipment, these kinds of metadata should be known.

An algorithm is trained on a dataset with training data, which is compiled from images derived from one or more original datasets. It is assumed that an original dataset of images is gathered on one specific field and its conditions on crop, field, and environment. To have a robust algorithm which works under a wide variation of conditions, the training data should contain a balanced set of images derived under a wide variation of conditions, which is often the case with existing algorithms. The specifications of an algorithm should therefore include a range for conditions, preferably a list per variable, rather than a specific condition which is preferable for image datasets. In rmAgro the conditions are specified for the ParameterFit by the associations "is_valid +from" and "is_valid +until" to one or more PropertyValues, as can be seen in the class diagram TrainedNeuralNetwork[47]. It is assumed that environmental conditions, characteristics on field and crop and process variables, such as row width and plant distances, can be covered by PropertyValue's of PropertyVariable's.

Some objects like PlantSpecies, which is specified by an association to Crop, can have two roles in the context of Neural Networks.

- *One is that it is a characteristic of a (trained) Neural Network. A user who selects a Neural Network wants to know for which Crops, and under which conditions it is appropriate to use. This can be specified by associating an Algorithm, or its subclass NeuralNetwork to appropriate classes representing these specifications, like in in the class diagram TrainedNeuralNetwork[47].*
- *A second role is that as input or output variable (input or output parameter by ONNX) for a Neural Network. In that case the specifications should be a PropertyValue of a PropertyVariable as seen in in the class diagram AlgorithmAndParameters[44].*

The consequence of these two roles is that there might be some redundancy in the reference model drmAgro by using classes with attributes for some cases and by using members of code lists for PropertyVariables in other cases. This suggest to use a CropVariable code list, which includes CropSpecies and WeedSpecies. WeedSpecies will be an output PropertyVariable for robot weeding. Both, CropSpecies and WeedSpecies have a CodedValue, which is in a Botanical code list.

The phenotype of plants or weeds could look different from year to year depending on environmental conditions like the weather, climate, and soil properties. But also, the perception of objects on images can be influenced by the weather conditions due to changing light conditions (cloudy vs bright sky). Algorithms are

[46] https://www.agroconnect.nl/Portals/10/EnterpriseArchitect/rmAgro_SubModelRobotImages/EARoot/EA33.html
[47] https://www.agroconnect.nl/Portals/10/EnterpriseArchitect/rmAgro_SubModelRobotImages/EARoot/EA43.html

ideally trained and published with image datasets gathered in divers environmental conditions to work robustly. Therefore, it is assumed that only specifying the region and soil types is sufficient. Soil type could be modelled as a PropertyVariable as can be seen in the class diagram CropFieldSpecification[48]. The assumption for soil types is that there is a national coding list for soil types. The PropertyValue has in that case a CodedValue. The PropertyVariableCode includes an ListIdentifier for the coding list which is used.

As stated, before the algorithm is assumed to be valid for a range of crop, field, and environmental conditions. Besides metadata describing the algorithm itself also metadata about these conditions should be included, which is presented in Table 4. In the column example sometimes a list of examples is given rather than one specific example as an algorithm could be used in more than one condition.

Table 4 *Minimal metadata included in specifications algorithm.*

What	class rmAgro	attributes	Example
Crop	Crop	CropDesignator (String) CropIdentifier (IdentifierType)	*[Sugar beet, Onion]*
	PlantSpecies	BotanicalName (IdentifierType) PlantSpeciesDesignator (String)	*[Beta vulgaris, Allium cepa]*
Variety	Variety	VarietyDesignator (String) VarietyIdentifier (IdentifierType)	*[BTS115N, Jewel, Queena KWS, ...; Hybound, Sharon, Red Baron, ...]*
Weeds	PlantSpecies	BotanicalName (IdentifierType) PlantSpeciesDesignator (String)	*[Solanum tuberosum, Rumex obtusifolius, ...]* *[Potato, broad-leaved dock, ...]*
Region	Crop → Region → Polygon	Boundary (SurfaceBoundary)	*Polygon ((6.418889 52.985346, 6.415462 52.641085, 6.995595 52.644569, 6.947328 52.990952, 6.418889 52.985346))*
Soil type	Cropfield → PropertyZone → PropertyVariable → PropertyValue	 SoilPhysicalVariable QuantityType SoilPhysicalVariable QuantityType SoilPhysicalVariable QuantityType SoilPhysicalVariable QuantityType	 *ClayFraction* *2%* *LoamFraction* *7%* *SandFraction* *83%* *OrganicMatterContent* *9.2%*
Suitable for which Machine(s) / implement(s)?	Implement	ImplementDesignator (String) ImplementIdentifier (IdentifierType)	*BBleap Spotsprayer*
Suitable for which data-acquisition system(s)	Component	Designator (String)	*BBleap LeapEye*

*The assumption is that NeuralNetworks for robot weeding are developed for a Crop with a particular **PlantSpecies**. This can be specified by an association to **Crop**, which has an association to **PlantSpecies**. Eventually other aspects of a Crop like **ProductionPurpose** can be specified also.*

An alternative for specifying the soil type with a national coding list is the specification of sand, loam and clay fraction as PropertyValue of SoilPhysicalVariable's.

*It is assumed that an Algorithm is trained on datasets acquired by a specific data acquisition system, consisting of a **Camera**, Lens and **ElectronicControlUnit** and therefore only suitable for the same type of system. These parts could be seen as **Components** of an **Implement** and is further detailed in paragraph 5.1.3.1. Furthermore the orientation of the data acquisition system on the Implement is of importance and further specified in paragraph 5.1.3.2.*

[48] https://www.agroconnect.nl/Portals/10/EnterpriseArchitect/rmAgro_SubModelRobotImages/EARoot/EA19.html

5.1.3.1 Used equipment

Algorithms are suitable for specific Data Acquisition Systems and Equipment. Data Acquisition Systems consists of a Camera, Lens and ElectronicControlUnit. All these parts could also be seen as Components of an Implement, as can be seen in the class diagram UsedEquipment[49].
If applicable, CameraLight is also one of the components of a data acquisition system. So, whether the Robot (Implement) uses artificial lightning, it possible to specify this with CameraLight.
The class diagram describes the equipment which is used during execution of a Task and the relevant components of that equipment. A robot for weed control is an Implement which belongs to an ImplementAssembly. This construct is used to model assembled implements which always work together and are seen as one unit of operation. An example of the latter is a drill permanently mounted on a rotary harrow. But in the case of robot weeding the Assembly can consists of a robot platform, a Data Acquisition System and an actuator removing the weeds (e.g., a spot sprayer or hoeing device).

For algorithm providers and robot service providers it is necessary to know the details of the data acquisition system, like the type and settings of the camera('s), and the used equipment to know if the algorithm will work. Table 5 shows the minimum metadata about the data acquisition system which should be included.

Table 5 *Minimum metadata that specifies the image data acquisition system.*

What	class rmAgro	attributes	Example
Specs camera light	CameraLight	Lightmodel (IdentifierType) LightCategory (CameraLightCategoryEnumeration) ColorTemperature (Integer) Lumen (Integer) Shielded (Boolean) LightWavelength (Integer) LightMaterial PowerRequirement (Real)	
Camera	Camera ➔ Component ➔ Equipment ➔ EquipmentIdentifier	Designator (String) SerialNumber (String) Model ModelYear (Int) Series PartNumber (String)	*Stereolabs Zed 2i Stereo Camera* *SerialNumber* *Zed 2i Stereo* *2022* *ZED* *ZED 2I (or EAN 0096718605293?)* *Integrated*
	Lens ➔ Component ➔ Equipment ➔ EquipmentIdentifier	Designator (String) SerialNumber (String)	*N.A.*
Specs Camera	Camera	Height (Int) Width (Int) NumberOfBands (Integer) SpectralRange (SpectralRangeType)	*1080* *3840* *3* *RGB*
	Lens	MinimumPhocalLength MaximumPhocalLength	*2.1mm* *2.1mm*

[49] https://www.agroconnect.nl/Portals/10/EnterpriseArchitect/rmAgro_SubModelRobotImages/EARoot/EA49.html

A data acquisition system is conform ISO11783 an ElectronicControlUnit, which is in drmAgro a subclass of Component.

drmAgro uses the term Equipment for machinery. Equipment has an EquipmentIdentifier for use in management systems. Equipment, Implements and Components are more in detail identified by EquipmentIdentification, as can be seen in the class diagram CameraIdentification[50]. Equipment can be a stationary Installation, a Tractor, an ImplementAssembly, etc. ImplemenstAssembly exists of one or more Implements. A clear example is a planter combined with a rotary harrow. This is seen as one piece of equipment, but existing of two implements which perform different OperationTechniques for different CulturalPractises

SensorML has an extensive description of Camera, by using the Community Sensor Model, CSM, which is developed for remote sensing purposes. A simpler description is given in ROS2, as presented in chapter "Related Work". For now, only the Height and Width of the sensor array are used in the drmAgro specification.

5.1.3.2 Camera position and orientation

The position and orientation of the camera determines how objects are perceived on images and therefore essential information when using algorithms for computer vision. The published class model [51] shows how this is modelled in rmAgro.
The camera has a position and orientation in the EngineeringCoordinateSystem of the Equipment it is mounted on. Sensors can also be mounted on movable sections of a boom on the equipment, for example a movable spraying boom. In that case, the camera has a position in the EngineeringCoordinateSystem of the section, which has on its turn a position and rotation in the engineering coordinate system of the equipment itself. The equipment has a position in a (world) coordinate reference system. It might require several transformations to determine the position and rotation of the camera in the (world) coordinate reference system. It is suggested to give the ActivityField, where the operation is performed, an engineering coordinate reference system itself to improve performance for passing the position of weeds to the actuator for removal.

The attributes of Camera in the SensorML show lack of normalization to correct the image distortion. These should be attributed to the Lens, modelled in rmAgro as a separate class. Furthermore, platform and timing of images should be modelled as separate classes. A more extensive modelling of Sensor is available in ISO19130 Geographic Information – Imagery sensor models for geopositioning, consisting of three parts[52].

Table 6 *Minimum metadata required for camera settings.*

What	class rmAgro	attributes
Camera position	Point in EngineeringCRS Equipment	Position (DirectPositionType)
Camera orientation	EulerRotation in EngineeringCRS Equipment	Alpha (Real) Beta (Real) Gamma (Real)

A decision must be made whether Roll, Pitch and Jaw should be used or the Euler angles Alpha Beta and Gamma. For now the Euler angles are used in drmAgro, as they are also applicable for ISO11783.

A **Camera** *as being a* **Sensor** *has a* **Position** *and an* **EulerRotation** *in an* **Implement***'s* **EngineeringCRS.** *The* **Implement** *itself has also a* **Position** *and an* **EulerRotation** *but now in a* **CoordinateReferenceSystem.** *From this specification the distance to the object can be calculated, but also other data required to georeference the image on the soil surface.*

[50] https://www.agroconnect.nl/Portals/10/EnterpriseArchitect/rmAgro_SubModelRobotImages/EARoot/EA11.html
[51] https://www.agroconnect.nl/Portals/10/EnterpriseArchitect/rmAgro_SubModelRobotImages/EARoot/EA13.html
[52] https://committee.iso.org/sites/tc211/home/projects/projects---complete-list/iso-19130-1.html

5.2 Image training datasets

Algorithms are trained on image datasets that consists of images and annotations of objects in those images. For AlgorithmProviders the context of how and where those images are acquired is of importance for adequate selection of the right datasets for development of new algorithms or retraining existing algorithms. Therefore, the specifications of the used equipment, crop, field, and environmental conditions are important. Furthermore, an AlgorithmProvider could also provide services like weed maps to farmers to support decision making. Therefore, the images should be georeferenced.

This chapter describes the structure of message flows containing images, annotations, and its associated data. The class Image is part of the class Dataset, which can be seen in the published diagram[53]. Dataset itself results in an instantiation of the class Resource**.**

Some metadata is extremely specific for each image (such as CameraSetting, cameraID during image capturing), whereas other metadata is applicable to the complete set of images in a Dataset (like the used equipment). The metadata specific for each image is described in paragraph 5.2.1, the metadata that concerns the annotations of images is presented in paragraph 5.2.2 and the information applicable to the dataset is described in paragraph 5.2.3.

5.2.1 Images

The class model Image[53] is modelled by reusing the following standards: TSML, ROS2, and OpenCV. TSML is considered as it is up to now the only model description which specifies classified images. ROS2 is a logical choice as in our applications a combination is made of image taking with the use of robots. OpenCV is not considered yet, as the documentation is difficult.

As can be seen from the class model an image is an Observation of a ObservationSurface measured at a ObservationTime. Each image is made with a camera with a specific camera setting to compensate for e.g., light conditions, as can be seen in the published diagram CameraSettings[54].

Intrinsic and extrinsic parameters are transformation matrices that convert points from one coordinate system to the other[55]. The Datum describes the projection which is used and is modelled in GML3.2. More information on datums can be found on the web as published by OpenGIS[56].
Intrinsic parameters are internal parameters of the camera and lens, also known as camera to image and image to pixel transformation, whereas extrinsic parameters are parameters describing the positions and orientations of the camera and sensors, also known as the world to camera transformation, which measures positions on the vehicle coordinate system and translated to a world coordinate system. It is assumed that during an agricultural task the intrinsic and extrinsic parameters remains the same and could therefore be part as Associated Data as described in 5.2.3.

For the intrinsic parameters there are several coordinate systems specified for images, and sometimes different expressions are used for the same coordinate system. We could not identify a standard identifier list of image coordinate systems. In paragraph 5.2.4 we elaborate more on this topic and give an overview of several image coordinate systems. Our proposal is to follow OGC’s structure as much as possible specifying geometries.

To measure the position often GNSS (Global Navigation Satellite System) receivers or wheel encoders are used. The measured GNSS positions or encoder values are ‘unique’ to each image and should therefore be included with each image.

[53] https://www.agroconnect.nl/Portals/10/EnterpriseArchitect/rmAgro_SubModelRobotImages/EARoot/EA31.html
[54] https://www.agroconnect.nl/Portals/10/EnterpriseArchitect/rmAgro_SubModelRobotImages/EARoot/EA15.html
[55] https://towardsdatascience.com/what-are-intrinsic-and-extrinsic-camera-parameters-in-computer-vision-7071b72fb8ec
[56] http://schemas.opengis.net/gml/3.2.1/datums.xsd

In rmAgro a GNSS receiver or encoder is a Sensor which measures the PropertyValue as value of the PropertyVariable. This proposes to use VariableCodeList for GNSS, as can be seen in the published diagram GNSS_Receiver[57].
If encoders are used instead of GNSS receivers to describe the real-world position of images, the value and datetime of acquisition are important. Those PropertyValues can have a TimeInterval. By the optional attributes of TimeInterval it is also possible to specify only one time moment. The ProcessVariable identifier list should contain a variable which could be called PulseFrequency, which is of the RateType.

Table 7 *Minimal metadata included with images.*

What	Class rmAgro	Attributes	Example
Imagefile	Image	ImageFileFormat (ImageFileFormatEnumeration) ImageFileDesignator (String) ImageIdentifier (IdentifierType) Height (Int) Width (Int) ImageURL (AnyURL)	*PNG* *1080* *3840*
Date and time of acquisition	Observation	ObservationTime (DateTimeType)	
CameraID	Component	ComponentIdentifier (SerialNr)	
Camera settings	Image ➔ CameraSetting	Shutterspeed (Real) Aperture (Real) ISO (Integer) FocalLength (Real) WhiteBalance (integer)	
GNSS position & heading	GNSS_Variable ➔PropertyVariable	PositionStatus PositionNorth (will be in decimal degrees) PositionEast (will be in decimal degrees) PositionUp Heading	
(Optional) Encoder datetime and value	Sensor ➔ PropertyValue ➔ TimeIntervalType ➔ PropertyVariable ➔ PropertyValue	 Duration (DurationType) StartTime (DateTimeType) Status (Status Enumeration) StopTime (DateTimeType) PulseFrequency (RateType) MaximumRate MinimumRate PulseFrequency	
License	License	LicenseIdentifier (Identifiertype)	

It is assumed that in most field applications the camera lens has a fixed focal length (or manually adjusted), as to keep the intrinsic parameters the same. FocalLength could then be included as AssociatedData. However sometimes equipment also has autofocus properties which adjusts the FocalLength for each image taken. Then FocalLength should be included as metadata in each image.

In drmAgro there was a class Position, which is deprecated in actual GML specifications. This has to be replaced by Point in a number of diagrams. There is in drmAgro still a class GNSS_Position, which holds a number of GNSS_variable's, which is equivalent to an entity in ISO11783. This GNSS_Position is a subclass of Point. The proposal is to deprecate the class GNSS_Position.

ObservationTime is of DateTimeType. The proposed format like GMT format depends on the exchange format chosen (XML, JSON, etc)

The ProcessVariable identifier list for Encoders should contain a variable which could be called PulseFrequency, which is of the RateType. This would require additional attributes MaximumRate and MinimumRate to Propertyvariable.

[57] https://www.agroconnect.nl/Portals/10/EnterpriseArchitect/rmAgro_SubModelRobotImages/EARoot/EA29.html

5.2.2 Annotations of images

The published class model[58] shows the class model of AnnotatedImages, which is an Image in which objects are identified (boxed, segmented) as Annotations. Annotations refer to certain ImageSegments in the Image which are spatially described either as ImageRaster or as ImageVector. The expression Image is used here to indicate that it is a raster, or a vector described by the pixels of the image.
A deep learning algorithm converts an image to a numeric matrix in n dimensions. For example, a coloured image of 640x640 pixels will be converted to a 3D matrix with 3 layers (red, green, blue values between 0-255) x 640x640. This matrix is then processed along the neural network and results in an output which is a 2D matrix of identified objects with the classification and positions of pixels or objects. Different expressions are used to define the positions of objects[59,60] on the images. There is not a standard list of image coordinate systems used. In Annex 4 we elaborate more on image coordinate systems.
The Annotation is classified in a class hierarchy. In COCO dataset format the classifications are ordered in main categories (supercategories) and sub-categories. In our use case for weed detection this will be a PlantGroup respectively a PlantSpecies.
An ImageRaster and ImageVector could be specified in more detail using the TSML standard, as described in overview standards in chapter 2.7.3.

Table 8 *Minimum metadata provided with annotations.*

What	class rmAgro	attributes	Example
ID	Annotation	AnnotationIdentifier (IdentifierType)	450
ImageID	Image	ImageIdentifier (IdentifierType)	
Classification categories	PlantGroup PlantSpecies	GroupDesignator (String) TaxonomicClass (Enumeration) BotanicName (IdentifierType) PlantSpeciesDesignator (String	Plants Solanum tuberosum, Beta vulgaris Altissima Group, ...
CategoryID	Annotation	AnnotationIdentifier (IdentifierType)	(0, 1,...)
Method	Annotation ➔ ImageSegment	AnnotationMethod (CodeType) ImageSegmentEnumeration (Enumeration)	Segmentation Polygon
Segmentation	Annotation ➔ ImageSegment ➔ ImageVector	SegmentationArea (Real) IsCrowded (Boolean) ImageSegmentEnumeration (Enumeration) ImageBoundingBox (x,y,width, height)	600.4 1 BoundingBox [473.05, 395.45, 38.65, 28.92]

5.2.3 Associated image data and annotations

The message AssociatedData, as identified in the process models, specifies the data associated to images which are annotated and is relevant for an algorithm developer to select appropriate datasets for a training/learning process of algorithms. This kind of metadata is part of the class Dataset rather than part of each individual instantiation of the class Image. In a typical business processes a dataset can be generated after each task, in that case the dataset represents data gathered on one field. However, a new dataset could also be compiled from images from different datasets, so that it entangles data gathered on several fields in a region under different circumstances. In the description below a dataset for one field is described.

AssociatedData can include:

- General info about dataset as described under 5.2.3.1.
- Data from the farm management information system to identify the farm, field, crop, and purpose of the cultivation as described under 5.2.3.2.
- Crop conditions which are described under 5.2.3.3.
- Field Conditions which are described under 5.2.3.4.
- Environmental conditions which are described under 5.2.3.5.
- Specification of other sensors like GNSS receiver or encoders under 5.2.3.6.

[58] https://www.agroconnect.nl/Portals/10/EnterpriseArchitect/rmAgro_SubModelRobotImages/EARoot/EA9.html
[59] https://albumentations.ai/docs/getting_started/bounding_boxes_augmentation/
[60] https://albumentations.ai/docs/getting_started/keypoints_augmentation/

- Specification of the used camera(s), lens(es) and lightning. This is like Used equipment as described in paragraph 5.1.3.1.
- The position and orientation of the camera(s). This is like Camera position and orientation as described in paragraph 5.1.3.2.

A data acquisition system is conform ISO11783 an ElectronicControlUnit, which is in drmAgro a subclass of Component.

drmAgro uses the term Equipment for machinery. Equipment has an EquipmentIdentifier for use in management systems. Equipment, Implements and Components are more in detail identified by EquipmentIdentification, as can be seen in the class diagram CameraIdentification. Equipment can be a stationary Installation, a Tractor, an ImplementAssembly, etc. ImplemenstAssembly exists of one or more Implements. A clear example is a planter combined with a rotary harrow. This is seen as one piece of equipment, but existing of two implements which perform different OperationTechniques for different CulturalPractises

SensorML has an extensive description of Camera, by using the Community Sensor Model, CSM, which is developed for remote sensing purposes. A simpler description is given in ROS2, as presented in chapter "Related Work" . For now only the Height and Width of the sensor array are used in the drmAgro specification.

5.2.3.1 General metadata

Table 9 shows which general info should be included with a dataset.

Table 9 *Minimum metadata of a dataset with images and annotations*

What	class rmAgro	attributes	Example
Title	Dataset	DataSetDesignator	*VolunteerPotatoField1*
Description	Dataset	Description	*Images of volunteer potato in sugar beet derived under rainy conditions on field [position] with equipment Y...*
URL	Dataset ➔ Resource	ResourceLocatorURL (anyURL)	
Year	Dataset	BeginDateTime EndDateTime CreationDateTime	
Version	Dataset, DataProcess, DataAggregation	Version (String) DataSetIdentifier Data	*v5.2*
Geometric Extent	Dataset ➔ Resource ➔ from ➔ Region	 RegionCode (CodeType) RegionDesignator (String) Boundary (SurfaceBoundary)	 *MultiPolygon (((6.932773 52.876038, 6.926804 52. 880459, 6.933228 52.876261, 6.932773 52.876038)))*
Contributors	Dataset ➔ Resource ➔ PartyRole ➔ Party	 Role (PartyRoleEnumeration) Designator (String) (Third)PartyIdentifier (IdentifierType) ...	 *Resource_Provider* *KVK nr*
Owner	Party	Designator (String PartyIdentifier (IdentifierType) ...	*Name company* *KVK nr*
Licensing	License	LicenseDesignator (String) LicenseIdentifier (Identifiertype) LicenseURL: (anyURL) ValidFrom (Datetype) ValidTo (Datetype)	
Classification categories	Plant ➔ PlantSpecies	BotanicalName	*Solanum tuberosum, Beta vulgaris*

It is questionable whether DataSet's can have a version. They are generated once, either by direct Observation, or by a DataProcess which can process DataSet's and create new ones. These are new DataSet's with a new Identifier. The source is always tracable by providing the DataProcess information, or the information of sensors and equipment which originated them.

The geometric extent of the dataset could be one field, but also several fields in a certain region. Both can be specified using the class Boundary.

Based on Inspire it is decided that DataSet and Algorithm are a subclass of Resource. Resource has a responsible party as defined by Inspire and a PartyRole. To eliminate redundancy there is only an association to PartyRole, which on its turn is associated with the Party. See paragraph 5.1.2 Licencing and Ownership.

5.2.3.2 Info from FMIS

From the business processes as elaborated in Chapter 4.1, it can be derived that AlgorithmProviders do a TrainingFieldRequest to acquire a dataset for developing or improving algorithms. The class diagram TrainingFieldResponse[61] shows the specifications of information that can be found in a Farm Management Information System.
The diagram TrainingFieldResponse should contain the CropField with the required Crop, PropertyValues describing the soil type by PhysicalSoilVariables, but should also specify the Operation for the CulturalPractice which has the code for planting. For the operation, the planting time is given by AbsoluteTiming and the values for the ProcessVariable's that have the codes for row distance and plant distance in the row, or plant density. Table 10 shows the minimum metadata preferred for including with a dataset.

Table 10 *Minimum metadata from FMIS preferred for including with a dataset.*

What	class rmAgro	attributes	Example
FarmID	Party ← Organization ← Farm	PartIdentifier	*AGRONL09098104FRMAVCRPV1995794*
FieldID	Cropfield	PlotIdentifier	*bc37a8bd0253478db4ede4b019842420*
Crop	Crop PlantSpecies	CropDesignator (String) CropIdentifier (IdentifierType) BotanicalName (IdentifierType) PlantSpeciesDesignator (String)	*Sugar beet* *1010201* *Beta vulgaris*
Variety	Variety	VarietyDesignator (String) VarietyIdentifier (IdentifierType)	*BTS 6740* *20231*
Cultivation Purpose	CropField → Crop → CropProductionPurpose	CropProductionPurposeDesignator (String)	*Sugar*
Operation Type	OperationTechnique	OperationTechniqueDesignator (String)	*Weeding*

5.2.3.3 Crop conditions

In respect of the crop, the growth stage can be specified, but also Leaf Area Index (LAI), height of the canopy, pest, diseases, etc, are of importance. These are PropertyVariables, as these conditions might vary within a Field. They are specified for PropertyZone's within a CropField or ActivityField while for the growth stage the BBCH index is proposed (Jki 2010). Table 11 shows the metadata about the crop conditions, while Table 12 shows a list with suggested variables for the crop. The published diagram that specifies crop field and crop conditions can be found via the following reference[62].

[61] https://www.agroconnect.nl/Portals/10/EnterpriseArchitect/rmAgro_SubModelRobotImages/EARoot/EA47.html
[62] https://www.agroconnect.nl/Portals/10/EnterpriseArchitect/rmAgro_SubModelRobotImages/EARoot/EA17.html

Table 11 *Metadata describing crop conditions.*

What	Class rmAgro	Attributes	Example
Area / region	**Crop → Region → Polygon**	Boundary (SurfaceBoundary)	*Polygon ((6.418889 52.985346, ..., 6.418889 52.985346))*
Crop conditions	**PropertyVariable**	GrowthStage (CodedValueType) LeafAreaIndex (Real) PlantHeight (Real) ...	*16* *0.8* *10*

Table 12 *Proposal for a CropVariable code list.*

Designator	Definition	Unit of Measure
CropSpecies	A PlantSpecies which can be grown as Crop	CodedValue (BotanicalName)
WeedSpecies	A PlantSpecies growing where it is not wanted	CodedValue (BotanicalName)
Pest		CodedValue
PlantDisease		CodedValue
PlantDamage		
Logging		
GrowthStage		CodedValue
Region	The geographical region in which the Crop is grown	

5.2.3.4 Field conditions

The class diagram FieldConditions can be specified similar as class diagram CropConditions, but with a list for PhysicalSoilProperties as can be seen in the diagram CropFieldSpecification[63] [64]. Those variables describe a wide spectrum of properties, such as clay, sand and loam fraction, soil moisture content, the size distribution of aggregates, etc. Table 132 shows a list with suggested variables for the field conditions. It is proposed that the code list, including its definitions and units of measures should be elaborated in more detail with AgroConnect and developers.

Table 13 *Proposal for a code list of physical soil variables.*

Designator
ClayFraction
LoamFraction
SandFraction
OrganicMatterContent
BulkDensity
HydraulicConductivity
FallingLeavesCovered
SurfacedStones
WindErodedSoilSurface
WaterErodedSoilSurface
VisibleSoilSurfaceMoisture

We might lack of a set of coded values for the soil surface like: Dry, Wet, Mixed wet and dry, Flooded, Snow covered, partly snow covered, etc. For weed removal on roads and parking lots: brick pavement, gravel, etc.

[63] https://www.agroconnect.nl/Portals/10/EnterpriseArchitect/rmAgro_SubModelRobotImages/EARoot/EA25.html
[64] https://www.agroconnect.nl/Portals/10/EnterpriseArchitect/rmAgro_SubModelRobotImages/EARoot/EA19.html

5.2.3.5 Environmental data

The appearance of a plant is influenced by the environmental conditions like climate (which is the weather over a longer period) and the local weather conditions. Weather conditions describe the conditions of the weather at a certain time or time interval. For the conditions, during capturing of images the information on the location of the camera is relevant. In theory it is possible to instrument the robot with a weather station, but in practice the conditions of a nearby weather station will be used, or an interpolation will be made from several weather stations. The published class diagram shows specifications for weather data and its conditions[65] [66].
In all cases the weather conditions will be measured by the class Sensor as component of the class SensorSystem. A sensor has a location in the coordinate system of the class SensorSystem. However, in some cases the details of the individual sensors are not specified. The class WeatherStation enables to specify where the location of the complete system is provided. Measured values are seen as the result of the complete system that are measured during a time or time interval.

To obtain estimates of the weather near the robot at a particular time, from different weather stations, a new dataset must be generated by an algorithm which interpolates the data from the different weather stations in respect of location and time. It is however more likely that weather conditions are measured by a nearby weather station. In that case there is no link from the class Task, but in the present design of drmAgro there is a link from a sensor, which is part of the class SensorSystem which subsequently has, as subclass from Equipment, a position.

In practice the property values of meteorological variables will be linked to a weather station, which has an identifier, a designator, and a geographical position. Details on the used sensors are not specified since the weather station should be seen as equipment in drmAgro. Table 144 shows the minimal metadata of environmental conditions.

Table 14 *Minimal metadata of environmental conditions.*

What	class rmAgro	attributes	Example
Weather conditions	MeteorologicalVariable	PropertyVariableDesignator QuantityType PropertyVariableDesignator RateType PropertyVariableDesignator DoubleRate PropertyVariableDesignator QuantityType PropertyVariableDesignator DoubleRate	*Average Daily Temperature* *25 °C* *Average Daily Windspeed* *1.5 m/s* *Total Daily Rainfall* *10 mm* *Avarage Daily Cloudness* *75%* *Global radiation* *1000 J/cm2*
	TimeIntervalType	Duration (DurationType) StartTime (DateTimeType) StopTime (DateTimeType) Status (StatusEnumeration)	*24h* *2022-05-07T00:00:00.00000000* *2022-05-08T00:00:00.00000000*

The assumption is here that weather conditions are provided for the class CropField on which the images are made or the weeding operation is performed. A value of a class MeteorologicalVariable can be of the type QuantityType (for example temperature), the rate type (for example windspeed in m/sec) and DoubleRate (required for evaporation, which is liter per square meter per time unit).

[65] https://www.agroconnect.nl/Portals/10/EnterpriseArchitect/rmAgro_SubModelRobotImages/EARoot/EA53.html
[66] https://www.agroconnect.nl/Portals/10/EnterpriseArchitect/rmAgro_SubModelRobotImages/EARoot/EA51.html

5.2.3.6 Used sensors

Typically weed robots make use of sensors to define their position on the field. This could be for instance a GNSS receiver or wheel encoder. Weed robots use these sensors to time the action of an actuator after the detection of objects with an algorithm. Furthermore, an image can be geotagged using a GNSS receiver. With the intrinsic and extrinsic parameters of the camera(s) and the coordinates derived from the GNSS receiver, the objects on images can be translated to real world positions. For this the extrinsic parameters (position in engineering coordinate system of vehicle) of the GNSS receiver or wheel encoder must be known. Table 15 shows the minimal metadata required for sensors.

Table 15 *Minimal metadata on sensors.*

What	class rmAgro	attributes	Example
Model and serialnr	Sensor ➔ Component ➔ EquipmentIdentifier	Designator (String) SerialNumber (String) Model ModelYear (Int) Series PartNumber (String)	
Position sensor on vehicle	NavigationReferencePoint ➔ Point → in → EngineeringCRS	Position (DirectPositionType)	
Sensor orientation	EulerRotation in EngineeringCRS Equipment	Alpha (Real) Beta (Real) Gamma (Real)	
Used coordinate system GNSS receiver	Point ➔ Primitive ➔ Geometry.srsName	srsName	*WGS84*
(Optinal) Projection / Datum	GeodeticDatum ➔ AbstractDatum		*WGS 84 UTM Zone 32N*
GNSS-date and time	PropertyVariable & PropertyValue	GpsUtcDate GpsUtcTime	
Accuracy GNSS	PropertyVariable & PropertyValue	HDOP PDOP	

The output of National Marine Electronics Association (NMEA) messages from GNSS receivers is WGS84 by default. At certain stages in processes, the lat long in NMEA messages coming from a GNNS receiver has to be converted to a Point in a coordinate system with transformation matrices. Point is subclass of the classess Primitive and Geometry. Furthermore, the class Geometry has the attribute srsName which indicates the coordinate system.

Sometimes the geodetic coordinate system is projected with a certain datum (e.g. WGS 84 UTM Zone 32N). Datum's are described in GML, as can be seen in Figure 5*. An addition of GeodeticDatum and ImageDatum in drmAgro is required.*

A GNNS receiver has no rotation in a coordinate system. It is a "point" which receive radio signals which are used to determine length to satellites. It might be that the GNNS receiver belongs to a sensor system with an additional gyroscope or compass like sensors, but then it will always be mounted such that it has no rotation in the vehicle it is mounted on.

A specification of encoders are the Ticks per mm. This requires calibration of a sensor. This requires an algorithm with parameters, which in this example can be very simple (Distance = parameter x tics). This is processed on the equipment and in most cases will not be relevant for exchange of image data.

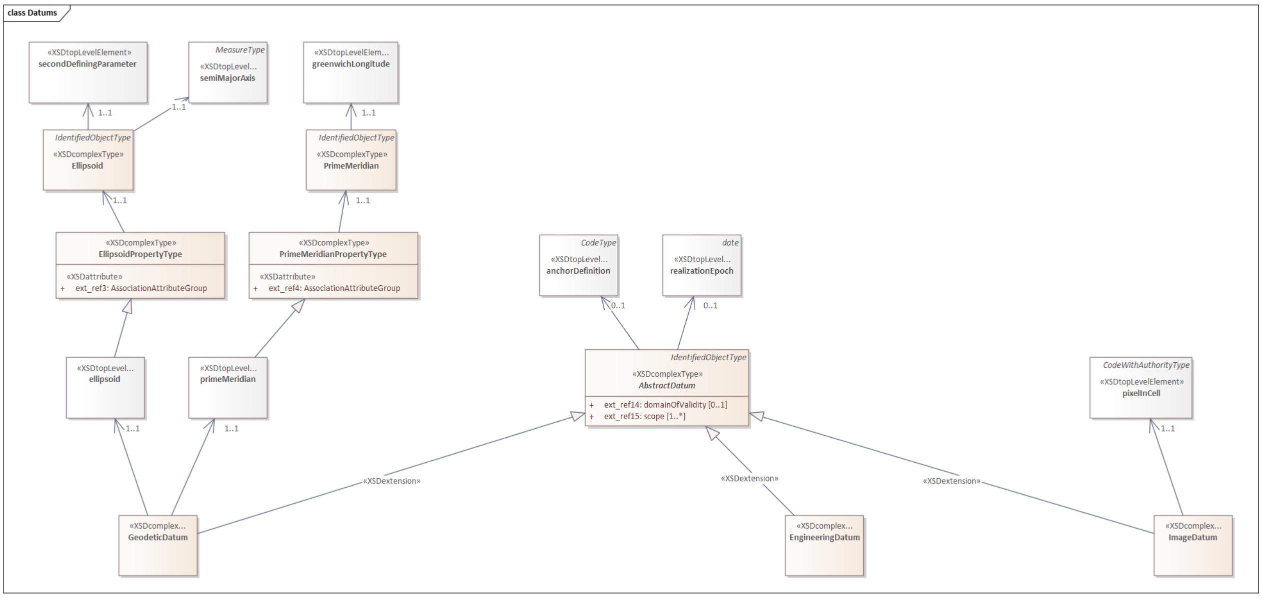


Figure 5 *Three Datum's as specified in GML3.2 (GeodeticDatum, EngineeringDatum and ImageDatum).*

5.2.4 Image coordinate systems

There are several coordinate systems specified for images, and sometimes different expressions are used for the same coordinate system. For example, there are multiple formats of bounding box annotations where each format uses its specific representation of the bounding box coordinates in the image coordinate system. These differ for example between Pascal VOC, COCO, albumentation and YOLO[67] .

We could not identify a standard identifier list of image coordinate systems. Several image coordinate systems were found after conducting a query on the internet from existing software such as Wolfram, Open eVision, Matlab and Polarmask [68] [69] [70] [71] :

- Fractional: The origin is the top/left corner of the image and has value 0.0,0.0 . The bottom/right corner of the image has 1.0,1.0. The coordinate values in the image are the fractional numbers between 0.0 and 1.0.
- Image or graphics coordinates. The origin is bottom, left corner of the image. It has a real value starting with 0.0,0.0 on the left bottom corner.
- Integer coordinates indicates each pixel by row number and column number, starting left on top, with 0,0 (Zero counting)
- Matrix or Index coordinates. indicates each pixel by row number and column number, starting left on top, with 1,1 (So no zero counting is used following Wolfram). Is identical to Pixel indices.
- Pixel indices indicates each pixel by row number and column number, starting left on top, with 1,1 (So no zero counting is used following MatLab). Is identical to Matrix or Index coordinates.
- Real coordinates references to the image itself and corresponds with the pixel indices. The integer values correspond with the left top of the pixel. As counting starts with 0,0 the centre of the first pixel is 0.5,0.5. The boundaries of the image are therefore 0,0 and numCols,numRows.
- Spatial coordinates are positions on a continuous plane. It is a cartesian coordinate system which can be intrinsic or world coordinate.
  - Intrinsic references to the image itself and corresponds with the pixel indices. Be aware however that the sequence of the indices is reversed! The integer values correspond with the center of the pixel. The boundaries of the image are therefore 0.5,0.5 and numCols +0.5, numRows+0.5.
  - World coordinate can be any other coordinate system.

[67] https://albumentations.ai/docs/getting_started/bounding_boxes_augmentation/
[68] https://support.wolfram.com/25330?src=mathematica
[69] https://downloads.euresys.com/PackageFiles/OPENEVISION/22.12.0.1176WIN/375403288/open_evision-release-notes-22.12.0.1176.pdf
[70] https://www.mathworks.com/help/images/image-coordinate-systems.html
[71] https://www.programmersought.com/article/53473033654/

- Polar coordinates are represented by the angle and distance of points relative to the center of mass of an object, describing a contour of an object in the image[72].

When looking at the examples for the COCO dataset format, one sees in some example's integer values, indicating either pixel indices and matrix or index coordinates, or integer coordinates (which have zero counting). In another example real values are used, indicating intrinsic spatial coordinates or real coordinates. COCO does, as far as we could find, not specify which image coordinate system to use.

Within ROS the tf-package takes care of keeping track of multiple coordinate frames within a system[73]. A camera system with its image coordinate system could be part of the tf-tree structure. In case of a camera often a second coordinate system is used where the z-axis faces forward, x-axis right and y-axis down. However, for outdoor purposes the North-East-Down convention is sometimes used, where the X faces North, Y faces East and Z faces down. For geographic locations ROS works standard with a (local) coordinate frame where the X-axis faces east and the Y-axis faces (true) north and the Z-axis faces Up. This system is called East-North-Up[74].

The overview of used Image Coordinate Systems shows that coordinates can have both integer and real values. Therefore, it is not required to define a specific image geography object like ImagePolygon, ImagePoint or ImageBoundingBox. (This was proposed in an earlier stage when the impression was that ImageCoordinates would have only integer values, while OGC defined geometries have real values). In case pixel or image coordinate systems are intended, only the whole number fraction of the real should be used.

The proposal is to follow OGC's structure for the geometries. DrmAgro has its own platform independent specification of geometries, but in case of transformation to a platform specific model, these will be replaced by the platform specific ones. In case of an XML based platform this is GML, in case of generating java interface model, this is based on the GeoTools library.
OGC defines an ImageCRS as a specific category of engineering coordinate system to indicate which coordinate system is used for images. As all geometries in GML (and in ISO19111 and in GeoTools) inherit from the Class Geometry, the geometries Envelope and Polygon inherit the attribute srsName, which is the reference to the coordinate system used. The attribute srsName is of type anyURI, so an appropriate URI must be determined by, for example, AgroConnect. It is proposed to use the expressions Image, Integer, Pixel, Real and Spatial as the name for the image coordinate system used.

5.3 Other relevant class diagrams

5.3.1 Evaluation data

During vision-based weed control there are two aspects which determine performance. The first is the quality of object detection based on the images including the algorithm and its parameters. With a validation dataset a theoretical average performance of the algorithm can be evaluated and expressed in several accuracy parameters as described in the class ClassificationValidation under Algorithms. However, when the algorithm is used in a practical situation the algorithm performance can be evaluated by monitoring the DetectionProbability's of objects on the image, which is calculated by the algorithm itself.

The second is the successful removal of identified weeds without removal of crop plants (field evaluation). This can be controlled by comparing images of the detection camera with images made by a second camera behind the actuator. This requires that an algorithm must be able to make the difference between "standing plants" and plants which are cut or uprooted and lying on the soil surface. The result is the WeedingPerformanceData. WeedingCriteria are used to determine whether the vision-based robot weeding operation can be continued or should be stopped.

[72] https://towardsdatascience.com/object-localization-segmentation-with-polar-coordinates-62be64da0097
[73] http://wiki.ros.org/tf
[74] https://www.ros.org/reps/rep-0103.html

1. When the probabilities for correct recognition of weeds as determined by the algorithm for vision-based weed detection, DetectionProbability, is below a certain critical level during a period it might not make sense to continue weed removal based on these input data.
2. A second criteria is the fraction of weeds which are successfully removed. This WeedRemovalFraction wil be a ProcessVariable. It should not be lower than the set criteria.
3. A third criteria will be the CropPlantRemovalFraction, also to be a member in the identifier list of ProcessVariables. This fraction should not be higher than the set criteria.
4. If the weed robot removes to much crop plants, the algorithm misidentifies the crop as a weed, meaning that the algorithm gives the crop plant a higher DetectionProbabilty for the class weed than the class crop. An identification for this is to monitor the probabilities for the crop plants as well. If they are below a certain critical level during a period is also make no sense to continue weed removal.

WeedRemovalFraction and CropPlantRemovalFraction are ProcessVariables, of which values should not be exceeded during an Operation following a particular OperationTechnique. These values are specified by WarningCriteria, as shown in the published class diagram EvaluationData [75].

Another specification could be to use PropertyValues of the PropertyVariable's "Weed density" and "Plant density", which are measured by the camera which is used for weeding detection, and that of a second camera which is used for control/validation/evaluation. A difference of more than a certain percentage could be used as criteria. The warning criteria must in that case refer to PropertyVariables, which includes Cropvariables and PhysicalSoilvariables. For now, it is decided to limit it to ProcessVariables supported by the OperationTechnique.

When the WeedingCriteria are met, a WeedingAlarm is send and received at the home base, it will be evaluated and when seen as appropriate, a message is sent to the RobotServiceProvider to abort the task. Furthermore, the Process "Retraining algorithm from images during weeding" could start by generating a dataset with images which has low DetectionProbability's.

5.3.2 Ordering services

5.3.2.1 Operation technique as a robot service

A robot service for weed disease or pest control is in fact the ability to perform a specific Operation following an OperationTechnique as specified in a coding list. This suggest expanding the existing coding lists in rmAgro with the technique of weeding itself, but also indicate that it is vision based and performed by an autonomous device. A choice must be made whether this is all specified in one flat coding list, or that more normalization is required, and multiple coding lists will be needed. RobotServices specify which Operations can be performed to realize a CulturalPractise following a specified OperationTechniques, as shown in the published class diagram Operations [76].

5.3.2.2 Robotic task weed control

For executing the robotic task weed control, a message RobotTaskData must be send to the RobotServiceProvider. The published class diagram Task shows how in rmAgro the different classes are connected to assemble this message[77]. The following paragraphs shows the attributes of the message flows to request and respond on algorithms.

5.3.2.3 Request an algorithm

A request for a specific algorithm is done after a judgement is made on which one to choose[78]. The expression is however a little misleading. An Algorithm is only suited for conditions with the ParameterSet which is established after training under comparable conditions. The same algorithm can be used with different parameter sets for different conditions under which it is trained and for which it is useful. It is

[75] https://www.agroconnect.nl/Portals/10/EnterpriseArchitect/rmAgro_SubModelRobotImages/EARoot/EA23.html
[76] https://www.agroconnect.nl/Portals/10/EnterpriseArchitect/rmAgro_SubModelRobotImages/EARoot/EA37.html
[77] https://www.agroconnect.nl/Portals/10/EnterpriseArchitect/rmAgro_SubModelRobotImages/EARoot/EA41.html
[78] https://www.agroconnect.nl/Portals/10/EnterpriseArchitect/rmAgro_SubModelRobotImages/EARoot/EA3.html

therefore that the ParamerSet holds the required information on the conditions and plant species it is valid for.

5.3.2.4 AlgorithmResponse

The message AlgorithmResponse delivers the Algorithm and the parameter set which is required to apply the algorithm for the specified conditions[79]. In the use case of weed detection this algorithm will be a neural network.

5.3.2.5 AlgorithmsResponse

In the message AlgorithmsResponse all available Algorithms are send, with all additional data which is required to select on which algorithm to use[80]. The message provides one or more algorithms and for each algorithm one or more ParameterFits. The ParameterFit describes for which PlantSpecies the Algorithm is suited.

5.3.3 Licensing of datasets and services

What is called DataLicenses are in fact the delivery of Orders. When the attribute IsBlanket is set to TRUE, this means that the order allows multiply deliveries which can be at certain intervals of time. Orders can be covered by a Contract.
The delivery of data is seen as a Service as shown in the class diagram DataLicenses [81].
When developing and improving algorithms a AlgorithmServiceProvider could contact ParticipatingFarms, which can be specified using the class diagram LicensingAndOwnership[46]. Farm is specified as a subclass of Organisation, which on its turn is a subclass of Party. The specification that a Farm is participating with another Organisation must be seen as a role of an Organisation, which can be Partner. The activity in which it is a Partner can be specified by a string.

5.4 Real-world example of a dataset with annotated images

The real-world example is one image given as example as presented in Figure 17. It shows in the text file that six objects of interest are detected, which in the text file are coded as 1 or 0. Most obviously stands 1 for (volunteer) potato and 0 for sugar beet. They should be coded with their botanical name as "Solanum Tuberosum" and "Beta Vulgaris" respectively.
Several characteristics of the image are not shown in this example, so they are in the XML example file filled with data which will, not correspond with the real data of the image.

The figures on the bounding boxes suggest that a fractional image coordinate system is used and that the bounding box definition of COCO is used, since it indicates one coordinate point and the width and height. Though it must be remarked that the 6th annotation would show a lower corner which is larger than 1.0.

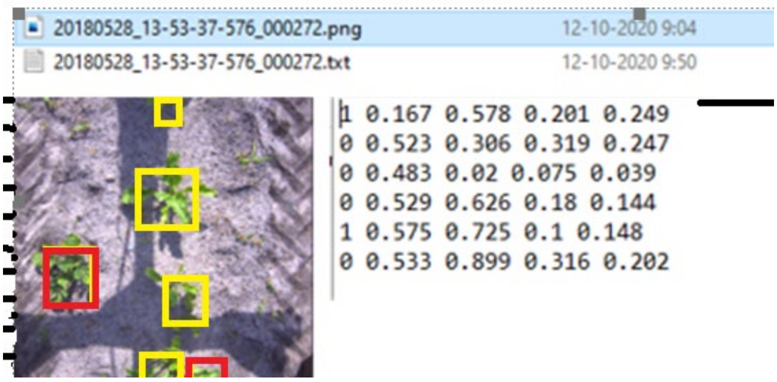


Figure 6 *Real-world example of annotated image.*

[79] https://www.agroconnect.nl/Portals/10/EnterpriseArchitect/rmAgro_SubModelRobotImages/EARoot/EA5.html
[80] https://www.agroconnect.nl/Portals/10/EnterpriseArchitect/rmAgro_SubModelRobotImages/EARoot/EA7.html
[81] https://www.agroconnect.nl/Portals/10/EnterpriseArchitect/rmAgro_SubModelRobotImages/EARoot/EA21.html

The example as XML file is shown in AnnotatedImages.xml, as part of the .zip file which is elaborated in the next section.

5.5 Steps to construct the XML file

In this section the steps are presented that are performed to construct the example XML and XSD files. These files are compressed and published in as a .zip file [82].

1. With the schema generator of Enterprise Architect, a sub model for RobotVision is generated from the drmAgro domain model for agriculture. This is done by selecting all classes, attributes of those classes, datatypes, and enumerations, which are required as data in the messages that are indicated in the BPMN's and specified in Chapter 4.
2. From the sub-model for Robot Vision, a transformation is made towards an XML model as described in paragraph 6.1.1 of rmAgroDocumentation.docx[34].
3. All classes and datatypes which correspond with GML entities or those of the CoreComponents from UNCEFACT are moved to a GML or UNCEFACT sub-package.
4. From the drmAgro specific XML model a RobotVision schema is generated, which imports the original schemas for GML and UNCEFACT respectively [83] [84].
5. The example XML file is realized by using the RobotVision.xsd as the schema constraining the XML document and DataSet as the root entity. (The disadvantage of using the xsd of the complete sub model RobotVision, is that an XML document composer will include all hierarchical related elements of the root element. For a specific message not all depending on elements are relevant, so the not relevant elements must be manually deleted in the composed xml document. For this reason, it might be valuable to evaluate the EA (Enterprise Architect) Message Composer from Bellekens[85].

PropertyVariables and Parameters

In drmAgro a clear distinction is made between PropertyVariables and Parameters.
Variable is defined by the Concise Oxford Dictionary as "that can be varied or adapted", while Parameter is defined as "quantity constant in case considered but varying in different cases".
Consequently, the expression PropertyVariable is used for all characteristics which vary, like for example soil moisture content, ambient temperature, nitrogen content, working depth, etc.
Parameters are the constants which are used in algorithms or in calibration tables and are obtained after parameter fitting or calibration.
An example is the gas law: P x V = R x T. In this case P, V and T are the PropertyValues of PropertyVariables, while T is the Parameter value of the Parameter. This is called the "gas constant".
This example shows also the second part of the Oxford definition, "… varying in different cases". The gas constant changes when the composition of the air changes.
In agriculture there can be some situations where the distinction between variable and parameter is not always that clear. An example is the saturated hydraulic conductivity. In soil moisture simulation models this could be seen as a constant, sometimes even obtained by parameter fitting instead of measurement in the field. On the other hand, the values can vary over locations within a field, and over time by tillage activities and soil compaction. Then it must be seen as a variable.

[82] https://www.agroconnect.nl/Portals/10/documenten/RobotImages/rmAgro_RobotImages_XMLandXSDs.zip
[83] http://www.opengis.net/gml/3.2
[84] https://unece.org/trade/uncefact/xml-schemas
[85] https://bellekens.com/product/bellekens-enterprise-architect-toolpack/

6 Discussion & Recommendations

An initial objective of the study was to identify gaps and provide a basis in standardization for data and algorithms concerning the domain of interest, namely image data derived from agricultural robots. Several standards were analysed, and the most important finding was that each standard covered part of the domain of interest. It is interesting to note that in most of the preferred standards for algorithms, such as ONNX and ROS2, the metadata is insufficiently specified. Therefore, in this report, we presented a class model as a basis for data sharing in the domain of interest based on the process models. It is expected that the class model will foster data sharing that support advanced analysis and facilitate a sustainable level playing field for existing and new actors.

One unexpected finding was that characteristics of standards are difficult to map with functional requirements. It is debatable which parts of standards are relevant. For example, standards that contain data dictionaries, such as the ISOXML DDI's, might not be as flexible as a standards user want to further develop the metadata that that needs to be exchanged. This finding, while preliminary, suggests that flexibility and semantic specifications could be an answer.

The relevance of abiding the GDPR compliance is clearly supported by the current findings. Each EU member state translates the GDPR regulations into a Code of Conduct for agricultural data which should give a fair level playing field for the farmer, among other things. This is ensured by requiring third parties to consider data sovereignty (consent should be given by data owners), data interoperability (apply standards for software services) and data portability (users are able to easily switch from provider). There is, therefore, a definite need for considering these principles when designing new data infrastructures.

Another important finding was that governance conditions needs to be developed further. Especially for the use of identifiers it is important that the roles and responsibilities are clear for all the value chain actors. This would facilitate image data sharing on an inter- and intra-organisational level. These results provide further support for the hypothesis that diverse types of databases, such as triple stores, and different data sharing concepts, such as federated learning, could provide a solution for governance and the use of identifiers when exchanging data.
For example, data sharing platforms such as JoinData process administrative data while it remains unclear what the exact governance guidelines should be for the proposed FAIRDataEcosystem platform (Booij, et al. 2022). Furthermore, there are ongoing initiatives and projects that address concepts like x-as-a-service and a data economy including its characteristics for an agricultural dataspace [86] [87] [88]. A further study with more focus on data governance aspects, such as roles and responsibilities, within the broader ecosystem for image data is therefore suggested.

This study aimed to assess the importance of semantic interoperability in image data and algorithms derived from agricultural robots. It is interesting to note that in all the standards that are analyzed in this study, different methods were used by the creators of these standards. It is questionable whether the established standardization methods with standards like ISOBUS, provide support for sharing image data. There is a growing body of evidence that indicates the dynamics of parties using API's, such as the MIAPPE initiative, and ontologies with a standardization method which is more agile. There are different syntaxes, such as XML, JSON, etc., while it is important to understand the practical specifications, such as binary, of image data. Additionally, it is important to clarify how this data is exchanged and referred to in messages and its relationship with protocols such as S2 buckets and MQTT. For example, response messages following request could be covered by HTTP messages. Further studies, which consider these implications, will be undertaken.

[86] https://www.atlas-h2020.eu/
[87] https://agridataspace-csa.eu/
[88] https://data4food2030.eu/

The results of this study indicate a first attempt of specification for the domain of interest, while a note of caution is due here since the completeness of the domain model is debatable. Many reference data and code lists are reused, and many things exist in rmAgro, while these findings raise intriguing questions regarding the nature and extent of the relation with other standards from an international perspective (Cantera 2019). Despite an extensive elaboration of relevant standards in chapter 2, these questions remain unanswered at present. Currently, AgroConnect publishes code lists for the Dutch sector, while more standardized code lists are preferred[89]. Although organizations such as EPPO do publish such lists, there are implications with regard to denormalization[90]. An example is the specification of winter and summer wheat, which indicates another crop meaning than season plants.
An example for the completeness of the model as mentioned is the class WeatherConditions that describes the conditions of the weather at a certain time or time interval. For the conditions during capturing of images the conditions on the location of the camera are relevant. In theory it is possible to instrument the robot with a weather station, but in practise the conditions of a nearby weather station will be used, or an interpolation will be made from several weather stations. In all cases the weather conditions will be measured by the class Sensor as component of the class SensorSystem. A sensor has a location in the coordinate system of the class SensorSystem. However, in some cases the details of the individual sensors are not specified. The class WeatherStation is a clear example, where the location of the complete system is provided. Measured values are seen as the result of the complete system and these values are measured during a time or time interval. To obtain estimates of the weather near the robot at a particular time, from different weather stations, a new dataset must be generated by an algorithm which interpolates the data from the different weather stations in respect of location and time. Further research should be undertaken to investigate ways to attach weather conditions to individual images.

Further, an example of considering the reuse of existing standards in the proposed class model is the extensive description of Camera in the SensorML, for remote sensing purposes. One of the questions is how much detailed additional information is needed about the correction of image distortion. For example, D, K, R and P should be attributed to a separate class Lens. The attributes Camera in SensorML do show lack of normalization. It can thus be suggested that we might model Lens, Platform, Timing of images as separate classes.
In this report we assumed that intrinsic and extrinsic parameters of the camera and lens remain the same during an operation. Therefore, these specifications could be attached once to a dataset of images. However, there are also many cases where the intrinsic and extrinsic parameters of the camera and lens vary during data collection. For example, when the camera and lens are attached to a spraying boom and the user of the sprayer decides to lower or raise the boom, then the extrinsic parameters of the camera are changed. Or if the camera has an autofocus option which varies the focus point of the lens, then the intrinsic parameters of the lens vary. In those cases, the intrinsic and extrinsic parameters should be attached to each individual image.

Another important finding was that interoperable algorithms and data for images could lack of certain quality characteristics. For example, metadata concerning used equipment, field conditions and environmental conditions needs to be made available. Although the structure of the message is similar, the content can differ since the algorithm could be trained with multiple datasets that are collected under different conditions. By adding these metadata, datasets will be retrievable for participants of the ecosystem. This allows algorithm developers to quickly assess datasets on added value for improving algorithms or developing new ones.

The results of this study do not explain the specific information need that might be there for algorithm developers in other subdomains of agriculture. Further work is required to establish the viability of the proposed minimum metadata with the determination of the information need. It can thus be suggested to conduct, for example, an analysis based on a survey that is set out for researchers in the domains of greenhouse horticulture and remote sensing.

[89] https://www.agroconnect.nl/en-us/aboutagroconnect.aspx
[90] https://gd.eppo.int/

For this study early findings were discussed with a group representing a broad expertise of developers and users of image data, algorithms, and innovative platforms. From the discussion it was found that the development of new algorithms could be speed up if algorithm providers have an overview and access of available resources and repositories. It is important to note that the pictures are of high quality, trusted and well labelled (well tagged) with the most important characteristics (type of camera used, lightning systems used, angle, etc.). One issue algorithm providers encounter is that datasets acquired by different camera systems are not easy interoperable when training algorithms. This shows that it is important to include metadata about the type of camera and data-acquisition and field- and light conditions in datasets, so developers can interpret if a dataset is useful or not to train their algorithm. In the paragraph

Used equipment the focus is on the specifications of the camera (how are images taken). However, when exchanging algorithms or image datasets it would be more interesting to point the unique selling points of the dataset with the focus on how images are perceived. Examples are the ground resolution in mm/pixel, accuracy of positioning (centimetre level, millimetre level), spectral range (RGB, Multispec, Hyperspec, etc.), used bands and bandwidth per band and output size (number of megapixels). We will include this in the model in a follow-up of the study.

Furthermore robot service providers emphasize the importance of monitoring the quality of executed jobs and flag a warning when the system detects anomalies. The client or service provider should also be able to give feedback about the quality. Also, the robot service provider needs information about regulations, weather and soil conditions and take that into account when controlling the robot. We tried to incorporate most of these considerations in the BPMN models and the class diagrams. However, we did not incorporate any data sources about local regulations of fields regarding the use of robots. This is still something which could be explored in further research.

In future investigations, it might be possible to elaborate on the process models and domain models. Due to limited time within this study, the following additions are proposed.
As part of the ecosystem, the process that describes the evaluation of weed control might be of importance. It is assumed that a farmer will in general evaluate the result when weed is controlled by a robot with a vision system that uses algorithms and actuators.
As part of the class models, the message that contains the available robot services that can be requested might need a coding list. These services could consist of available workers and equipment. The weeding robot should be covered in a coding list that contain the techniques of weeding itself, such as vision based and autonomous features. It is required to determine whether this list should be flat or normalised with multiple coding list. Furthermore, some addition in the message which specifies the availability is required.
For camera position and orientation, actors in the ecosystem should determine the required additional information on a camera. The ISO19130 contains an extensive model for the class sensor dealing this aspect. However, the mapping and modelling of this standard is expected to be intensive and complex. For example, the class DetectionArray is separately modelled which could imply as the camera in other models. Also, the classes ImageRaster and ImageVector could be described in more detail that has the TSML standard as a source (Soares, et al. 2011). For weeding criteria and weeding evaluation data a specification could be to use the attributes PropertyValues and PropertyVariable's as "Weed density" and "Plant density," which could be calculated from the image data measured from the camera used for detection and from the camera used for evaluation. The difference between weed or plant density before and after the weeding action could be used for evaluation. A difference of more than a certain percentage could be used as criteria.

7 Conclusions

The current study's purpose was to determine minimum interoperability mechanisms concerning standardization of image data and deep learning algorithms for vision-based applications. For this study, the focus was on a robot equipped with a spot sprayer which takes images of the crop, recognizes individual weed plants, and then sprays them.

We described four different business processes in this domain:

1. The development and improvement of algorithms;
2. Ordering Robot weed control. The selection of an appropriate algorithm for vision-based applications and ordering an operation.
3. Robot Weed Control Execution. The use of an algorithm during the execution of field operations;
4. Retrain an Algorithm from Images made during Weeding.

Following these processes, we identified relevant messages and data flows and described the preferred metadata in the exchange of image data and algorithms. Considering the definitions from the normative reference model Agro (rmAgro) and preferred communication protocols like ROS and ISOBUS this resulted in a semantic sub model in Unified Modelling Language containing multiple class diagrams.

The processes, messages, dataflows, and classes that are identified and modelled in this study indicate a first attempt of specification and could facilitate data sharing between actors in the proposed ecosystem.

References

Aarts, H. F. M., Arendse, H. A., & Brands, A. E. 1987a, Het globale informatiemodel Open teelten PAGV 67. 1987b Informatiemodel

Open Teelten - bedrijf. PAGV 36.

Bianco, Simone, et al. 2018. Benchmark analysis of representative deep neural network architectures. IEEE access 6:64270-64277.

Booij, Johan, Daoud Urdu, and Koen Graumans van Boheemen, Conny, 2022, Architecture principles for vision-based applications in agriculture : Initial step towards standardization. Stichting Wageningen Research.

Breemer, A., Soum, C. ,2021. Guidelines for the use of IoT related standards in Smart Farming and Food Security.

Cantera, Jose, M.; Sundmaeker, Harald; Campos, Rita; de la Vega, Fransisco; Neira, Isabel; Urdu, Daoud; Gomez, Igancio; Maroto; Fransisco, Forderer, Klaus, 2019. d3.6.-enhancement-and-configuration-of-open-platforms-and-reusable-components-002.pdf. Project Deliverable, Internet of Food and Farm 2020.

Christoph Hellmann Santos, Ragesh Ramachandran, 2020. Deliverable D1.8: Standardization needs analysis and potential contributions. AgROBOfood.

Data Sharing Coalition Expert Group, Data Sharing Coalition project team, 2021. Data Sharing Canvas : A stepping stone towards cross-domain data sharing at scale.

dos Santos, F. N., et al., 2016. Towards a Reliable Robot for Steep Slope Vineyards Monitoring. Journal of Intelligent & Robotic Systems 83(3-4):429-444.

Goense, D., 2017. a. rmAgro. Hamilton, New Zealand: Programma Precisie Landbouw. 2021, Standardisation of electronic data exchange and architecture.

Goense, Daan, 2017. b. rmAgro, a reference model for data exchange in precision agriculture. Asian-Australasian Conference on Precision Agriculture 7.

Hendler, Jim, 2009. Web 3.0 Emerging. Computer 42(1):111-113.

Jki, Ojs, 2010. Growth stages of mono-and dicotyledonous plants, 2010.

Kempenaar, Corné, et al, 2020. Haalbaarheidsstudie PL4. 0 data-ruimte: knelpuntenanalyse datagebruik op boerenbedrijf en aanbevelingen om de impasse te doorbreken. Stichting Wageningen Research, Wageningen Plant Research, Business Unit ….

Lassila, Ora, and James Hendler, 2007. Embracing "Web 3.0". IEEE Internet Computing 11(3):90-93.

Lin, Tsung-Yi, et al, 2014, Microsoft COCO: Common Objects in Context. Computer Vision – ECCV 2014, Cham, 2014, pp. 740-755. Springer International Publishing.

Ren, Junsong, and Yi Wang, 2022. Overview of object detection algorithms using convolutional neural networks. Journal of Computer and Communications 10(1):115-132.

Rudman, Riaan, and Rikus Bruwer, 2016. Defining Web 3.0: opportunities and challenges. The Electronic Library 34(1):132-154.

Sirin, Evren, et al, 2007. Pellet: A practical OWL-DL reasoner. Journal of Web Semantics 5(2):51-53.

Soares, Marinalva Dias, Luciano Vieira Dutra, and Nandamudi Vijaykumar, 2011. TSML: A XML-based Format for Exchange of Training Samples for Pattern Recognition in Remote Sensing Images. Revista de Informática Teórica e Aplicada 17(1).

Urdu, Daoud, et al, 2022. Semantic modelling of plant protection products data: Proof of concept: variable rate application of soil herbicide.

List of footnotes

Websites

1. https://viso.ai/computer-vision/what-is-computer-vision/
2. https://github.com/ultralytics/YOLOv5
3. https://agrobofood.eu/project/
4. https://data.overheid.nl/dataset/10674-basisregistratie-gewaspercelen--brp-
5. https://nationaalgeoregister.nl/geonetwork/srv/dut/catalog.search#/metadata/b812a145-b4fe-4331-8dc6-d914327a87ff
6. https://gdpr-info.eu/issues/personal-data/
7. https://onnx.ai/get-started.html
8. https://github.com/onnx/onnx/tree/main/onnx
9. https://www.openrobotics.org/
10. https://www.dds-foundation.org/what-is-dds-3/
11. https://www.omg.org/spec/DDS/1.4/PDF
12. https://design.ros2.org/articles/ros_on_dds.html
13. https://docs.ros.org/en/foxy/Tutorials/Custom-ROS2-Interfaces.html
14. https://roboticsbackend.com/ros2-create-custom-message/
15. https://github.com/ros2/common_interfaces
16. https://docs.ros.org/en/ros2_documentation/galactic/Concepts/About-Different-Middleware-Vendors.html
17. https://www.dds-foundation.org/omg-dds-standard/
18. https://www.omg.org/spec/DDS/
19. https://opendds.org/
20. https://cyclonedds.io/index.html
21. https://zenoh.io/
22. https://www.eprosima.com/
23. https://gitlab.inesctec.pt/agrob/vineslam_stack/vineslam/-24. /blob/master/docs/interfaces.md
24. http://docs.ros.org/en/api/vision_msgs/html/msg/Detection2DArray.html
25. http://docs.ros.org/en/melodic/api/sensor_msgs/html/msg/PointCloud2.html
26. https://www.miappe.org/
27. https://emphasis.plant-phenotyping.eu/services/emphasis-pilots/harmonisation-pilot#:~:text=The%20harmonisation%20pilot%20service%20will,of-the-art%20procedures
28.
29. https://inspire.ec.europa.eu/about-inspire/563
30. https://www.iso.org/standard/53798.html
31. https://inspire.ec.europa.eu/documents/Metadata/INSPIRE_MD_IR_and_ISO_v1_2_20100616.pdf
32. https://www.iso.org/standard/59221.html
33. https://docs.geostandaarden.nl/md/mdprofiel-iso19119/
34. https://www.iso.org/standard/67253.html

35. https://www.agroconnect.nl/Portals/10/documenten/rmAgro/rmAgroDocumentation_oct_2020.docx
36. http://www.opengis.net/gml/3.2
44. https://github.com/onnx/onnx/blob/main/docs/IR.md
46. https://www.neuraldesigner.com/learning/tutorials/testing-analysis
53. https://committee.iso.org/sites/tc211/home/projects/projects---complete-list/iso-19130-1.html
56. https://towardsdatascience.com/what-are-intrinsic-and-extrinsic-camera-parameters-in-computer-vision-7071b72fb8ec
57. http://schemas.opengis.net/gml/3.2.1/datums.xsd
60. https://albumentations.ai/docs/getting_started/bounding_boxes_augmentation/
61. https://albumentations.ai/docs/getting_started/keypoints_augmentation/
68. https://albumentations.ai/docs/getting_started/bounding_boxes_augmentation/
69. https://support.wolfram.com/25330?src=mathematica
70. https://downloads.euresys.com/PackageFiles/OPENEVISION/22.12.0.1176WIN/375403288/open_evision-release-notes-22.12.0.1176.pdf
71. https://www.mathworks.com/help/images/image-coordinate-systems.html
72. https://www.programmersought.com/article/53473033654/
73. https://towardsdatascience.com/object-localization-segmentation-with-polar-coordinates-62be64da0097
74. http://wiki.ros.org/tf
75. https://www.ros.org/reps/rep-0103.html
84. http://www.opengis.net/gml/3.2
85. https://unece.org/trade/uncefact/xml-schemas
86. https://bellekens.com/product/bellekens-enterprise-architect-toolpack/
87. https://www.atlas-h2020.eu/
88. https://agridataspace-csa.eu/
89. https://data4food2030.eu/
90. https://www.agroconnect.nl/en-us/aboutagroconnect.aspx
91. https://gd.eppo.int/

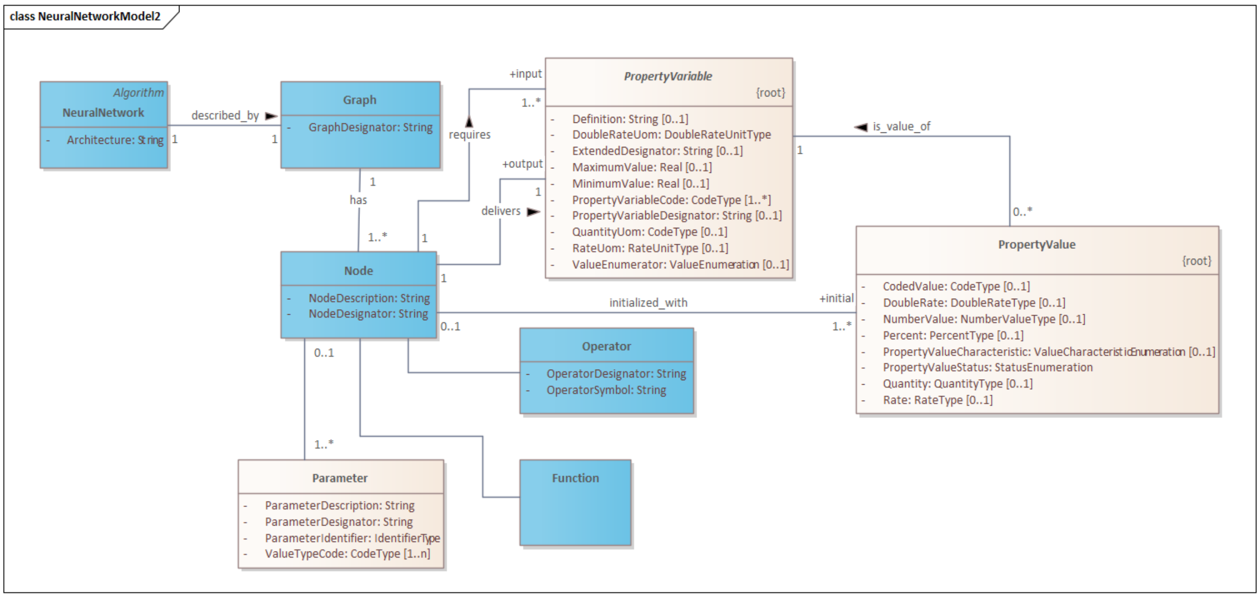
Reference model

37. **XML reference model os submodel RobotVision :**
https://www.agroconnect.nl/Portals/10/documenten/RobotImages/rmAgro_RobotImages_XMLandXSDs.zip
38. **BPMN schemes root :**
https://www.agroconnect.nl/Portals/10/EnterpriseArchitect/BPMNRobotImages/index.html
39. BPMN scheme "Developing and improving algorithms" :
https://www.agroconnect.nl/Portals/10/EnterpriseArchitect/BPMNRobotImages/EARoot/EA7.html
40. BPMN scheme "Ordering robot weed control with algorithms" :
https://www.agroconnect.nl/Portals/10/EnterpriseArchitect/BPMNRobotImages/EARoot/EA3.html
41. BPMN scheme "Executing robot weed control" :
https://www.agroconnect.nl/Portals/10/EnterpriseArchitect/BPMNRobotImages/EARoot/EA1.html
42. BPMN scheme "Retraining algorithm from images during weeding"
https://www.agroconnect.nl/Portals/10/EnterpriseArchitect/BPMNRobotImages/EARoot/EA5.html
43. **Class diagrams submodel Robotvision root :**
https://www.agroconnect.nl/Portals/10/EnterpriseArchitect/rmAgro_SubModelRobotImages/index.html
45. Class diagram algorithm and parameters :
https://www.agroconnect.nl/Portals/10/EnterpriseArchitect/rmAgro_SubModelRobotImages/EARoot/EA1.html
47. Class diagram Licensing and ownership:
https://www.agroconnect.nl/Portals/10/EnterpriseArchitect/rmAgro_SubModelRobotImages/EARoot/EA33.html
48. Class diagram TrainedNeuralNetwork:
https://www.agroconnect.nl/Portals/10/EnterpriseArchitect/rmAgro_SubModelRobotImages/EARoot/EA43.html
49. Class diagram CropFieldSpecification :
https://www.agroconnect.nl/Portals/10/EnterpriseArchitect/rmAgro_SubModelRobotImages/EARoot/EA19.html
50. Class diagram Used Equipment :
https://www.agroconnect.nl/Portals/10/EnterpriseArchitect/rmAgro_SubModelRobotImages/EARoot/EA49.html
51. Class diagram CameraIdentification :
https://www.agroconnect.nl/Portals/10/EnterpriseArchitect/rmAgro_SubModelRobotImages/EARoot/EA11.html
52. Class diagram CameraPositionAndOrientation:
https://www.agroconnect.nl/Portals/10/EnterpriseArchitect/rmAgro_SubModelRobotImages/EARoot/EA13.html
54. Class diagram Image :
https://www.agroconnect.nl/Portals/10/EnterpriseArchitect/rmAgro_SubModelRobotImages/EARoot/EA31.html
55. Class diagram CameraSettings :
https://www.agroconnect.nl/Portals/10/EnterpriseArchitect/rmAgro_SubModelRobotImages/EARoot/EA15.html
58. Class diagram GNSS_Receiver :
https://www.agroconnect.nl/Portals/10/EnterpriseArchitect/rmAgro_SubModelRobotImages/EARoot/EA29.html
59. Class diagram AnnotatedImage :
https://www.agroconnect.nl/Portals/10/EnterpriseArchitect/rmAgro_SubModelRobotImages/EARoot/EA9.html
62. Class diagram TrainingFieldResponse :
https://www.agroconnect.nl/Portals/10/EnterpriseArchitect/rmAgro_SubModelRobotImages/EARoot/EA47.html
63. Class diagram CropConditions :
https://www.agroconnect.nl/Portals/10/EnterpriseArchitect/rmAgro_SubModelRobotImages/EARoot/EA17.html
64. Class diagram FieldConditions :
https://www.agroconnect.nl/Portals/10/EnterpriseArchitect/rmAgro_SubModelRobotImages/EARoot/EA25.html
65. Class diagram CropFieldSpecifications :
https://www.agroconnect.nl/Portals/10/EnterpriseArchitect/rmAgro_SubModelRobotImages/EARoot/EA19.html
66. Class diagram Weatherdata :
https://www.agroconnect.nl/Portals/10/EnterpriseArchitect/rmAgro_SubModelRobotImages/EARoot/EA53.html
67. Class diagram WeatherConditions :
https://www.agroconnect.nl/Portals/10/EnterpriseArchitect/rmAgro_SubModelRobotImages/EARoot/EA51.html
76. Class diagram EvaluationData :
https://www.agroconnect.nl/Portals/10/EnterpriseArchitect/rmAgro_SubModelRobotImages/EARoot/EA23.html

77. Class diagram Operation : https://www.agroconnect.nl/Portals/10/EnterpriseArchitect/rmAgro_SubModelRobotImages/EARoot/EA37.html
78. Class diagram Task : https://www.agroconnect.nl/Portals/10/EnterpriseArchitect/rmAgro_SubModelRobotImages/EARoot/EA41.html
79. Class diagram AlgorithmRequest : https://www.agroconnect.nl/Portals/10/EnterpriseArchitect/rmAgro_SubModelRobotImages/EARoot/EA3.html
80. Class diagram AlgorithmResponse : https://www.agroconnect.nl/Portals/10/EnterpriseArchitect/rmAgro_SubModelRobotImages/EARoot/EA5.html
81. Class diagram AlgorithmsResponse : https://www.agroconnect.nl/Portals/10/EnterpriseArchitect/rmAgro_SubModelRobotImages/EARoot/EA7.html
82. Class diagram DataLicenses : https://www.agroconnect.nl/Portals/10/EnterpriseArchitect/rmAgro_SubModelRobotImages/EARoot/EA21.html
83. **Download XSD scheme of submodel RobotVision :** https://www.agroconnect.nl/Portals/10/documenten/RobotImages/rmAgro_RobotImages_XMLandXSDs.zip

Annex 1 Mapping ONNX based specification of neural network models on drmAgro

This appendix copies parts of the specification by ONNX and proposes a mapping on drmAgro [91].
Where italics are used, it is additional text from the author(s) of this appendix. *Cursive* text indicates some issues or question that were found during the analysis and needs follow-up actions. The analyses of the ONNX specification is not complete yet and requires at least a thorough review.



[91] https://github.com/onnx/onnx/blob/main/docs/IR.md

Model[92]

Name	Type	Description	*drmAgro*
ir_version	int64	The ONNX version assumed by the model.	*Not relevant for drmAgro*
opset_import	OperatorSetId	A collection of operator set identifiers made available to the model. An implementation must support all operators in the set or reject the model.	*OperatorSet*
producer_name	string	The name of the tool used to generate the model.	*Not relevant*
producer_version	string	The version of the generating tool.	*Not relevant*
domain	string	A reverse-DNS name to indicate the model namespace or domain, for example, 'org.onnx'	*Algorithm.AlgorithmIdentifier*
model_version	int64	The version of the model itself, encoded in an integer.	*Algorithm.Version*
doc_string	string	Human-readable documentation for this model. Markdown is allowed.	*Algorithm.Description*
graph	Graph	The parameterized graph that is evaluated to execute the model.	*Algorithm → Graph*
metadata_props	map<string,string>	Named metadata values; keys should be distinct.	
training_info	TrainingInfoProto[]	An optional extension that contains information for training.	*Nnn.TrainingInformation*
functions	FunctionProto[]	An optional list of functions local to the model.	*Function [0..*]*
model_author	string	A comma-separated list of names.	*[0..1]*
model_license	string	Name or URL.	*Algorithm → [0..1] License*

OperatorSet[93]

Name	Type	Description	*drmAgro*
magic	string	The value 'ONNXOPSET'	*Not relevant*
ir_version	int32	The ONNX version corresponding to the operators.	*Not relevant*
ir_version_prerelease	string	The prerelease component of the SemVer of the IR.	
ir_build_metadata	string	The build metadata of this version of the operator set.	
domain	string	The domain of the operator set. Must be unique among all sets.	***OperatorSet**.OperatorSetIdentifier*
opset_version	int64	The version of the operator set.	***OpersatorSet**.Version*
doc_string	string	Human-readable documentation for this operator set. Markdown is allowed.	***OpersatorSet**.Description*
operator	Operator[]	The operators contained in this operator set.	***Operator [1..*]***

[92] https://github.com/onnx/onnx/blob/main/docs/IR.md#models
[93] https://github.com/onnx/onnx/blob/main/docs/IR.md#operator-sets

Operators[94]

Name	Type	Description	*drmAgro*
op_type	string	The name of the operator (case sensitive), as used in graph nodes. MUST be unique within the operator set's domain.	
since_version	int64	The version of the operator set when this operator was introduced.	
status	OperatorStatus	One of 'EXPERIMENTAL' or 'STABLE.'	
doc_string	string	A human-readable documentation string for this operator. Markdown is allowed.	

Functions[95]

Name	Type	Description	drmAgro
name	string	The name of the function	**Function.**FunctionDesignator
domain	string	The domain to which this function belongs	
doc_string	string	Human-readable documentation for this function. Markdown is allowed.	**Function.**FunctionDescription
attribute	string[]	The attribute parameters of the function	Function → Attribute
input	string[]	The input parameters of the function	?? Function → PropertyVariable
output	string[]	The output parameters of the function.	?? Function → PropertyVariable
node	Node[]	A list of nodes, forming a partially ordered computation graph. It must be in topological order.	**Function → Node**
opset_import	OperatorSetId	A collection of operator set identifiers used by the function implementation.	**Function → OperatorSet**

[94] https://github.com/onnx/onnx/blob/main/docs/IR.md#operators
[95] https://github.com/onnx/onnx/blob/main/docs/IR.md#operators

Graphs[96]

Name	Type	Description	drmAgro
name	string	An optional name of the node, used for diagnostic purposes only.	*Some attributes of Graph are defined as attributes of other classes.* ***Graph.****GraphDesignator*
input	string[]	Names of the values used by the node to propagate input values to the node operator. It must refer to either a graph input, a graph initializer or a node output.	***Graph*** *→ <input>* ***PropertyVariable*** *Or* ***Graph*** *→* ***PropertyValue*** *(in case of initialization)*
output	string[]	Names of the outputs used by the node to capture data from the operator invoked by the node. It either introduces a value in the graph or refers to a graph output.	***Graph*** *→ <output>* ***PropertyVariable***
op_type	string	The symbolic identifier of the operator to invoke.	***Graph*** *→* ***Operator***
domain	string	The domain of the operator set that contains the operator named by the op_type.	***Graph*** *→* ***Operator.****OperatorIdentifier*
attribute	Attribute[]	Named attributes, another form of operator parameterization, used for constant values rather than propagated values.	***Graph*** *→* ***Parameter***
doc_string	string	Human-readable documentation for this value. Markdown is allowed.	***Graph****.Description*

Names Within a Graph [97]

Namespace	Description	drmAgro
Attribute	The names of attributes of an operator. Unique for each operator. **Operator →** [1..*] **Attribute**	
Value	The names of values – node inputs & outputs, tensor values (if named), graph inputs, outputs.	*Text is confusing; "Value" is used for input and output, while earlier in the document "Parameter" is used.*
Node	The names of graph nodes.	
Graph	The names of graphs within a domain, unique within the model domain.	
Operator	The names of operators within a domain.	
Shape	The names of tensor shape variables – scoped to the value information records of a graph, which is where shape variables occur.	

[96] https://github.com/onnx/onnx/blob/main/docs/IR.md#graphs
[97] https://github.com/onnx/onnx/blob/main/docs/IR.md#names-within-a-graph

Nodes [98]

Name	Type	Description	*drmAgro*
name	string	An optional name of the node, used for diagnostic purposes only.	***Node****.NodeDesignator*
input	string[]	Names of the values used by the node to propagate input values to the node operator. It must refer to either a graph input, a graph initializer or a node output.	***Node*** *→ <input>* ***PropertyVariable***
output	string[]	Names of the outputs used by the node to capture data from the operator invoked by the node. It either introduces a value in the graph or refers to a graph output.	***Node*** *→ <output>* ***PropertyVariable***
op_type	string	The symbolic identifier of the operator to invoke.	***Node → Operator***
domain	string	The domain of the operator set that contains the operator named by the op_type.	***OperatorSet.****Identifier → [1..*]* ***Operator.***
attribute	Attribute[]	Named attributes, another form of operator parameterization, used for constant values rather than propagated values.	***Node*** *→ [1..*]* ***Parameter***
doc_string	string	Human-readable documentation for this value. Markdown is allowed.	***Node****.Description*

Attributes [99]

Name	Type	Description	*drmAgro*
name	string	An optional name of the node, used for diagnostic purposes only.	***Node****.NodeDesignator*
input	string[]	Names of the values used by the node to propagate input values to the node operator. It must refer to either a graph input, a graph initializer or a node output.	***Node*** *→ <input>* ***PropertyVariable***
name	String	The name of the attribute. Must be unique among attributes, inputs, and outputs for any given operator and node.	***Parameter****.Designator*
doc_string	String	Human-readable documentation for this value. Markdown is allowed.	***Parameter.****Description*
type	AttributeType	The type of the attribute, determining which of the remaining fields is used to hold the value of the attribute.	***Parameter.****ValueTypeCode*
f	Float	A 32-bit floating-point value.	*ValueTypeCodeList.Real*
i	int64	A 64-bit integer value.	*ValueTypeCodeList.Integer*
s	byte[]	UTF-8 string.	*ValueTypeCodeList.String*
t	Tensor	A tensor value.	*ValueTypeCodeList.Tensor*
g	Graph	A graph.	*ValueTypeCodeList.Graph*
floats	float[]	A list of 32-bit floating-point values.	*ValueTypeCodeList.Real*
ints	int64[]	A list of 64-bit integer values.	*ValueTypeCodeList.Integer*
strings	byte[][]	A list of UTF-8 strings.	*ValueTypeCodeList.String*
tensors	Tensor[]	A list of tensor values.	*ValueTypeCodeList.Tensor*
graphs	Graph[]	A list of graphs.	*ValueTypeCodeList.Graph*
ref_attr_name	String	The name of a parent function's attribute.	

[98] https://github.com/onnx/onnx/blob/main/docs/IR.md#nodes
[99] https://github.com/onnx/onnx/blob/main/docs/IR.md#attributes

Annex 2 Table of standards for data and algorithms

Table 16 *Standards for data and algorithms.*

Standard name	Short description	Relevance	Source
ADAPT	Agricultural Data Application Programming Toolkit (ADAPT) aims to enable interoperability between software anad hardware applications and eliminate obstacles to use of precision agriculture data.	The standard includes a data model that is largely based on ISOBUS standard.	https://adaptframework.org/
Agricultural Information Model	The AIM is an ontology that reuses many agricultural standards and ontologies to ensure semantic interoperability. The standards that are considered are NGSI-LD, Saref4Agri, ADAPT, FOODIE, AGOVOC, INSPIRE.	Although the ontology is still in development and relatively new, it is a promising development for interoperability between systems in the food and agricultural domain.	http://agroportal.lirmm.fr/ontologies/DEMETER-AIM/?p=summary
AGROVOC	A long-term initiative of the FAO as a valuable tool and database for the classification of data that facilitate reuse and interoperability.	The concepts and terms in AGROVOC could contain relevant concepts and terms that could be reused for the case of this study.	https://www.fao.org/agrovoc/about
AgroRDF	A RDF based publication of the AgroXML.	Assumably, increased semantic interoperability compared to AgroXML.	http://data.igreen-services.com/agrordf
AgroXML	A standardized language, based on XML, for data sharing in agriculture between farm management information systems and other value chain actors.	Although the language seems to have potential, it is outdated and not easily findable and accessible.	https://link.springer.com/chapter/10.1007/978-0-387-77745-0_45
COCO	The COCO dataset is developed with the aim to advance the state-of-the-art in object recognition broadening the context of object recognition and the understanding of it.	Since object recognition is one of the key functionalities of an agricultural robot that deals with weed management, the COCO standard is considered as an important existing initiative.	https://arxiv.org/abs/1405.0312
E-Crop	UNCEFACT based standard for specifying plant products for the scope of farm management systems. The standard should support a wide range of actors in the food supply chain, such as grower, advisor, contractor, etc.	While a broad range of use cases are covered in this standard, the adoption of the standard is seemingly high. At least the Dutch paying agency (RVO) is one of the key-users of the standard. The standard uses many normative specifications of the rmAgro.	https://www.wur.nl/upload_mm/6/f/9/0e55dbbc-4874-4e6c-9399-cfee01a1c27a_Presentatie%20Webinar%20FarmDigital%20Frans%20van%20Diepen.pdf
FOAF	Friend of a Friend (FOAF) is a machine readable ontology to describe persons, their activities and their relationships to other people and objects.	FOAF describes an element Image. Though it is in FOAF only intended for images representing a Person, as Person is the central element in FOAF.	http://xmlns.com/foaf/0.1/
FOODIE	Farm-Oriented Open Data in Europe (FOODIE) was a co-funded research project within the Competitiveness and Innovation Framework Programme (CIP) in 2017. The main purpose is to facilitate the use and promotion of open data for agricultural applications. The project aimed to enable the (re)use of open data and provide added value to stakeholders in the agricultural domain[100].	Reuse of data is one of the main pillars of FAIR. Therefore, it is assumed that this project contributed and is relevant to some extent to interoperability developments of images data.	https://www.researchgate.net/profile/Karel-Charvat/publication/305851288_FOODIE_DATA_MODELS_FOR_PRECISION_AGRICULTURE/links/57a3af9d08ae3f4529247b39/FOODIE-DATA-MODELS-FOR-PRECISION-AGRICULTURE.pdf

[100] https://ec.europa.eu/eip/agriculture/en/find-connect/projects/foodie-farm-oriented-open-data-europe

FoodOn	The FoodOn ontology can be used by both computers and people as a controlled vocabulary. It can be used to name all parts of animals, plants, and fungi, as well as derived food products and the processes used to make them.	While reusing interoperable technologies to model the food domain, FoodOn is well-known especially academia in Canada and is part of the Open Biological and Biomedical Ontology (OBO) Foundry.	https://foodon.org/
GS1 EPCIS	The EPCIS standard as published by GS1 aims to enable disparate applications to create and share visibility event data, both inter-and intra-organizational.	Event-based data modelling is increasingly import for the agricultural domain. Especially different common operations, such as harvesting, contains different events that are more suitable to capture with a standard such as EPCIS.	https://ref.gs1.org/standards/epcis/
Inspire	This encoding of the INSPIRE metadata in this technical specification is based on the ISO Standards ISO 19115, ISO 19119 and ISO 19139. See paragraph 2.3.2 for more information.	Inspire is relevant as it describes the use of meta data to facilitate an infrastructure for spatial information.	https://inspire.ec.europa.eu/
ISO	Standards by ISO/IEC JTC 1/SC 42 Artificial intelligence Agricultural machinery and tractors — Safety of highly automated agricultural machines — Principles for design Artificial intelligence — Data quality for analytics and machine learning (ML) — Part 1: Overview, terminology, and examples Information technology — Artificial intelligence — Assessment of machine learning classification performance Information technology — Big data reference architecture — Part 3: Reference architecture ISO19130 Geographic Information	These ISO standards were found as relevant besides the standards that are elaborated in paragraph 2.3. The ISO standards on AI and standards on automated agricultural machines are assumed to be ahead in development with regard to the common (mid-size and small size) organisations in practice.	https://www.iso.org/committee/6794475/x/catalogue/p/1/u/1/w/0/d/0 https://www.iso.org/standard/62659.html https://www.iso.org/standard/81088.html?browse=tc https://www.iso.org/standard/79799.html?browse=tc https://www.iso.org/standard/71277.html?browse=tc https://committee.iso.org/sites/tc211/home/projects/projects---complete-list/iso-19130-1.html
ISO 11783 (ISOBUS)	An official standard for data exchange between farm machinery and farm Management Information Systems. This standard is supported by all large farm machinery manufacturers.	Highly relevant as farm robots can be considered as farm machinery. The standard covers information on the machinery itself, the use of those machinery, the specifications on how to execute the field work and eventual sensor observations made during fieldwork.	ISO https://www.isobus.net/isobus/
Mozilla WebThings API	WebThings is an open platform that makes the user able to monitor and control devices over the web.	The API is an open source implementation of the W3C's Web of Things standard.	https://webthings.io/docs/
NGSI-LD	The Smart Data Models, published and maintained by the FiWare Foundataion, is the most well-known data models that reuses NGSI-LD in the food and agricultural domain.	As NGSI-LD is a standard which claims to use generic ontologies, it is worthwhile to investigate how it covers the requirements of the robot images business processes. Demeter AIM uses NGSI-LD.	https://github.com/smart-data-models/SmartAgrifood
OGC Observations and measurements	An XML implementation of the ISO Observation and Measurements conceptual model.	The schema for Sampling Features is included which is essential when using the OGC Sensor Observation Service (SOS) Interface Standard.	https://www.ogc.org/standards/om
OGC SensorThings API	The API provides two main functionalities to make the user able to interconnect in an open, geospatial-enables and unified way. These functionalities are Sensing and Tasking.	Since robotisation concerns mainly sensors and actuators. The OGC SensorThings API covers both of the essential elements for context of this study.	https://www.ogc.org/standards/sensorthings

ONNX	As described in Chapter 2.2, the ONNX standard aims to enable an ecosystem that facilitates interoperability between AI models.	A promising and community-based standard that is supported with partners that represent big tech organisations, such as Alibaba, AMD, Baidu, HP, IBM, Microsoft, Oracle, Siemens, Sony, etc.	https://onnx.ai/about.html
RDF	Resource Description Framework (RDF) is one the most common standard models for data interchange on the Web.	As one of the generic schema's, RDF has features that facilitate data integration and flexible schema development. Therefore, it could be one of the core component of the FAIRDataEcosystem. Although the RDF might be invaluable for describing images, an example of the the ImageDescription given in Source column.	https://www.w3.org/RDF/ https://www.w3.org/wiki/ImageDescriptionRdfExamples
rmAgro	Reference Model Agro, covers a wide range of the agricultural domain.	rmAgro will contain all required data for the business processes around robot images. Mapping to (parts of) other relevant standards will be shown.	https://rmagro.org/
Saref4Agri	With the extension of SAREF, the SAREF4AGRI is an ontology that is developed with a list of use cases, standards and requirements for the food and agricultural domain. The standard is published by ETSI in the TS 103 410-6 and more description is is provided in the TR 103 511.	The cross domain interoperability, namely the food and agricultural domain and IoT domain, makes the SAREF4AGRI relevant.	https://mariapoveda.github.io/saref-ext/OnToology/SAREF4AGRI/ontology/saref4agri.ttl/documentation/index-en.html
Semantic Sensory Network Ontology	An ontology published by the OGC as an W3C recommendation. The model describes sensors and their observation, the involved procedures, the studied feature of interest, the samples to do so, and the observed properties as well as actuators.	Since robotisation concerns sensors and actuators. The SSN ontology covers both essential elements for context of this study.	https://www.w3.org/TR/vocab-ssn/
SensorML	The Sensor Model Language (SensorML) has the primary aim to provide specifications for the semantics of the processes and processing components that concern the measurement and post-measurement transformation of observations.	Sensor ML describes and models sensors in a generic way with special attention to remote sensing recording from satellites.	https://www.ogc.org/standards/sensorml
TSML	The TSML effort proposes a structure based on eXtensible Markup Language (XML) to store training data sets for specifically supervised classification algorithms (Soares, et al. 2011). The main advantage is the ability to share examples among classifiers from different applications to analyse and compare results. This characteristic aligns with the initial goal of this study.	TSML is one of standards which covers the image part of the business processes. For geospatial and remote sensing data, the diversity of formats makes it difficult to share data. This is especially the case when there is a desire to design advanced applications, such as robot vision, knowledge discovery, pattern recognition, data analysis and data integration.	https://seer.ufrgs.br/rita/article/view/rita_v17_n1_p13
W3C Web of Things	The W3C's Web of Things (WoT) standard aims to prevent the fragmentation of the IoT by using and extending standardized web technologies, metadata and other re-usable technological building blocks.	Since W3C WoT enables easy integration across application domains and IoT platforms, it could potential support the operationalization the FAIRDataEcosystem.	https://www.w3.org/WoT/

To explore the potential of nature to improve the quality of life

Wageningen University & Research
Field Crops
Edelhertweg 1
PO Box 430
8200 AK Lelystad, The Netherlands
T (+31)320 29 11 11

www.wur.nl/openteelten
Report WPR-OT 1019

The mission of Wageningen University & Research is “To explore the potential of nature to improve the quality of life”. Under the banner Wageningen University & Research, Wageningen University and the specialised research institutes of the Wageningen Research Foundation have joined forces in contributing to finding solutions to important questions in the domain of healthy food and living environment. With its roughly 30 branches, 7,200 employees (6,400 fte) and 13,200 students and over 150,000 participants to WUR’s Life Long Learning, Wageningen University & Research is one of the leading organisations in its domain. The unique Wageningen approach lies in its integrated approach to issues and the collaboration between different disciplines.