Reference number of working document: ISO/TC 23/SC 19 N 000

Date: yyyy-mm-dd

Reference number of document: ISO/WD 7673‑1

Committee identification: ISO/TC 23/SC 19/WG 1

Secretariat: XXXX

Precision Agriculture Irrigation Language — Part 1: Core Concepts, Processes, and Objects

Document type: International standard

Document subtype: if applicable

Document stage: (20) Preparation

Document language: E

Warning

This document is not an ISO International Standard. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an International Standard.

Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

**Copyright notice**

This ISO document is a working draft or committee draft and is copyright-protected by ISO. While the reproduction of working drafts or committee drafts in any form for use by participants in the ISO standards development process is permitted without prior permission from ISO, neither this document nor any extract from it may be reproduced, stored or transmitted in any form for any other purpose without prior written permission from ISO.

Requests for permission to reproduce this document for the purpose of selling it should be addressed as shown below or to ISO’s member body in the country of the requester:

*[Indicate:*

*the full address*

*telephone number*

*fax number*

*telex number*

*and electronic mail address*

*as appropriate, of the Copyright Manager of the ISO member body responsible for the secretariat of the TC or SC within the framework of which the draft has been prepared]*

Reproduction for sales purposes may be subject to royalty payments or a licensing agreement.

Violators may be prosecuted.

Contents Page

[Introduction v](#_Toc160650399)

[1 Scope 1](#_Toc160650400)

[2 Conformance 2](#_Toc160650401)

[3 Normative references 2](#_Toc160650402)

[4 Terms and definitions 2](#_Toc160650403)

[5 Symbols (and abbreviated terms) 4](#_Toc160650404)

[6 The Management Process 4](#_Toc160650405)

[7 Business Process Model and Notation (ISO/IEC 19510:2013) 5](#_Toc160650406)

[7.1 The Actors 7](#_Toc160650407)

[7.2 Core Document Model 7](#_Toc160650408)

[7.2.1 Document Contents 8](#_Toc160650409)

[7.2.2 Relationship among the Core Documents 8](#_Toc160650410)

[7.2.3 Formalizing the concept of document 9](#_Toc160650411)

[8 Central Concepts 9](#_Toc160650412)

[8.1 Identity 9](#_Toc160650413)

[8.1.1 Compound Identifier Type 10](#_Toc160650414)

[8.1.2 UniqueId 10](#_Toc160650415)

[8.1.3 Sharing UniqueIds 11](#_Toc160650416)

[8.2 Time 12](#_Toc160650417)

[8.2.1 TimeScope 13](#_Toc160650418)

[8.2.2 TimeDataSourceEnum 14](#_Toc160650419)

[8.3 Space 14](#_Toc160650420)

[8.3.1 Position 15](#_Toc160650421)

[8.3.2 Location 15](#_Toc160650422)

[8.3.3 SpatialScope 15](#_Toc160650423)

[8.3.4 RadialSpatialScope 16](#_Toc160650424)

[8.3.5 MultiPolygonSpatialScope 16](#_Toc160650425)

[8.3.6 LineString 16](#_Toc160650426)

[8.3.7 PositionDataSourceEnum 16](#_Toc160650427)

[8.3.8 Gazetteer 16](#_Toc160650428)

[8.4 Units of Measure 17](#_Toc160650429)

[8.4.1 Implementation 18](#_Toc160650430)

[8.4.2 NumericValue 18](#_Toc160650431)

[8.5 Geopolitical-Context-Dependent-Data: ContextItems 18](#_Toc160650432)

[8.5.1 Handling Static and Dynamic Requirements within a Data Model 18](#_Toc160650433)

[8.5.2 The ContextItem System Data Model 19](#_Toc160650434)

[8.5.3 Properties of ContextItem and ContextItemDefinition 20](#_Toc160650435)

[8.5.4 ContextItem 21](#_Toc160650436)

[8.5.5 ContextItemDefinition properties 22](#_Toc160650437)

[8.5.6 ContextItemDefinition: Associations 23](#_Toc160650438)

[8.5.7 ContextItemEnumItem 23](#_Toc160650439)

[8.5.8 GeopoliticalContext 24](#_Toc160650440)

[8.5.9 Language 24](#_Toc160650441)

[8.5.10 ModelScope 24](#_Toc160650442)

[8.5.11 Label 25](#_Toc160650443)

[8.5.10 Implementation Examples 25](#_Toc160650444)

[9 Reference, Setup and Configuration Data 27](#_Toc160650445)

[9.1 Reference Data 27](#_Toc160650446)

[9.2 Setup Data 28](#_Toc160650447)

[9.2.1 Self-contained data vs. URLs 28](#_Toc160650448)

[9.2.2 Meaning of Producer 28](#_Toc160650449)

[9.2.3 Producer 29](#_Toc160650450)

[9.2.4 Party 29](#_Toc160650451)

[9.2.5 PartyRole 30](#_Toc160650452)

[9.2.6 PartyRoleEnum 30](#_Toc160650453)

[9.2.7 Farm 30](#_Toc160650454)

[9.2.8 Field 31](#_Toc160650455)

[9.2.9 FieldBoundary 31](#_Toc160650456)

[9.2.10 Cropzone 32](#_Toc160650457)

[9.2.11 ConfigurationDataPedigreeEnum 33](#_Toc160650458)

[9.3 Configuration Data 33](#_Toc160650459)

[9.3.1 Rationale for configuration management 34](#_Toc160650460)

[9.3.2 Guiding principles for configuration management 36](#_Toc160650461)

[Bibliography 36](#_Toc160650462)

Introduction

Agricultural and food producers find themselves needing to feed a growing global population with limited resources, in the context of ever-increasing regulatory pressure, market price volatility, and supply-chain interest in traceability and sustainability. This increasingly requires data-driven decision-making based on sound scientific principles regarding the use of crop inputs (applied to crops through *field operations*) such as water, seeds, fertilizers, and crop protection products. The data to drive these decision processes are often captured and shared across multiple hardware and software platforms, each of which may have its own formats and code lists for representing the meaning of data. This lack of uniformity in data representation prevents the agrifood industry from realizing the promise of data-driven, principled decision-making at scale, and making a greater contribution to the Sustainable Development Goals.

Data exchange standards that enable the free flow of data among the sources of the data inputs that drive decision-making, the systems that use those inputs to produce insights and support decisions regarding field operations, and the systems that control the equipment that performs those operations, are fundamental for allowing producers and vendors alike to manage data and to make better decisions that use less crop inputs and energy resources while protecting harvest yield, quality, and sustainability. Two noteworthy benefits they provide are:

* Data exchange standards facilitate broader (i.e., more technologies) and more comprehensive (i.e., greater level of detail and precision) systems integration. This is achieved by reducing development and maintenance costs for data handling systems.
* The standards promote reliable system interoperation: standards-compliant tools are expected to work together; the end user benefits by having confidence that two products from different sources will work together as expected.

Introductory element — Main element — Part n: Part title

# 1 Scope

This Standard enables the exchange at scale of agricultural field operations data. This includes:

* The data used to drive agricultural field operation (4.9) decisions, such as weather, soil moisture, and field scouting, and
* The data used to implement those decisions as field operations (such as irrigation, spraying, and planting)

These data are currently stored and represented in a variety of proprietary original equipment manufacturer (OEM) formats. This standard seeks to provide an industry-wide data model that can be used for unambiguous data exchange by actors in the agrifood industry.

**1.1** The scope of the Standard is based on two major aspects of agricultural and food processes:

1. **Observations**: the field, atmospheric, plant, or other *in situ* measurements that can drive decision-making in agricultural and food systems. This includes weather stations, soil moisture sensors, or crop-related sensing, as well as other inputs usable by decision-support systems (4.5). This is an agrifood systems implementation of the ISO 19156 standard for observations and measurements (ISO, 2011).

2. **Operations, or field operations** (4.9): activities associated with the production of a crop in the field. This includes application of products such as seeds (i.e., planting), chemicals, fertilizers, and water (using an irrigation system (4.11)). It includes, but is not restricted to, management-level communications and record keeping. Field operations data are based around a Recommendation(4.14), which describes a suggested course of action; a Work Order (4.15), which describes an ordered course of action; and a Work Record (4.16), which describes what *actually happened*. This work is based on, and extends, the ISO11783-10 standard (ISO, 2015) for communications between agricultural machinery and farm management information systems or FMIS (4.8).

**1.2** The Standard currently consists of three parts:

**1.2.1** Part 1 (this document): Core Concepts, Processes, and Objects

* Business processes associated with field operations
* The Actors and their role(s) in the process
* Core documents and their relationships
* Common components
* Identity ― Compound identifiers
* Time, space, people, geopolitical-context-dependent data
* Reference data
* Setup and configuration data

**1.2.2** Part 2: Observations (4.13) and measurements (4.12) in agrifood systems

* Observations: An agriculture-specific implementation of the ISO 19156 standard
* Field observations, including soil water content and local field weather
* Setup and configuration of sensors and loggers
* Derived weather data

**1.2.3** Part 3: Irrigation System Operations

* Location and geometry of the irrigation system
* Documenting the operation of different parts of an irrigation system
* Flows and pressure schedules (how much and when)
* Work orders for irrigation systems
* Work records: Reporting how much and where, water and (possibly) chemicals were applied.

# 2 Conformance

ISO 7673 defines an abstract object model. Applications intending to use the ISO 7673 object model will need an implementation of the object model that is appropriate for the particular application under development. Future parts of the standard (or versions thereof) will include reference serialization implementations.

A particular implementation of the object model, whether it is for serialization or application development, would need to demonstrate some conformance to the business rules and constraints defined in 7673. Conformance testing is essential for the expectation that two separate implementations can interoperate successfully. Given the large size and scope of ISO 7673, specification of conformance requirements and associated tests will be covered in a separate part of the series.

# 3 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/FDIS 6709, Standard representation of geographic point location by coordinates

ISO 8601, Data objects and interchange formats ― Information interchange ― Representation of dates and times

ISO/IEC 13249-3:2011, Information technology ― Database languages ― SQL multimedia and application packages ― Part 3: Spatial (WKT)

ISO 19109, Geographic information ― Rules for application schema

ISO 19112:2003, Geographic information ― Spatial referencing by geographic identifiers

ISO 19156:2011, Geographic information ― Observations and measurements

ISO/IEC 19501:2005, Information technology — Open Distributed Processing — Unified Modeling Language (UML) Version 1.4.2

# 4 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

4.1

centre pivot irrigation machine

automated irrigation machine consisting of a number of self-propelled towers supporting a pipeline rotating around a pivot point and through which water supplied at the pivot point flows radially outward for distribution by sprayers or sprinklers located along the pipeline

ISO 11545:2009, 2.1

4.2

crop plan

a high-level document describing how a crop will be grown on a given area of land during a crop season. It can range from very abstract to very specific.

4.3

cropzone

A parcel of land, not necessarily contiguous (e.g., an agricultural field or paddock, or a part thereof), that an agricultural producer is growing a particular crop or genotype on, during a specified period of type (e.g., a crop season).

4.4

datum

parameter or set of parameters that define the position of the origin, the scale, and the orientation of a coordinate system.

ISO 19111:2019, 3.1.15

4.5

decision support system

DSS

a computerized information system used to help individuals or organizations make informed, decisions that may involve data analysis, integration of model results, and the application of a set of defined rules, criteria and guidelines. A Decision Support System is a type of, or may be part of, a Farm Management Information System.

4.6

direct position

position described by a single set of coordinates within a coordinate reference system.

ISO 19129:2009(en), 4.1.11

4.7

document

recorded information or material object which can be treated as a unit in a documentation process.

ISO 5127:2017, 3.1.1.38

4.8

farm management information system

FMIS

office computer system used by a farmer or contractor that includes the software for farm management, such as book-keeping, payroll, resource management for machines, products, workers, field management, geographical information system, decision support systems and task management

ISO 11783-1:2017(en), 3.23

4.9

field operation

a task or activity, whether planned or actual (i.e., that already happened), that is connected with the production of a crop. Planting, spraying, spreading, tillage, irrigation and harvest are all examples of field operations.

4.10

global navigation satellite system

GNSS

satellite based navigation system that provides autonomous global positioning of a receiving device

Note 1 to entry: Global positioning system (GPS), and global navigation satellite system (Glonass), Galileo and BeiDo are typical examples of global navigation satellite systems.

ISO 19901-10:2021(en), 3.41

4.11

irrigation system

assembly of pipes, components, and devices installed in the field for the purpose of irrigating a specific area

ISO 20670:2023(en), 3.49

4.12

measurement

a set of operations having the object of determining the value of a quantity.

ISO 19101-2: 2018, 3.1.41

4.13

observation

act carried out by an observer to determine the value of an observable property of an object (feature-of-interest) by using a procedure, with the value provided as the result.

ISO 19156:2023, 3.13

4.14

recommendation

a course of action proposed by someone with *expertise* (e.g., an agronomist, crop scout or irrigation consultant). A recommendation is advisory (“This is what I recommend we should do.”)

4.15

work order

a set of instructions, given by someone with authority for a specific course of action to take in a field operation. (“This is what we are going to do.”) Two examples: an irrigation work order describes the amount of water to apply to a specified area of land over a certain time period; a seeding work order describes seed products (e.g., varieties, seed treatments) and the rates to apply them with (e.g., seeds per unit area) over a specified area of land, over the time period of interest.

4.16

work record

a document that describes what actually happened while performing the field operation (4.9). ("This is what was actually done.") It may differ from the work order because the operator performing the field operation may have adjusted to particular conditions (e.g., suspended the operation because the ground was too wet and returned the next day.)

# 5 Symbols (and abbreviated terms)

No symbols are defined for this standard.

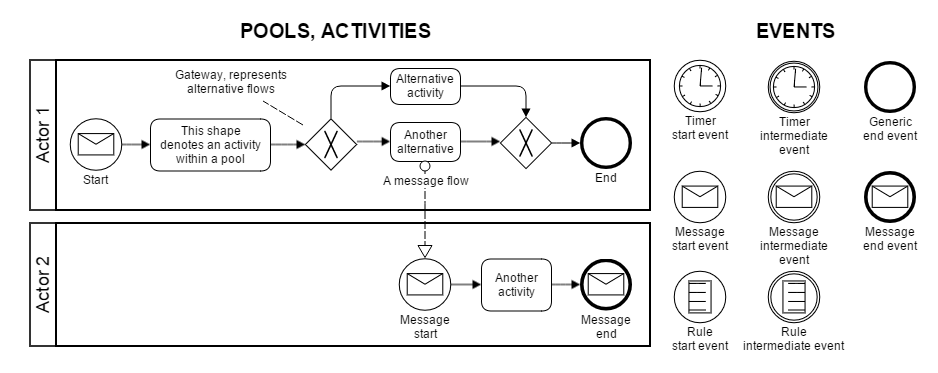
# 6 The Management Process

Field operations such as planting, irrigation and harvest are producer business processes. These processes must be supported with data exchange, both for coordination among the different actors directly involved in each process, and for communication with information consumers, such as trade partners and regulatory agencies. These processes vary in detail from producer to producer, but there is value in defining a broadly applicable core set. Modelling field operations processes enables clarity and quality control in decision-making. The models are inevitably a simplification of what happens on a farm. Nonetheless, they can serve as effective tools for supporting data-driven decisions.

# 7 Business Process Model and Notation (ISO/IEC 19510:2013)

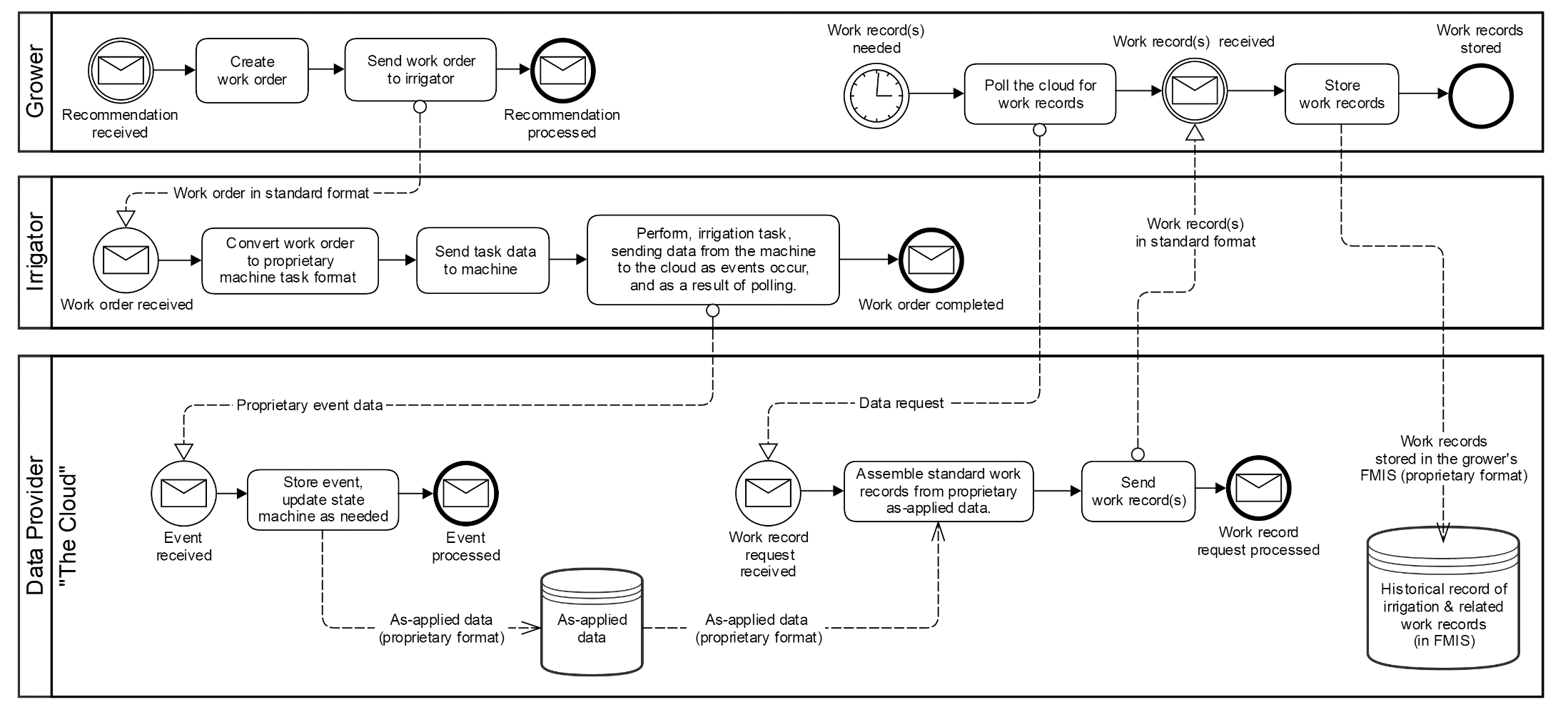
Business Process Modelling Notation (BPMN) diagrams provide a visual representation of a workflow, and the exchanges of data that take place among the actors participating in it. In this standard, BPMN diagrams represent the workflow for data-driven field operations decision processes, communicating a decision as a Recommendation (4.14) and Work Order (4.15), and acquiring a Work Record (4.16) of what happened as the work was executed. The models provide a clear representation of the workflow that can be easily understood by all participants, thus supporting a structured method for achieving high quality field operations decision-making.

Figure 1 below shows a BPMN diagram with a key / guide to help interpret the example shown in Figure 2. The diagram is divided into horizontal pools. Each pool describes a process performed by an actor, as a sequence of activities. Interactions among actors are represented as *messages* (shown as dotted lines) that can flow between pools.



**Figure 1 – BPMN key diagram**

Different processes are initiated in different ways. Those processes shown starting with a clock face icon have a scheduled start time. Those that are shown starting with an envelope icon start when the corresponding message is received. As an example, the diagram in Figure 2 shows the multiple concurrent processes that take place in an irrigation operation.



**Figure 2 – Sample BPMN diagram for precision irrigation data**

Part 2 of this Standard uses a specific BPMN for observations and measurements, while Part 3 describes a BPMN for irrigation operations.

## 7.1 The Actors

This standard seeks to enable the data exchange needs of the various *Actors* in agricultural field operations, such as producers, their farm staff, consultants, and service providers. The BPMN diagrams (e.g., Fig. 2) specify field operations actors, the processes they participate in and the component activities thereof, and the data flows that support those processes. There are many actors in agrifood systems; an exhaustive listing is outside the scope of this standard, but a small set in the context of the irrigation example (Fig 2) is shown in Table 1.

Actors exchange data in the form of documents (4.7) shown as dotted lines in the BPMN diagram. These documents are described in abstract terms in 7.2. There is some field operation specificity in the properties of these core documents, that will be covered in separate parts of this standard, e.g., core documents for irrigation are described in Part 3.

**Table 1 –Actors in the irrigation example and the core documents they exchange**

|  |  |
| --- | --- |
| **Actor** | **Description** |
| Producer | Has authority. Uses that authority to create Work Orders out of Recommendations received from the Consultant. |
| Irrigator | Uses Work Order received from the Producer to initiate a Field Operation. |
| Consultant | Has expertise. Uses that expertise to translate data into a document called a Recommendation. The data is received from the Producer (Crop Plan (4.2)) and procured from a Data Provider (observations & measurements). |
| Data Provider | A data provider:   * Collects and stores various forms of observations and measurements (O&M) data. * Makes the O&M data available to the Consultant. * Collects and stores proprietary irrigation operations event data. * Derives Work Records from the irrigation operations event data * Makes the Work Records available to the Producer.   NOTE: more than one Data Provider could perform these tasks. For example, the irrigation operations could be handled by one provider, the weather data by another, and the soil moisture by yet another. |

Please note: All of these actors, or parties, may refer to persons or to organization fulfilling the stated role (Irrigator, Consultant, etc.). See 9.2.4 and 9.2.5 for more details.

## 7.2 Core Document Model

In the BPMN diagrams shown in Figure 2, Producers and their partners use multiple documents to exchange field operations information as part of their business processes. There has been work done on standardizing a nomenclature of farm processes (ISO 22006), but the documents and many of the terms used within them have yet to be unambiguously defined. The core documents shown in Table 2 define the data exchanged during a field operation (4.9). The definitions are flexible, in view of the myriad ways in which different producers implement their record-keeping in response to region-specific regulatory requirements, particular characteristics of their markets or farming operations, operation size, and personal preference.

**Table 2 – The core documents in field operations**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Document** | **Role** | **What It Conveys** | **Actor Involved** | |
| Crop Plan | Strategic | A high-level document describing how a crop will be grown on a given piece of land during a crop season.  “This is how we’re going to grow this crop this season.” | Producer, or other actor involved in the strategic planning for the field operations. | |
| Observations and Measurements | Tactical/ Predictive | A document containing data measured/ observed in the field.  “This is what’s happening (or what we think might happen) in the field.” | Crop scout, remote observation or a party tasked with monitoring or forecasting conditions in the field. | |
| Recommendation | Tactical | “This is what I recommend we should do.”  This document is not always acted upon; it is acted upon via a work order, upon approval. | An individual, such as a consultant or agronomist, or other party with the expertise/ licensing necessary to recommend a course of action. | |
| Work Order | Tactical | “This is what we are going to do.” | An individual or organization with authority to order the work done. | |
| Work Record | Tactical/ Historical | “This is what we actually did in the field.” | May be automatically generated; otherwise, an operator that performed the task. | |
| **Reference, Setup, and Configuration Data** | | | |  |
| Reference Data and Setup File | All | “This is the common information we need in order to set up and support accurate and efficient data exchange.” | Producer, or other party involved in managing the producer’s production data. |  |

NOTE The use of the word “document” in this context is not intended to refer to XML documents, JSON files, or any serialization method in particular. It refers to the ISO 5127 definition (4.7) of a document as recorded information that can be treated as a unit in a documentation process.

NOTE The “Role” column qualifies the documents in terms of their timeliness, or time interval in which they are actionable: ‘Strategic’ refers to longer (e.g., crop season) time scales and ‘Tactical’ refers to short time scales (e.g., one field scouting mediated decision cycle).

### 7.2.1 Document Contents

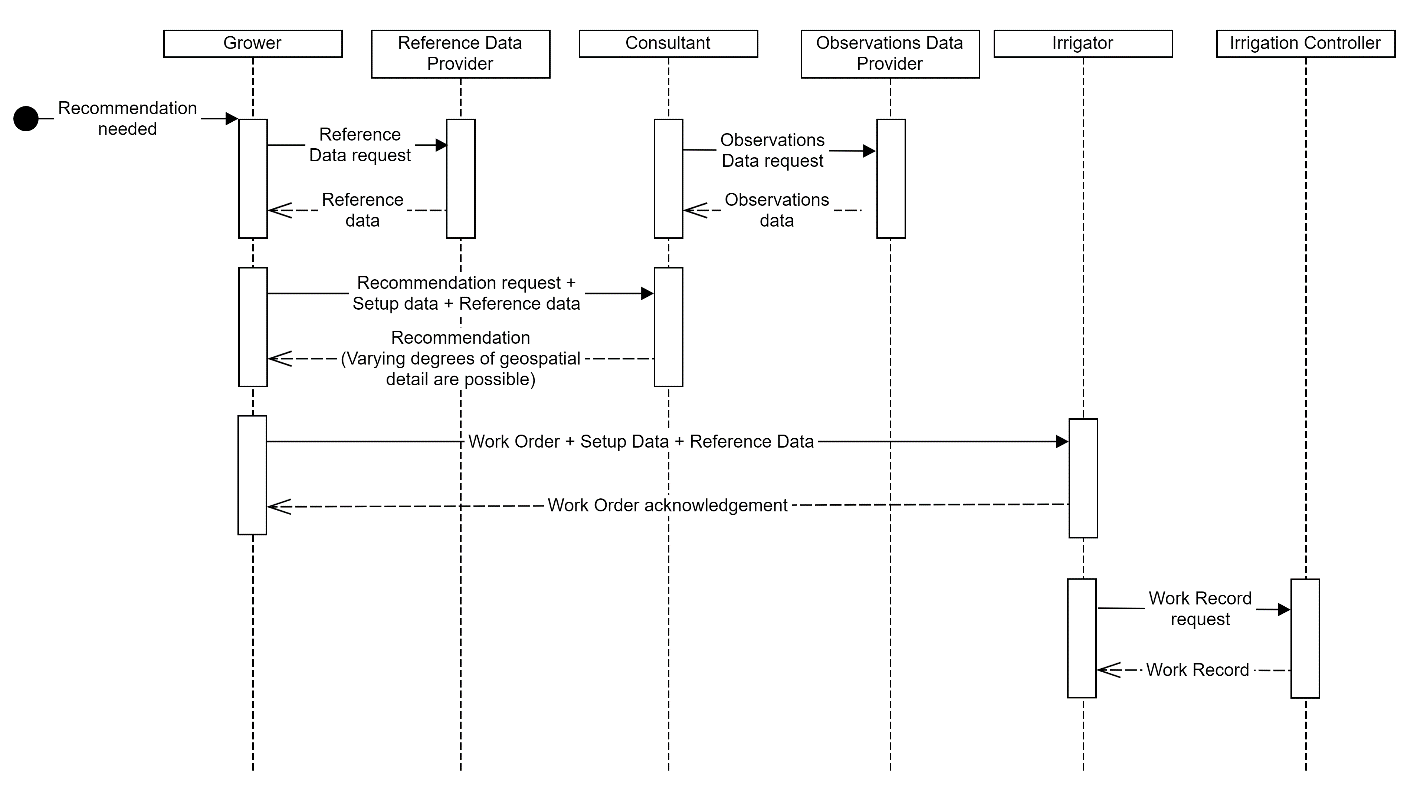
Documents contain information focused on completing a field operation (4.9). A work item may, for example, be the specific order to place a specified amount of water over a specific area of a field. Specific contents of a document may include some, or all, of the detailed items shown in Table 3.

**Table 3 – Core document contents (using irrigation example from part 3 of this standard)**

|  |  |
| --- | --- |
| **Content** | **Description** |
| What | Amounds of water (and possibly other products), specified using IrrItems in the specific classes that use them (Plan, Recommendation, Work Order, and Work Record); e.g., water or chemicals applied. |
| Where | ProducerRef, FieldRef, CropZoneRef, etc. These are references to the corresponding object instances of a producer, a field, a crop zone, etc. where the irrigation operation takes place. |
| Who | PersonRoleRefs, references to objects encapsulating the actors involved, and the role they play. |
| When | TimeScopes represent various times and time intervals associated with the field operation, ranging from crop season, to requested start time, to actual start time. |
| How | This is specified using IrrItems in the corresponding classes; e.g., application rates. |
| With What | This is specified using IrrItems in the corresponding classes; e.g., references to the machine (or sections thereof) that are applying water and other products. |

### 7.2.2 Relationship among the Core Documents

The documents rarely stand on their own: they exist as part of a complex network of decision-making processes within a producer’s operation. Core Documents are linked through various relationships: causal (e.g., a Recommendation *informs* a Work Order), contextual (e.g., a product label helps identity a fertilizer used in a Work Order), and compositional (e.g., observations and measurements can form part of a Work Record (4.16), documenting weather conditions during the field operation). Figure 3 shows the Core Documents, the causal relationships among them, and the contextual relationships afforded by Reference, Setup and Configuration data, in the form of a UML sequence diagram.

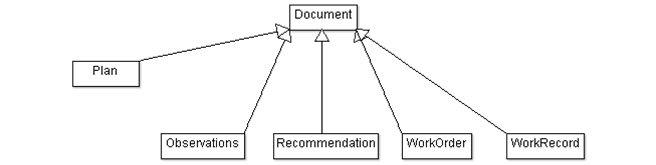


**Figure 3 – Core Document relationships**

### 7.2.3 Formalizing the concept of document

This standard defines a class called Document, shown in UML class diagram form in Figure 4, along with its descendant classes, the core documents. The Documents class contains data common to all the Core Documents; each of the specific core documents additionally contain specific properties not shared by the others.

NOTE The Plan class isn't defined in this document but will be defined in a subsequent part of ISO 7673.



**Figure 4 – The Document class, along with its descendant classes**

# 8 Central Concepts

This section describes fundamental concepts that are used throughout the standard, introducing each object as a concept. The concepts presented in this section manifest in the standard as class definitions. This section introduces each object as a concept and describes its purpose.

## 8.1 Identity

Many objects specified by this standard are referenced in other objects (for example, a producer, farm and field may be referenced in a work order) and thus need identifiers that can be used by the referencing object. Figure 5 shows a Unified Modeling Language (UML, as per ISO/IEC 19501:2005) class diagram of the mechanism used by this standard to manage identifiers and references thereto. It centers on an object class called CompoundIdentifier, which provides objects with a string identifier (the ReferenceId) for use in the local scope of any particular instance of a data model and allows associating an arbitrary number of (optional) unique identifiers (the list of UniqueIds) to that ReferenceId. CompoundIdentifiers work as a sort of “basket of identifiers” that associate an object with one or more identifier-source pairs, analogous to what is enabled in ISO11783 Annex E-mediated Linklist files.

Every class that must “stand alone” and be referenced by other objects uses compound Identifiers. The exception is reference data classes, which may define a more human-friendly code as the mechanism to use to reference an instance of that reference data class.



**Figure 5 – The CompoundIdentifier and UniqueId classes and supporting enumerated vocabularies.**

### 8.1.1 Compound Identifier Type

The CompoundIdentifier accommodates the need for unique identifiers in the context of keeping datasets as compact as possible. The properties of a CompoundIdentifier are:

**referenceId**: This string serves as a unique identifier for a particular instance of an object. The scope of uniqueness shall be appropriate for its use: it shall be at least locally unique, but if the data objects are meant to be exchanged through some form of application programming interface (API), they shall be globally unique. The value of a referenceIds can be created in a distributed fashion (i.e., without a centralized naming authority), if there is reasonable certainty of their uniqueness.

The CompoundIdentifier also has a compositional relationship with 0..\* UniqueId objects (see 8.1.2).

### 8.1.2 UniqueId

CompoundIdentifiers are meant to be used in a distributed context, where a document may circulate among two or more FMIS (4.8). Since each FMIS may have its own set of unique identifiers (for chemical products, or farms, fields and cropzones, for example), it can be convenient to include multiple available identifiers, i.e., instances of the UniqueIds class, for the same object during exchange.

The UniqueId defines a unique identifier that will be associated with a locally scoped reference identifier string as part of a CompoundIdentifier. The uniqueness constraints shall be identical to the constraints applied to CompoundIdentifier’s referenceId property. It can reference a source, i.e., the organization that “minted” (created) that identifier.

The properties of the UniqueId class are:

**idText**: A string that contains the identifier proper. Required. Can be created in a distributed fashion (i.e. without a centralizing naming authority, as long as there is a reasonable certainty of its uniqueness. The idText property of a UniqueId instance is itself stored as a string, but the category of UniqueId, which indicates how to interpret the string, is specified with the idCategory property.

**idCategory**: An enumerated value from the class UniqueIdCategoryEnum that describes what kind of identifier the idText string should be interpreted as (e.g., UUID, URI, string). Required

**idSource**: A string that contains an identifier indicating the originating organization of the unique identifier. Optional. Source may be a GLN (GS1, 2014) or URI. Should be a unique string from a controlled vocabulary.

**idSourceCategory**: An enumerated value from the class idSourceTypeEnum, that describes how the idSource text should be interpreted. Optional.

### 8.1.3 Sharing UniqueIds

One of the purposes of bundling UniqueIds along with a CompoundIdentifier when exchanging data among systems is to enable actors to avoid the unwelcome and relatively painful (e.g., requiring direct user interaction) process of record linkage; i.e., scenarios where correspondences (“mappings”) must be established among UniqueIds referring to the same object, but originating in different systems.

Figure 6 below shows two data exchange example scenarios, where a Producer, an Advisor, and an Irrigator exchange documents pertaining to planning, executing and documenting an irrigation field operation. In the example, each actor is assumed to have their own locally derived unique identifier to represent the Producer’s field.

The top panel in Figure 6 below shows the process of data exchanges as existing without cooperation; i.e., the three actors do not share uniqueIds, just the referenceId of the exchanged document. Note how, upon receiving a data payload, every one of the actors must spend time mapping the incoming ids to known ids.

The lower pane of Figure 6, in contrast, shows how, when the actors do include UniqueIds within the CompoundIdentifiers they share, eventually the record linkage step can be avoided enabled by network effects that make record linkage unnecessary. Idx refers to the identifier for a resource such as a field: Idp is the identifier that the Producer had for the resource a priori, IdA is the one the Advisor had, and IdI is the one originally held by the Irrigator.







**Figure 6 –UniqueId mapping scenarios. The top figure shows three data exchanges where the UniqueIds of a CompoundIdentifier are not shared, and record linkage had to be performed in all cases. The bottom figure shows a scenario where each actor shares all the identifiers it has for a resource, ultimately enabling network effects that make record linkage unnecessary. Idx refers to the identifier for a resource such as a field: Idp is the identifier that the Producer had for the resource *a priori*, IdA is the one the Advisor had, and IdI is the one originally held by the Irrigator.**

## 8.2 Time

Accurately capturing the time at which various events happen is an important part of agricultural record keeping. Examples include capturing the application date of chemicals in spraying operations, which can have regulatory compliance implications, and resolving mass balance in irrigation operations, where water volumes are frequently calculated as a flow rate (e.g., in liters per minute) multiplied by duration. Figure 7 shows (as a UML class) the mechanism used by this standard to represent time. It consists of two timestamps, a required **contextCategory** property that specifies the meaning of the TimeScope through an enumerated vocabulary (See Table 4), and an optional human-readable **description**.



Figure 7 – UML Class diagram of the TimeScope type

### 8.2.1 TimeScope

The TimeScope class is a flexible tool for specifying time. It is used both to specify timestamps (i.e., moments in time) and time intervals. The ISO8601 notation shall be used when serializing the **start** and **end properties** in text form, regardless of which serialization technology is used (e.g., XML JSON, CSV).

**contextCategory:** Enumerated value, defined by the DateContextEnum type, which describes the meaning of the TimeScope. The enumerated values are shown in Table 4.

**start:** This time stamp is optional to accommodate the case where there is a time interval or timing event that does not have a precise date but does have a time-related meaning; e.g., a crop season.

**end:** Optional second time stamp, used to denote a time interval.

**description:** Optional name for the TimeScope. A usage example: “2024 Crop Season.”

Note: This class is a simplification of the TimeScope used in AgGateway’s ADAPT toolkit (AgGateway, 2016).

Table 4 – DateContextEnum enumerated values

|  |  |  |  |
| --- | --- | --- | --- |
| **Value** | **start** | **end** | **Annotation** |
| ACTUAL | Required | Optional | A time stamp, used to specify the date/time in which a field operation (or part thereof) actually happened, as represented in a Work Record. |
| APPROVAL | Required | N/A | A time stamp, used to specify the date/time in which a Work Order is approved. |
| CALIBRATION | Required | Optional | A time stamp or (optionally) interval, used to specify the date/time when a given device was calibrated. |
| CREATION | Required | N/A | A time stamp, used to specify the date/time in which a document (or part thereof) was created. |
| EXPIRATION | Required | N/A | A time stamp, used to specify the date/time after which a Work Order for a field operation will no longer be valid (i.e., will expire). |
| INSTALLATION | Required | Optional | A time stamp, used to specify the date/time in which a piece of equipment was installed. |
| LOAD | Required | Optional | A time stamp or (optionally) interval, used to specify the date/time when a given load / amount of product was loaded onto a vehicle or container. |
| MAINTENANCE | Required | Optional | A time stamp, used to specify the date/time in which maintenance was performed on a piece of equipment. |
| MODIFICATION | Required | N/A | A time stamp, used to specify the date/time in which a document (or part thereof) was modified. |
| PHENOMENON | Required | Optional | A time stamp or (optionally) interval, used to specify when a phenomenon under observation occurred. |
| PROPOSED | Required | Optional | A time stamp, used to specify the date/time for a field operation (or part thereof) as proposed in a Recommendation. |
| REQUESTED | Required | Optional | A time stamp, used to specify the date/time for a field operation (or part thereof) as requested in a Work Order. |
| REQUESTED\_SHIPPING | Required | N/A | A time stamp, used to specify the date/time when a given load / amount of product should be shipped. |
| RESULT | Required | N/A | A time stamp used to specify when the result of an observation became available. |
| RESUME | Required | N/A | A time stamp used to specify when a suspended operation was resumed. |
| SUSPEND | Required | N/A | A time stamp used to specify when an operation was suspended. |
| UNLOAD | Required | Optional | A time stamp or (optionally) interval, used to specify the date/time when a given load / amount of product was unloaded from a vehicle or container. |
| UNSPECIFIED | Required | N/A | A time stamp with an unspecified meaning. |
| VALIDITY | Required | Optional | A time interval, used to specify the time interval during which a document such as a Recommendation is valid. |

### 8.2.2 TimeDataSourceEnum

Different data sources and equipment use different strategies for associating time to events. The TimeDataSourceEnum enumerated class, the values of which are shown in Table 5, lists a simple set of these strategies and enables the user to interpret the timing of received data.

Table 5 – TimeDataSourceEnum enumerated values

|  |  |
| --- | --- |
| **Value** | **Annotation** |
| GNSS\_ON\_EVENT | The system tags the event with a GNSS (4.10)-mediated timestamp corresponding to the moment when the event happened. This is the most accurate approach. |
| DEVICE\_CLOCK\_ON\_EVENT | The system tags the event with an internal device clock-mediated timestamp corresponding to the moment when the event happened. Unless the device clock is synchronized via GNSS or a timeserver, there may be drift, and timestamps may have systematic error. Durations will be accurate, however. |
| SERVER\_CLOCK\_ON\_TRASNMISSION | The event is tagged with a timestamp corresponding to when the event was successfully communicated to a server, which is not necessarily the same as when the event occurred. These time sources are susceptible to error in the presence of communication errors in the field. |
| DEVICE\_CLOCK\_ON\_TRANSMISSION | The event is tagged with an internal device clock-mediated timestamp corresponding to when the event was successfully communicated to a server, which is not necessarily the same as when the event occurred. These time sources are susceptible to error in the presence of communication errors in the field, as well as susceptible to clock drift. |
| MANUAL\_INPUT | The start and/or end values were input manually by a user. |
| UNKNOWN | The origin of the timestamp data used to tag events in the system is unknown. |

## 8.3 Space



**Figure 8 – UML class diagram of the space and position types**

### 8.3.1 Position

Position (formally, 4.6 direct position) specifies the geographic coordinates for an object or action, in terms of latitude, longitude and elevation. Recorded direct position data should conform to an established geographic coordinate system and datum (4.4); specifically, the geographic coordinate system and the World Geodetic System 1984 (WGS 84) datum. The properties of the Position class are:

**lat:** Degrees of latitude (WGS 84). Positive north of the equator, negative to the south. Double-precision floating point number, required.

**lon:** Degrees of longitude (WGS 84). Positive east of Greenwich meridian, negative to the west.  Double-precision floating point number, required.

**elev:** Assumed to be meters above the WGS 84 geoid. This value shall be positive for values above the geoid and negative for elevations below it.  Double-precision floating point number, optional.

**gNSSSource**: An enumerated item from the GNSSSourceEnum class, indicating data quality for that position’s geographic information. Optional.

### 8.3.2 Location

Location can be understood as a Position with an identifier and additional metadata. It should also be compliant with the WGS 84 global datum. The properties of the Location class are:

**id:** A compound identifier. Used primarily in observation datasets to provide an efficient way for multiple observations to identify their (common) location. (Required)

**externalDocURL:** A URL for providing (presumed human-readable) external documentation, metadata, etc.

**description:** An optional string name.

**parentRef:** An optional reference to a parent location. This enables describing a hierarchical system of stations, sites/shelters and sampling features.

**category:** An optional description of the location’s meaning (Station, Site, Shelter, SamplingFeature, Unknown). Should emerge from a controlled vocabulary.

**lat:** Degrees of latitude (WGS 84). Positive north of the equator, negative to the south. Double-precision floating point number, required.

**lon:** Degrees of longitude (WGS 84). Positive east of Greenwich meridian, negative to the west. Double-precision floating point number, required.

**elev:** Assumed to be meters above the WGS 84 geoid. Negative values indicate meters below the WGS 84 geoid. Double-precision floating point number, optional.

**gNSSSource**: An enumerated item from the GNSSSourceEnum class, indicating data quality for that position’s geographic information. Optional.

### 8.3.3 SpatialScope

This class provides a mechanism for specifying the coverage of a field operation. When dealing with a radial section, such as a section of an irrigation centre pivot, it is most convenient to use the RadialScope (see ISO 7673-3, Clause 7.1.5); use MultiPolygonScope otherwise. The SpatialScope class is simply a container for zero or more instances of either RadialScope or MultiPolygonScope. An instance of SpatialScope may not contain both RadialScope and MultiPolygonScope instances, except in special cases (e.g., an irrigation system’s IrrSetup, described in Part 3 of this standard).

### 8.3.4 RadialSpatialScope

Specifies a start angle and end angle to describe the movement of a centre pivot system (4.1) or sections thereof. The startDeg and endDeg azimuth values are arranged such that the pivot is traversed clockwise from the start angle to the end angle. Note that a value of 0 is defined as true north as per the WGS 84 datum.

**startDeg:** Start angle, in degrees.

**endDeg:** End angle, in degrees.

### 8.3.5 MultiPolygonSpatialScope

This is a multi-polygon expressed in OGC’s WKT (Open Geospatial Consortium Well Known Text) format.

**shape:** A string encoding a multi-polygon in OGC’s WKT format.

### 8.3.6 LineString

This is a multi-segment line expressed in OGC’s WKT format.

**shape:** A string encoding a linestring in OGC’s WKT format.

### 8.3.7 PositionDataSourceEnum

The PositionDataSourceEnum provides information about the origin and accuracy of geographical position and angular position information. See examples in Table 6.

**Table 6 – PositionDataSourceEnum values**

|  |  |
| --- | --- |
| **Value** | **Annotation** |
| GNSS\_BASIC\_FIX | Single GNSS unit, with no augmentation |
| GBAS\_DGNSS | Unspecified ground-based augmentation system differential GNSS |
| GBAS\_DGNSS\_CLASSICAL | Classical differential GNSS. |
| GBAS\_DGNSS\_RTK | Real-time kinematic |
| GBAS\_DGNSS\_PPP | Precise point positioning |
| DIFFERENTIAL\_GNSS | Geographic position data is being derived from a set of two GNSS units: one mobile, and one fixed. |
| SBAS | Satellite-based augmentation system (i.e., EGNOS, WAAS) |
| MECHANICAL | Position data emerges from a mechanical sensor of some form; for example, an optical encoder in the case of a centre-pivot irrigation system. These systems are susceptible to biases emerging from an incorrect reference (i.e., zero, or north) position; users are encouraged to calibrate accordingly. |
| GENERATED | Geographic position coordinates were obtained from data processing. |
| MANUAL\_DRAWN | Geographic position data was drawn on a geographical information system – enabled platform, such as a farm management information system, or other mapping software. |
| OTHER | The source of geographic position data is known but does not conform to one of the options specified above. Users are encouraged to contact the system manufacturer or installer for details. |
| UNKNOWN | The source of geographic position data is unclear, unknown, or unspecified. |

### 8.3.8 Gazetteer

The Gazetteer class is a container of locations (i.e., named positions). The class, and its associated component, Location, are a partial implementation of ISO19112. The purpose of Gazetteer is to contain a list of Location objects that represent named places relevant to setup and configuration, typically of facilities that produce observations (4.13) and measurements (4.12). Other objects in the model can then reference these places.

The **Gazetteer** specifies a list of locations, including geographical position, an identifier, and a description. The Gazetteer shown in Figure 9 is composed only of an optional list of Location instances.



Figure 9 — The Gazetteer class, a collection of Location (i.e., named Position) objects. Note that the properties of Location are described in 8.3.2

The expected use case for Gazetteer is with weather stations, where the instruments are distributed over a (usually small) geographic area. For example, a site containing a rain gauge, evaporation pan, and anemometer might have these instruments a few meters apart. Depending on the context, a setup document might refer to the instruments individually or refer to them jointly as a single location. A data provider could, for example, specify the boundary of a collection of instruments (i.e. the fence around the site). The setup data delivered by the data provider could then associate data with the name of that site rather than specifying latitude and longitude for each instrument. While the weather station use case is the exemplar, Gazetteers are not limited to use with weather monitoring sites.

The purpose of referencing the ISO 19112 standard is to enable associating an installation or collection of instruments with a named location in addition to the explicit spatial or geographic specification. By mirroring the SI\_LocationInstance class in ISO 19112, the Location can be compatible with external Gazetteer services that conform to the ISO 19112 standard.

The Location class in this Standard is an implementation of the ISO 19112 class SI\_LocationInstance. This Standard’s Gazetteer class is a container of location instances but, despite its name, this Standard’s Gazetteer is not an ISO 19112 SI\_Gazetteer. This omission is intentional, as the ISO 19112 standard is an abstract specification. This Standard includes a realization of an object model and would thus require a realization of the ISO 19112 model or referring to an existing implementation. At the time of this writing, there is no widely accepted implementation of ISO 19112 that is also relevant to agricultural observations and measurements. This Standard does not claim conformance as a spatial reference system nor as a Gazetteer, as specified in ISO 19112:2003. Instead, the Standard assumes that a gazetteer exists and the Location instances were supplied by the external Gazetteer instance, a farm management information system, etc.

## 8.4 Units of Measure

This standard involves representing numerical data, which are frequently associated with units of measure. In many standards (e.g. ISO 11783) the unit of measure is implicit when declaring a variable of interest (i.e the data dictionary entry for that variable declares the unit of measure being assumed); for example, distances might always be assumed to be expressed in millimetres. While this approach is convenient in terms of minimizing the size of a data file, or convenient for data producers, it carries with it the problem that subsequent use of the datasets beyond the original stakeholders involved may be hampered by a lack of shared understanding of the meaning of the data, and interoperability may suffer as a result. For that reason, this standard explicitly associates units of measure to numeric variables.

There are, however, numerous controlled vocabularies in the industry used for representing units of measure (including those in the International System), without the clear emergence yet of a single standard. Thus, this standard supports several relevant unit of measure code authorities, and users shall explicitly declare the system used in documents or data files. The supported systems (note: all these systems support SI units of measure) are:

* UN Recommendation No. 20: Lists the codes for units of measure in international trade. This Recommendation establishes a single list of code elements to represent units of measure for length, mass (weight), volume and other quantities (including units of count) and covering administration, commerce, transport, science, technology, industry etc.
* QUDT (Quantities, Units, Dimensions and Types): A collection of ontologies that define the base classes properties, and restrictions used for modelling physical quantities, units of measure, and their dimensions in various measurement systems.
* UCUM (Unified Codes for Units of Measure): Includes all units of measures being contemporarily used in international science, engineering, and business. The primary focus is on electronic communication, as opposed to communication between humans. Electronic data interchange (EDI) protocols would be a typical application for UCUM.
* ADAPT: A system of codes used by AgGateway’s open-source ADAPT toolkit.

### 8.4.1 Implementation

A document compliant with this standard shall contain a single instance of the UnitOfMeasureAuthority element, specified at the top level of the document. The UnitOfMeasureAuthority element is an enumeration having the following possible values shown in Table 7:

**Table 7 – UnitOfMeasureAuthority enumerated values**

|  |  |
| --- | --- |
| **Value** | **Meaning** |
| UNREC20 | UN/CEFACT Recommendation 20 codes are used to specify units |
| QUDT | QUDT codes are used to specify units |
| UCUM | UCUM codes are used to specify units |
| ADAPT | ADAPT system codes are used to specify units |

The value of the UnitOfMeasureAuthority element applies to the entire document. The UnitOfMeasureAuthority element is defined in this part of the standard and used in all subsequent parts thereof.

### 8.4.2 NumericValue

NumericValue is used for class properties where a numeric value is necessary, the units of measure are relevant, and the unit of measure is not specified elsewhere in the class’s properties. This class has the following properties:

**value:** The value; e.g., 7.

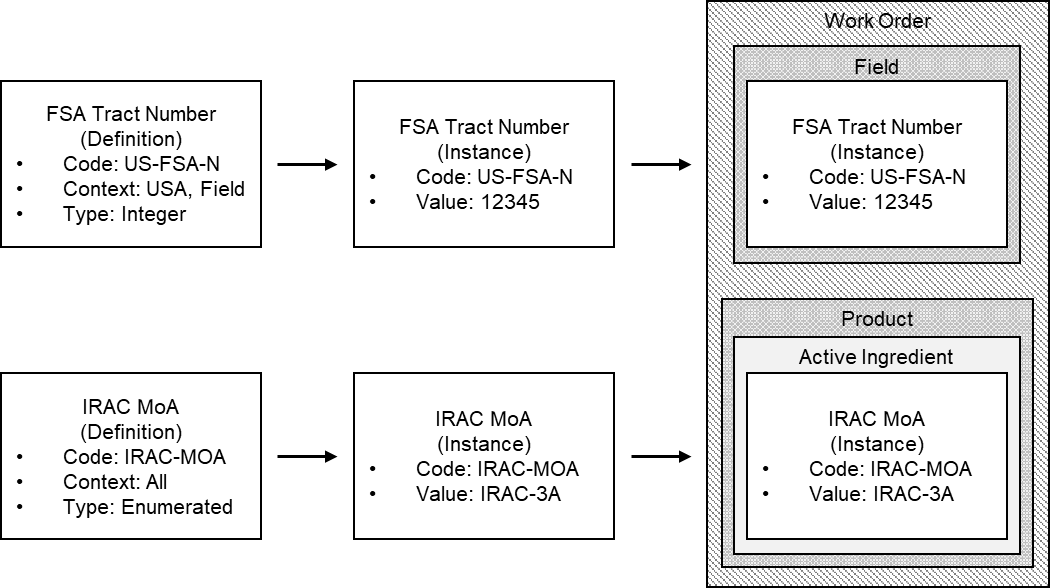
**UoM:** The unit of measure code, expressed according to the units of measure authority identified by UnitOfMeasureAuthority.

## 8.5 Geopolitical-Context-Dependent-Data: ContextItems

### 8.5.1 Handling Static and Dynamic Requirements within a Data Model

Current trends in sustainability, traceability, and compliance reporting often include significant amounts of frequently changing, geopolitical- context-dependent information, such as identification numbers specific to the government agencies that interact with the producers in their jurisdiction. Fulfilling all of these requirements in the data model of farm management information system (FMIS) software is a moving target, with both infrequently and frequently changing data. The Standard reconciles these contradictions by using an object class: the ContextItem. The ContextItem can be attached to various objects in a common object model. The ContextItem system can be used jointly with The ISO 11783-10 XML schema’s feature of associating unique identifiers to its own locally scoped identifiers defined in ISO11783-10 Annex E (ISO, 2015). This enables adding geopolitical-context-dependent data to that schema’s otherwise generic and highly machine-specific scope, with no modifications.

The ContextItem system provides a way to preserve the simplicity of data models that utilize it by allowing these models to first focus on expressing “universal” ideas with their core objects and then enhancing those objects with geopolitical-context-dependent specifics through ContextItem(s). Existing data models (like ISO 11783) can be similarly extended in a dynamic, data-driven way.

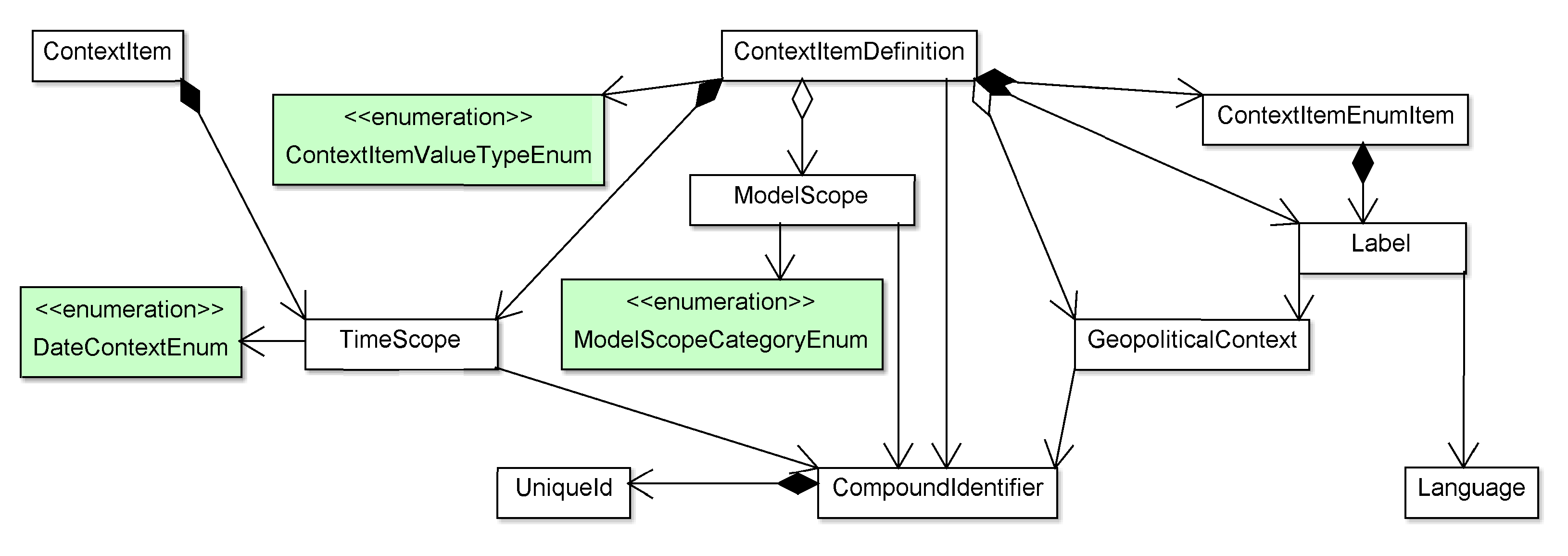


**Figure 10 – How a ContextItem is used to provide data of a specific instance of a general concept**

In Figure 10, the concept of FSA (United States Department of Agriculture; Farm Service Agency) Tract Number has an identifying code: US-FSA-N. It also has a property that indicates what data type to expect, in this case an integer. The ContextItemDefinition provides some additional properties, a set of Context(s) that states WHERE (the USA) and on WHAT (a Field) this definition is relevant. As defined earlier, the documentation of a planned field operation is known as a Work Order. It often contains WHERE to conduct the operation (a Field) and WHAT to do (apply a Product). In this example, the Field contains an Instance (or use of) the ContextItem for FSA Tract Number. Again, think of this as a key-value pair where the key is ContextItem code “US-FSA-N” and the value is “13245.” This data type registry, the ContextItem System, contains variable definitions that are used in very specific places or times.

### 8.5.2 The ContextItem System Data Model

The ContextItem is a key/value structure where the “key” code references a ContextItemDefinition that defines what each ContextItem means. The “value” is composed of a string value along with data needed to interpret it (such as a unit of measure) or a nested list of other ContextItems (e.g. Public Land Survey System cadastral information.) Figure 11 shows a simplified (in that the classes’ properties are not shown) data model of the ContextItem system, emphasizing the relationships among the classes.



**Figure 11 – Simplified ContextItem System Data Model**

Table 8 provides a brief description of the purpose of each class, including whether it is a simple enumeration, whether it is used by value or reference, and how it relates to others.

**Table 8 – ContextItem classes**

|  |  |  |
| --- | --- | --- |
| **Class** | **Type** | **Description** |
| CompoundIdentifier | Object | Provides a mechanism to reference an object from other objects, as well as a mechanism for associating one or more external unique identifiers to an object. |
| ContextItem | Object | A key/value pair used to attach geopolitical-context-dependent information to other objects in AgGateway’s ADAPT, ISO11783 files, etc. . |
| ContextItemDefinition | Object | A definition of a specific kind of ContextItem, including the type of data it can contain, what classes in an object model it can be attached to, and how to display/enter instances of it. |
| ContextItemEnumItem | Object | Description of specific enumeration items for ContextItemDefinitions with an enumerated ContextItemValueType |
| ContextItemValueTypeEnum | Enumeration | An enumerated type that describes the type of value carried by the ContextItem (Integer, Boolean, Double-precision floating point, string, enumeration, datetime, nested…) |
| DateContextEnum | Enumeration | Specifies the meaning of a TimeScope. |
| GeoPoliticalContext | Object | Describes particular jurisdiction(s) or geopolitical context(s) that a ContextItemDefinition is applicable to. |
| Label | Object | Represents a way to express something in a given combination of GeoPoliticalContext and Language. |
| Language | Object | Describes a language used to express terms in instances of Label. |
| ModelScope | Object | Used by ContextItemDefinition to denote a data model class to which a corresponding ContextItem instance can be attached. |
| ModelScopeCategoryEnum | Enumeration | Describes what data model a particular class belongs to. This version of the Standard supports ISO11783 and AgGateway’s ADAPT. |
| TimeScope | Object | Associates a date/time or range to an object. Also encapsulates the meaning of the timestamp/time interval. |
| UniqueId | Object | Captures a unique identifier as part of a CompoundIdentifier. |
| UniqueIdTypeEnum | Enumeration | An enumerated type that describes what kind of unique identifier is contained in a UniqueId (Integer, String, URI, UUID). |

### 8.5.3 Properties of ContextItem and ContextItemDefinition

A ContextItem is a single, specific, use of a ContextItemDefinition. Put a simpler way, a ContextItem is a key/value pair. Figure 12 shows the full data ContextItem system model.

The ContextItem’s Code is the “key” (corresponding to the unique Code property of a single ContextItemDefinition), while the rest of its properties describe the “value.”

* The Value property is ALWAYS expressed as a string even though that may not be how it was collected or how it is expected to be used elsewhere. The ValueType property of the associated ContextItemDefinition supplies the user with this data type information. In some cases, like the presence of NestedItems, there is no real Value to record, so this property is optional.
* The ValueUOM property contains the optional unit of measure, further defining the value. A DefaultUOM property is included with the associated ContextItemDefinition and may be used instead of including it locally with the ContextItem.
* If the ContextItemDefinition has a ValueType of “Nested,” the NestedItems collection allows a resulting ContextItem to support a hierarchical structure of other instances of ContextItem.
* An optional collection of TimeScope objects is used to record the various relationships a given Value may have with time. For example, there could be a TimeScope that captures when the Value was recorded and another TimeScope that expresses the duration for which the Value is considered valid.



**Figure 12 – The Full Data ContextItem System Model**

### 8.5.4 ContextItem

Provides a mechanism for representing geopolitical-context-dependent information as a name-value pair along with an optional unit of measure.

**code:** The code that explains the meaning of the ContextItem. Required. Expected to emerge from a controlled vocabulary (e.g., as managed by AgGateway’s Agrisemantics Committee).

**value:** The value; e.g., 7. Required unless the ContextItem is nested, in which case its children bear values.

**unitOfMeasure:** The unit of measure code, expressed according to the units of measure authority identified by UnitOfMeasureAuthority. Optional, but shall be used when the value of the ContextItem is a number with an associated unit of measure (e.g., 7 mm).

**timeScope:** Optional list of TimeScopes. Allows setting a validity range and capturing when something was tagged with the ContextItem (Use the VALIDITY and CREATION date context codes shown in Table 4, respectively)

**contextItem:** Optional nested ContextItems. A good example is the Public Land Survey System (PLSS) data used for cadastral identification in the US.

### 8.5.5 ContextItemDefinition properties

The ContextItemDefinition defines the ContextItem: what it means, how to enter it, display it, where it can be used, and so forth. It should communicate everything needed to enable that ContextItem to be captured and displayed appropriately. Properties of the ContextItemDefinition object include:

**code** (string): Required. This is the code used in a corresponding ContextItem. The following requirements apply to code values:

* The value should be universally unique within the ContextItemDefinition domain.
* The value should be used as part of the URI that forms the identity of a ContextItemDefinition.
* The value shall be the same “key” used by ContextItem in its Code property.
* The value should be issued by a central authority to ensure its uniqueness.

**valueType:** Required. Describes the expected data type for the “value” of a ContextItem. It is expressed as an enumeration with the values of BOOL, STRING, DOUBLE, INTEGR, DATETIME, ENUM, and NESTED.

* The first five data types (BOOL, STRING, DOUBLE, INTEGER, DATETIME) can be represented as a string and then parsed back to their original form.
* The “ENUM” data type indicates that this ContextItemDefinition is an encoded enumerated list. The items in the list are ContextItemEnumItem objects and are included by value in the EnumItems collection property. When creating a ContextItem using an “Enum” type ContextItemDefinition, the ContextItem Value property corresponds to the Value property of the selected ContextItemEnumItem.
* The “NESTED” data type indicates that this ContextItemDefinition is a container that encapsulates a group of other ContextItemDefinition(s). When creating a ContextItem using a “Nested” type ContextItemDefinition, the ContextItem Value property is left empty, but its NestedItems property contains a collection of ContextItem(s).
* **Description**: A human-readable name that makes it simple to correctly choose the ContextItemDefinition of interest from a pick list, etc.
* **defaultUOM:** An optional string describing the expected unit of measure. When an instance of ContextItem does not contain a value for ContextItem.unitOfMeasure, the defaultUOM property shall be used as the default unit of measure for the ContextItem.value property. The value of defaultUOM shall be a code compliant with the unit of measure authority identified by UnitOfMeasureAuthority.
* **unitOfMeasureAuthority:** An optional code that denotes the organization that minted / maintains the set of codes used to convey a unit of measure. This is an enumerated set of values, provided by UoMAuthorityEnum: QUDT, UCUM, ADAPT, and UNREC20. The default value is assumed to be ADAPT.

### 8.5.6 ContextItemDefinition: Associations

**labels:** A collection of Label objects. The concept that a ContextItemDefinition represents may be expressed verbally in multiple ways; not just in different languages, but also using different terms in the same language. For example, it may express the regional differences in the common name of a pest. A Label object contains the text, a reference to the language the text is in, and a list of GeopoliticalContext (8.5.8) objects that describe “where” that terminology is used. The Internet Assigned Numbers Authority (IANA) Language Subtag Registry (IANA, d.) will be the controlled vocabulary for languages (8.5.9).

**properties:** A collection of ContextItem objects. This is used to supply additional information needed to use the definition. For example, if a ContextItemDefinition involves capturing values for latitude and longitude, it might include in the Properties a ContextItem that indicates the geodetic datum is expected to be WGS 84.

**nestedDefIds:** A collection of references to other ContextItemDefinition(s). This is only populated if ValueType is set to “Nested”. It is used when there are groups of data points that need to be collected together rather than individually. For example, the Public Land Survey System used in the United States uses several properties (such as the Principal Meridian, Township, Range, and Section) to specify the location of a piece of land for cadastral purposes. Each property is defined through its own ContextItemDefinition and is represented as ContextItem(s) that are included by value in the NestedItems property of the PLSS ContextItem.

**enumItems:** a collection of ContextItemEnumItem objects. It is used to encode the enumerated values for a ContextItemDefinition of ValueType “ENUM”. The ContextItemEnumItem contains some of the same properties that ContextItemDefinition does. Instead of having a Code property, it has a Value. This Value is expected to be unique within the domain of the ContextItemDefinition to which it is attached. When creating a ContextItem using an “ENUM” type ContextItemDefinition, the ContextItem Value property corresponds to the Value property of the selected ContextItemEnumItem.

**timeScopes:** an optional collection of TimeScope objects. This collection enables attaching a variety of time- related properties to the ContextItemDefinition. For example, a TimeScope could describe when its ContextItemDefinition was created, another when it was updated, and a third for which the date range is valid.

**modelScopeIds:** A collection of references to ModelScope objects. The ModelScope object represents a business object in either the ADAPT model or ISO 11783 (potentially other models as well). For example, there is a ModelScope object for the ADAPT Farm class as well as a separate one for the ISO 11783 Farm element (FRM). This coded list of business objects forms the controlled vocabulary we use to specify which objects a given ContextItemDefinition can be used to describe.

**geopoliticalContextIds:** A collection of references to GeopoliticalContext objects. The GeoPoliticalContext object represents an entry in an external controlled vocabulary that describes a particular geographic/political domain or organization. This is a powerful feature that enables tagging ContextItemDefinition(s) with a marker that conveys “where” the data it represents is relevant. See 8.5.8 for more detail on the GeopoliticalContext class.

### 8.5.7 ContextItemEnumItem

This class holds the possible enumerated values that ContextItem instances of a given ContextItemDefinition with valueType = .ENUM can take. Its properties are:

**value**: A string representing a possible value taken by an enumerated ContextItem This Value is expected to be unique within the domain of the ContextItemDefinition to which it is attached.

**description**: A human-friendly description of the meaning of the value.

**labels:** A collection of Label objects. The concept that a ContextItemEnumItem represents may be expressed verbally in multiple ways; not just in different languages, but also using different terms in the same language. For example, it may express the regional differences in the common name of a pest. A Label object contains the text, a reference to the language the text is in, and a list of GeopoliticalContext (8.5.8) objects that describe “where” that terminology is used. The Internet Assigned Numbers Authority (IANA) Language Subtag Registry (IANA, d.) will be the controlled vocabulary for languages (8.5.9).

The ContextItemEnumItem class also has an association:

**properties**: An optional list of ContextItems that can provide additional context to the enumerated value in question. A good example of this are the geographical coordinates of the centers of coordinates for the different principal meridians of the US Public Land Survey System (PLSS).

### 8.5.8 GeopoliticalContext

The preferred source of codes depends on the geopolitical context level. The properties are:

**code**: a required string that represents the jurisdiction through a standardized encoding, depending on the administrative level of the geopolitical context of interest, as follows:

* country: If the geopoliticalContext of interest is a country (e.g., Argentina, or France, or Japan) populate the code property with an ISO 3166-1 Alpha 2 code (e.g., "AR", "FR", "JP") and populate the sourceVocabulary property with the string ISO\_3166\_1\_ALPHA\_2.
* first-level administrative (e.g., state, province, oblast): Use ISO 3166-2 codes, and populate sourceVocabulary with ISO\_3166\_2
* Second-level or something not otherwise fitting the above: Use either the FAO's Geopolitical Ontology (http://ww.fao.org.countryprofiles/geoinfo/en/) or geonames.org codes, and populate sourceVocabulary with FAO\_GEOPOLITICAL or GEONAMES.

**sourceVocabulary**: a required string from a controlled vocabulary, set as described in code, above.

**description**: a required string with a human-friendly description of the corresponding code value (e.g., “France”).

### 8.5.9 Language

This class describes a language used to express text in a Label. Its properties are:

**code**: A required string. Code values shall be obtained from the IANA Language Subtag Registry (<http://ww.iana.org/assignments/language-subtag-registry>). The codes in this registry are obtained from the composition of language codes taken primarily from ISO 639-1 and ISO 639-2 and country codes taken from ISO 3166-1. The following resource is helpful for understanding these tags' composition: <https://www.w3.org/international/articles/language-tags/index.en>

**description**: a required string with a human-friendly description of the corresponding code value (e.g., “Danish”, or “Portuguese-Brazil”).

### 8.5.10 ModelScope

This class enables specifying what classes of the underlying data model a ContextItemDefinition is in context of. For example, a number used to identify farms in a particular jurisdiction would be tagged with code(s) representing the Farm class. The properties of ModelScope are:

**code:** A code that represents a class, in the context of a data model. For example, ISO11783\_FRM could represent the ISO 11783-10 schema’s Farm (FRM) class, as would ADAPT\_FARM, albeit within the ADAPT data model.

**description:** A human-readable description of the item, e.g., “ISOXML FRM element”, or “ADAPT Farm class”.

This class has an association, named **category**, with the ModelScopeCategoryEnum enumeration, which specifies what vocabulary a particular ModelScope code value comes from. This version of the standard includes three literals:

* ISO\_11783: Refers to classes defined in ISO 11783-10.
* ADAPT: Refers to classes defined by the ADAPT team (adaptframework.org)
* ISO\_TC\_347: Refers to classes defined in ISO TC 347’s upcoming reference architecture for data-driven agrifood systems.

### 8.5.11 Label

This class enables associating text labels in specific languages to objects in specific geopolitical contexts. For example, a ContextItemDefinition that represents a farm identifier specific to Canada could have names in multiple languages (e.g., English and French), and have labels in each. The properties of Label are:

**text:** The text of the label.

This class has two associations, both optional: one named **language** with an instance of the Language class, and the other named **geopoliticalContext** with an instance of the GeopoliticalContext class.

In summary, a Label object contains text, a reference to the language the text is in, and reference to a GeoPoliticalContext object that describes “where” that label can be used. The Internet Assigned Numbers Authority (IANA) Language Subtag Registry shall be the controlled vocabulary for languages.

### 8.5.10 Implementation Examples

Three examples are provided below which illustrate the various uses of ContextItem.

#### 8.5.10.1 Simple integer and string data entry ContextItemDefinitions

Table 9 below shows the relevant properties of an example ContextItemDefinition representing a USDA Farm Service Agency (FSA) Farm Number, an identifier very frequently used by producers in the United States. Comments:

* It has an arbitrary ID (ReferenceId) that is simply meant to be unique within the set of ContextItemDefinitions.
* It also has a Code, shown as “US\_FSA\_FN”, which represents the FSA Farm Number value.
* There is a list of keywords.
* ModelScopeIds is a list of string codes; Table 9 shows them dereferenced into an object model class names, in brackets.

Similarly, the GPCIds list shows a dereferenced identifier for the United States. An FMIS in Belgium, for example, would know that the FSA Farm Number is not relevant to their geopolitical context.

**Table 9 – ContextItemDefinition representing a**  
**USDA Farm Service Agency (FSA) Farm Number**

|  |  |
| --- | --- |
| **Property** | **Value/Comment** |
| id | ReferenceId = 9c0ab03e-bd42-447e-923a-9c298f8f6f7f (Could be any string value) |
| code | US\_FSA\_FN |
| value Type | INTEGER |
| description | FSA Farm Number |
| modelScopeIds | ISO\_11783\_FRM, ISO\_11783\_PFD, ADAPT\_FARM, ADAPT\_CROPZONE |
| gPCIds | USA |

#### 8.5.10.2 Numeric ContextItemDefinitions for auxiliary purposes (e.g. Properties of ContextItemEnumItems)

Table 10 shows the relevant properties of an example ContextItemDefinition representing Latitude and Longitude values. The purpose of this kind of ContextItemDefinition is to provide the infrastructure for ContextItemEnumItems to express properties. This is done with the Latitude and Longitude ContextItems shown. Comments:

* They have (arbitrary, simply meant to be unique within the set of ContextItemDefinitions) ReferenceIds.
* They have unique Codes, shown as 107 and 108, which represent the Latitude and Longitude
* Presentation objects are not included, because these ContextItem definitions are not meant to be user-entered.

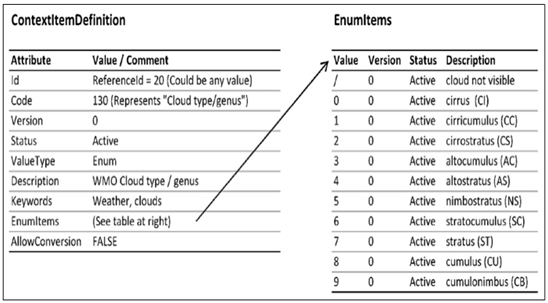
**Table 10–Two ContextItemDefinition examples, representing latitude and longitude values**

|  |  |  |
| --- | --- | --- |
| **Property** | **Value/Comment** | **Value/Comment** |
| id | ReferenceId = 13 (Could be any value) | ReferenceId = 14 (Could be any value) |
| code | 107 (Represents “latitude”) | 108 (Represents “longitude”) |
| value Type | DOUBLE | DOUBLE |
| description | Latitude | Longitude |
| keywords | Geographical coordinates, geodesy | Geographical coordinates, geodesy |
| default UoM | deg | deg |
| modelScopeIds | Farm, Field, Cropzone | Farm, Field, Cropzone |

#### 8.5.10.3 Enumerated ContextItemDefinition

Figure 13 shows the relevant properties of an example ContextItemDefinition representing a World Meteorological Organization (WMO) code for cloud type. Comments:

* It has an arbitrary ReferenceId (simply meant to be unique within the set of ContextItemDefinitions).
* It also has a Code, shown as 130, which represents the Cloud type / genus
* The ValueType is shown as “enum”. That implies that there must be a list of ContextItemEnumItems. There is no need for a Presentation object, since the user would be shown the ContextItemEnumItems’ Label strings corresponding to the user’s Language and GeoPoliticalContext (or, in their absence, the Description). What would “travel” with the ContextItem, however, would be the ContextItemEnumItems.



**Figure 13 – WMO cloud type/genus, an example of an enumerated ContextItemDefinition**

# 9 Reference, Setup and Configuration Data

Figure 14 illustrates the relationship among Reference, Setup and Configuration data. The categories are:

* **Reference Data** represents information, typically provided by a manufacturer, that is applicable to all instances of a thing (e.g., the series or model number of a kind of machine, or the formulation of a particular branded chemical product). Controlled vocabularies typically fall under this category.
* **Setup Data** represents basic information about a producer, farm, fields, and actors. This may include farm names, field boundaries, etc. Setup data is often also called *master data*.
* **Configuration Data** specifies the particular state of specific instances of things such as farm equipment and instruments (e.g. soil sensors, irrigation pivots, combines, etc.) This may include their location, what they are connected to, who installed them, etc.



**Figure 14 – Reference vs. setup vs. configuration data**

## 9.1 Reference Data

Reference data describes information that a manufacturer makes available for the purchase, setup and/or use of their products. It pertains to *all instances* of a manufacturer’s equipment and/or product and product components. Reference data is therefore not producer-specific or specific to an individual sale or single instance of a thing. For example, the product name, EPA number and active ingredients are reference data for a crop protection product, but a lot number is not. In another example, the model and series number are reference data for a centre pivot irrigation machine, but the serial number is not. The intent is to share reference data sets across the whole industry so that different stakeholders can interpret shared documents the same way. This includes names and identifiers of seed varieties, crop protection products, active ingredients, etc. .

## 9.2 Setup Data

Setup data, unlike reference data, is producer-specific. It provides information needed to set up data exchange between the producer and machinery or other actors (e.g., crop advisors), but it does not contain state information. Examples of Setup data include:

* Producer
* Farm
* Field
* Cropzone
* FieldBoundary
* Person
* PersonRole

### 9.2.1 Self-contained data vs. URLs

The Standard defines two ways of establishing a relationship between a dataset (of observations or operations data) and the corresponding setup and configuration data records:

* Self-contained: Reference data and Setup data are embedded within the dataset, and referred to using locally-scoped identifiers
* URL-referenced: Reference data and Setup data exist separately from the dataset, and are available in the standard format by dereferencing a uniform resource locator (URL) contained in the dataset.

### 9.2.2 Meaning of Producer

A central concept in the model is that of Producer. The Producer should not be interpreted as a specific individual *person* (the PartyRole class takes care of that, by assigning a role of PRODUCER to an instance of Party); instead, an instance of the Producer class should be interpreted as the underlying business entity.

Figure 15 shows the relationships among different objects corresponding to Setup (Producer-specific) Data. This is further broken out in Table 11.

As shown in Figure 15, the Producer class anchors many important concepts in the data model.



**Figure 15 – The Producer class and relationships**

### 9.2.3 Producer

This type represents the producer business entity that an irrigation system (4.11), resource, or document is associated with.

**id:** Required CompoundIdentifier. Includes a string used to refer to this producer in the scope of a document, along with a collection of unique ids that represent the producer in FMIS and other systems.

**description:** String, required. Human-readable name of the producer or business entity.

**contextItems:** Optional list of 0..\* ContextItem instances. These allow associating geopolitical-context-specific information to otherwise universal data types. For example, the Producer instance might contain a ContextItem instance that identifies the USDA Farm Service Agency identifier for this producer, as described in Clasue 8.5.10.1.

### 9.2.4 Party

This type represents a party, i.e., a person or organization, that a field operation or resource is associated with.

**id:** Required CompoundIdentifier. Includes a string used to refer to this party in the scope of a document, along with a collection of unique ids that represent the party in FMIS and other systems.

**firstName:** String, optional. First name of the person or organization; e.g., “James” in “James T Kirk.”

**middleName:** String, optional. Middle name or initial of the person or organization; e.g., “T” in “James T Kirk.”.

**lastName:** String, optional. Last name of the person or organization; e.g., “Kirk” in “James T Kirk.”

**combinedName:** String, required. Human-readable name of the person or organization.

**contextItems:** Optional list of 0..\* ContextItem instances. These allow associating geopolitical-context-specific information to otherwise universal data types.

### 9.2.5 PartyRole

This type represents a role carried out by a party (i.e., a person or organization).

**id**: Required CompoundIdentifier. Includes a string used to refer to this PartyRole in the scope of a document, along with a collection of unique ids that represent the PartyRole in FMIS and other systems.

**partyRef**: String, required. Corresponds to the referenceId of the party in question.

**role**: Required. Role being carried out by the party referenced above. This is an enumerated value from the PartyRoleEnum class (See 9.2.6 and Table 12).

**producerRef**: Optional string. Corresponds to the referenceId of the Producer business entity to which the PartyRole pertains. It accommodates use cases such as when a party (i.e., person or organization) is a crop advisor for one producer and an operator for another.

### 9.2.6 PartyRoleEnum

This describes possible roles a party (i.e., person or organization) may have, as shown in Table 12.

### 9.2.7 Farm

For the purposes of field operations, a farm is a collection of one or more fields that are grouped together functionally or administratively under a business entity (i.e., Producer). That relationship with Producer is optional, however, to maintain compatibility with the ISO11783-10 standard, where a Farm (FRM) can stand alone without linking to a Customer (CMR, equivalent to Producer). A producer can have multiple farms.

The Farm type represents a farm that the irrigation system or document is associated with. Note that, in turn, the farm may be associated with a producer.

**id**: Required CompoundIdentifier. Includes a string used to refer to this farm in the scope of a document, along with a collection of unique ids that represent the farm in FMIS and other systems.

**description:** String, required. Human-readable name of the farm.

**producerRef:** Optional shorthand string id corresponding to the reference identifier of the producer’s CompoundIdentifier Id.

**timeScopes:** Optional list of 0..\* TimeScope instances. Can be used to qualify the farm.

**contextItems:** Optional list of 0..\* ContextItem instances; allows associating geopolitical-context-specific information to a farm.

**Table 12 – Party roles**

|  |  |
| --- | --- |
| **Value** | **Annotation** |
| ASSET\_SUPPLIER | A person or organization that supplies or maintains mobile assets (e.g., tractors, sprayers, mobile irrigation systems) or stationary assets (e.g., grain bins, scales, stationary irrigation systems) used in crop or food production. |
| AUTHORIZER | Someone who authorizes the execution of a field operation. |
| CROP\_ADVISOR | Someone who provides Recommendations regarding how to grow a crop. |
| CUSTOMER | Someone who buys an agricultural commodity or product in the agricultural value chain. |
| CUSTOM\_SERVICE\_PROVIDER | A person or organization that provides field operations-related services (e.g., spraying, spreading, harvesting). |
| DATA\_SERVICES\_PROVIDER | A person or organization that provides data used to plan, prepare for, or execute a field operation. |
| END\_USER | Someone who will be consuming a harvested commodity, or a product thereof. |
| FARM\_MANAGER | A person or organization responsible for making management decisions regarding the growing of a crop on a piece of land. |
| FINANCIER | A person or organization that provides capital / funding associated with the production of a crop. |
| GOVERNMENTAL\_AGENCY | An agency of local, state/provincial, or national government, typically considered in a regulatory role. |
| INPUT\_SUPPLIER | A person or organization that supplies crop inputs (e.g., seed, fertilizers, and crop protection products) used to grow a crop. |
| INSURANCE\_AGENT | A person or organization that provides insurance services associated with the production of a crop. |
| IRRIGATION\_MANAGER | A person or organization that manages irrigation operations. |
| LABORER | Someone who harvests (typically manually) a commodity or performs other direct labor toward the production of a crop. |
| MARKET\_ADVISOR | A person or organization that provides advice to a producer regarding conditions of the market, typically for a harvested commodity |
| MARKET\_PROVIDER | A person or organization that can influence or create markets for a commodity (e.g., a wheat flour producers’ association) |
| NGO | A non-governmental organization. |
| OPERATOR | Someone who is in charge of performing a given field operation. |
| OWNER | A person or organization who owns the farm, field or cropzone on which a crop is being grown. |
| PRODUCER | The person or organization that is responsible for growing a crop. |
| TRANSPORTER | Someone who transports crop inputs or harvested commodities. |

### 9.2.8 Field

For the purposes of field operations, a field is an area of land that is managed under a business entity (i.e., Producer) and belongs to a Farm. Analogous to the situation with Farm, a Field is allowed to stand alone for the purposes of ISO11783-10 compatibility. For this same reason, a Field (analogous to an ISO11783-10 PartField (PFD)) can bypass a relationship with a Farm and point directly to a Producer (business entity). A Farm can have multiple Fields.

This type represents a field that the irrigation system or document is associated with. Note that, in turn, the field can be associated with a farm.

**id:** Required CompoundIdentifier. Includes a string used to refer to this field in the scope of a document, along with a collection of unique ids that represent the field in FMIS and other systems.

**description:** String. Required. Human-readable name of the field.

**producerRef:** Optional string id corresponding to the reference identifier of the producer’s CompoundIdentifier Id.

**farmRef:** Optional string id corresponding to the reference identifier of the farm’s CompoundIdentifier Id.

**timeScopes**: Optional list of 0..\* TimeScope instances that can be used to qualify the field.

**area**: An optional NumericValue instance which quantifies the Surface area of the field.

**activeBoundaryRef:** An optional reference to the FieldBoundary instance that defines the current or active boundary of the field.

**contextItems:** Optional list of 0..\* Contextitem instances. These allow associating geopolitical-context-specific information to otherwise universal data types.

### 9.2.9 FieldBoundary

This type represents the boundary of a field that the irrigation system or document is associated with.

**id:** Required CompoundIdentifier. Includes a string used to refer to this field boundary in the scope of a document, along with a collection of unique ids that represent the field boundary in FMIS and other systems.

**description:** String, required. Human-readable name of the field boundary.

**fieldRef:** Optional shorthand string id matching the reference identifier of the corresponding field’s CompoundIdentifier Id.

**timeScopes:** Optional list of 0..\* TimeScope instances that can be used to qualify the field boundary.

**spatialData:** The multi-polygon representing the field area. Note that the multi-polygon shape enables complex geometries, including holes, etc. When serializing this object to a text format, data shall be encoded in WKT format as described in ISO 19125-1, and are assumed to be in geographical coordinates, WGS 84 datum.

**gNSSSource** [0..1]: Code from GNSSSourceEnum.

**contextItems:** Optional list of 0..\* ContextItem instances. These allow associating geopolitical-context-specific information to otherwise universal data types.

### 9.2.10 Cropzone

A *Cropzone* is an area of land that is grown with a given crop during a given time interval. Cropzones exist in the context of fields and can mean different things in different situations. A field can have multiple Cropzones.

The properties of a CropZone are:

**id**: Required CompoundIdentifier. Includes a string used to refer to this Cropzone in the scope of a document, along with a collection of unique ids that represent the Cropzone in FMIS and other systems.

**description**: Human-readable name of the CropZone.

**fieldRef** [0..1]: String. Optional string id corresponding to the reference identifier of the parent field’s CompoundIdentifier Id.

**timeScopes** [0..\*]: Optional list of TimeScope instances that can be used to qualify the temporal relevance of a CropZone instance.

**arableArea** [0..1]: Surface area of the CropZone. Uses the NumericValue type which includes the units of measure.

**boundingRegion** [0..1]: The CropZone boundary, a MultiPolygon. When serializing this object to a text format, data shall be encoded in WKT format as described in ISO 19125-1, and are assumed to be in geographical coordinates, WGS 84 datum.

**gNSSSource** [0..1]: Code from GNSSSourceEnum.

**contextItems** [0..\*]: Optional list of ContextItem instances. These allow associating geopolitical-context-specific information to otherwise universal data types.

Three examples follow:

* 1. A field is grown in a rotation of winter wheat and soybeans in a given year. The field is divided into two Cropzones. Both have the same spatial footprint, but each corresponds to a different crop (different crop, same spatial footprint, different time), and each contains a TimeScope instance defining when the wheat or soybeans were in production.
  2. Two different crops or crop varieties are intercropped (i.e., grown simultaneously) on the same field in response to soil type variability or a similar criterion. The field can be split spatially into two Cropzones, each corresponding to the spatial footprint of one of the varieties. (Different crop, different spatial footprint, same time).
  3. In a given year, a large citrus grove is split geographically into multiple rectangular blocks for regulatory, crop protection, and harvest management purposes. The grove is equivalent to a Field, and each block is a Cropzone. The same grove may be split into a different arrangement of blocks the following year. (Same crop, different spatial footprint, same time).

NOTE Whereas a Field will typically persist over time, CropZones must exist in the context of a specified time interval, and always represent **one** crop. See examples of CropZones in Table 13.

**Table 13 – Examples of multiple CropZones and their use cases**

|  |  |  |  |
| --- | --- | --- | --- |
| **Crop** | **Spatial Extent** | **Time** | **Example Use case** |
| Same | Same | Same | Improbable use case. Can represent a landlord/tenant arrangement, where different landlords are allocated different fractions of the production of the whole field. |
| Same | Same | Different | Multiple horticultural crops per season in a warm climate. |
| Same | Different | Same | Different management or traceability needs require separate record-keeping. This can reflect a setting where parts of a field have different customers, or a blocked arrangement on a citrus grove or vineyard. |
| Same | Different | Different | Successive plantings of the same crop but with different spatial footprints (e.g. due to different guidance patterns during planting). |
| Different | Same | Same | Can correspond to a tightly intercropped pair of crops or varieties, where it is not practical to separate them geographically. |
| Different | Same | Different | Crop rotation.  Note that the timeframes do not have to be mutually exclusive. For example, a cover crop can be sown before the preceding crop is harvested. The cover crop would correspond to a different cropzone, as the Standard defines a cropzone as a spatial region where a given crop is being sown during a given time interval. |
| Different | Different | Same | Block pattern in a field growing different horticultural crops in a single crop (short) growing season. |
| Different | Different | Different | Block pattern in a field growing different horticultural crops in a multiple crop (long) growing season. |

### 9.2.11 ConfigurationDataPedigreeEnum

Setup data can change over time. Different data providers manage this change in different ways; this enumeration in Table 14 lists the different possible strategies. Users will need to act accordingly, as the accuracy of historical records in the context of a LATEST\_ONLY strategy, for example, would suffer unless the user were keeping copies of successive versions of the setup.

Note: This class is used in other parts of the standard.

**Table 14 – Setup data pedigree**

|  |  |
| --- | --- |
| **Value** | **Annotation** |
| LATEST\_ONLY | Indicates a situation where the data provider only keeps the most recent system setup record. In these situations, it is incumbent upon the user to access and store the latest setup data frequently, so as to not introduce uncertainty regarding what version of the setup data record corresponds to any given downloaded historical record. |
| MATCHED\_BY\_TIME\_INTERVAL | Indicates a situation where the data provider keeps a time series of setup records, with timestamps such that data records can subsequently be sourced referencing the corresponding setup record. This eliminates the possibility of error whereby a user also cites a data record with the incorrect setup record. |
| UNKNOWN | Indicates a situation where the data provider’s strategy regarding versioning of system setup records is unclear, unknown, or unspecified. In these situations, users should act, as in the case of LatestOnly; i.e., access and store the latest setup data frequently, so as to not introduce uncertainty regarding what version of the setup data record corresponds to any given downloaded historical record. |

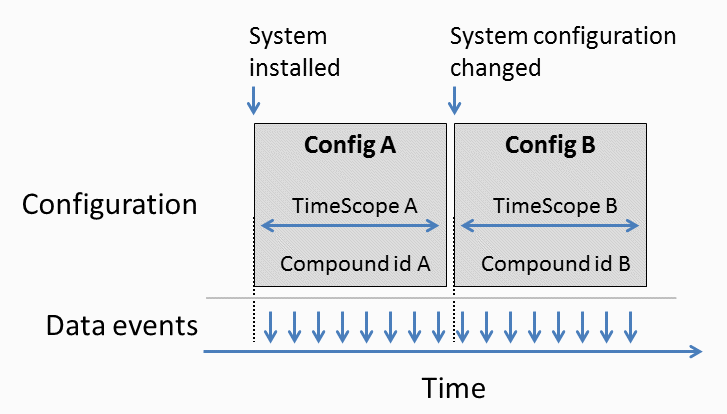
## 9.3 Configuration Data

Configuration Data specifies the particular state of instances of things such as farm equipment and instruments (e.g. soil sensors, irrigation pivots, combines). This may include their location, what they are connected to, who installed them, etc.

The material that follows is common to configuration data for both operations and observations. Specific information about Data Source Configuration and (Irrigation) System Configuration is presented in Parts 2 and 3 of this standard, respectively. The constraints specified in 9.3.2 apply to all producers and consumers of configuration data.

### 9.3.1 Rationale for configuration management

Since configuration data is state-dependent, a new instance of a configuration data object is needed when the state of the object it describes changes. Figure 16 shows this. When the system is originally installed and configured, an instance of a System Configuration object called ConfigA is created. This ConfigA object provides context to data events (e.g., as-applied data from an irrigation system, or measured values from an instrument), such as units of measure, installation location and other state-specific information. If some time later the system configuration is changed (for example, through a change in reported unit of measure, or a physical change in the system); a new instance of a system configuration object, ConfigB, would be necessary to provide context to subsequent data events.



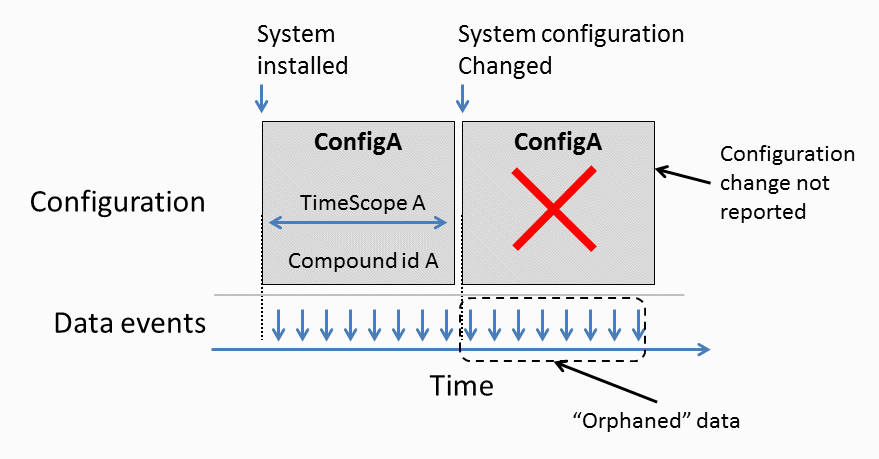
**Figure 16 – Timeline showing a proper succession of data events, changes in system configuration, and associated events.**

Proper interpretation of reported data requires being able to unambiguously understand the configuration of the system that is producing the data. Two examples follow:

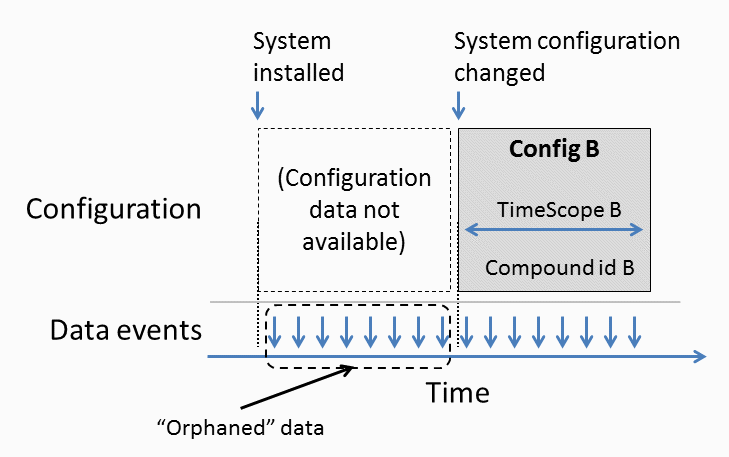
EXAMPLE 1 A data logging device, or data logger, reports air temperature data every 15 minutes. When the system was originally installed, the data logger was configured to express temperature values in degrees Celsius. Some time later the configuration is changed so the values are now expressed in degrees Fahrenheit. If the consumer of the data is not made aware of the change in system configuration, erroneous interpretation of the data following the configuration date may result.

EXAMPLE 2 A centre pivot irrigation system is originally configured with 5 spans of 60.8 meters (200 ft) each on a square field. Following the requirements in ISO 7673-3, the data provider configures the system with 5 SectionFlow instances, as described Clause 7.1.5 of ISO 7673-3. Some time later, the producer adds an endgun to irrigate the corners of the field. If the consumer of the data is not made aware of the change in system length, and is using scenario 3 shown in Table 3 of ISO 7673-3, erroneous interpretation of data following the configuration change will result. This error will result because, in scenario 3, the data consumer is responsible for quantifying throw distance and depth of application.

This is managed in a straightforward way: system configuration objects are required to have a unique identifier that can be associated with the reported data values.



**Figure 17 – Data is misinterpreted because the configuration change is not reported. This is the scenario represented by Example 1 and 2 above.**



**Figure 18 – Data is misinterpreted because the original configuration data is not available. This is the scenario corresponding to LATEST\_ONLY in Table 14.**

### 9.3.2 Guiding principles for configuration management

The ISO 7673 series applies a maxim presented as the Principle of Robustness in RFC 761, a foundational document of the Internet: “be conservative in what you do, be liberal in what you accept from others.” In that spirit, the following constraints apply to all producers and consumers of configuration data:

1. Producers of configuration data shall maintain the time series of changes in said configuration data, making it possible to clearly associate any given moment in time with the corresponding configuration data records. Note: This can be done explicitly by versioning the configuration data if it is kept apart from the operations / observations data they qualify, or implicitly if the configuration data enriches the observations / operations data immediately upon the production of the same and is stored along with them. The implicit case requires that the configuration data and the observations / operations data are produced by the same party; the explicit case does not.
2. Data producers and data consumers shall provide documentation that clearly and unambiguously determines who is responsible for communicating configuration changes.
3. Configuration changes shall be communicated as soon as possible. The qualification “as soon as possible” shall be appropriate for the context of the data exchange (i.e., dependent on the software and telecommunication technologies used for the data exchange).
4. Configuration data producers shall provide some mechanism for data consumers to request configuration data. This request may be implicit in a data query in the implicit scenario of point 1 above; i.e., when the configuration data and observations / operations data producers are the same party.
5. Configuration data producers may provide a mechanism for notifying, or sending configuration changes to, data consumers without an explicit request from a particular consumer.
6. When a configuration data producer has any uncertainty about whether a data consumer was or was not informed about a particular configuration change, the configuration data producer shall assume that the data consumer was not informed of such change.
7. When a data consumer has any uncertainty about whether configuration data is correct, the data consumer shall be responsible for obtaining updated configuration data.
8. When a system makes computations or estimates using received data, the receiving system is responsible for using appropriate configuration data.
9. If configuration data must be exchanged separately from the observations/operations data, the data consumer and data producers shall document the expectations placed on both sender and recipient prior to engaging in data exchange.

# Bibliography

[1] AgGateway. (2021). *UnitSystem.xml*. Retrieved from <https://github.com/ADAPT/ADAPT/blob/develop/source/Representation/Resources/UnitSystem.xml> on 31 Aug 2023.

[2] EPSG “WGS 84.” (<https://epsg.org/ellipsoid_7030/WGS-84.html>) Accessed February 19, 2024.

[3] ISO 11783-10, Tractors and machinery for agriculture and forestry — Serial control and communications data network — Part 10: Task controller and management information system data interchange

[4] ISO 22006:2009, Quality management systems — Guidelines for the Application of ISO 9001:2008 to crop production

[5] ISO/IEC 9834-8, Information technology ― Procedures for the operation of object identifier registration authorities ― Part 8: Generation of universally unique identifiers (UUIDs) and their use in object identifiers

[6] ISO/IEC 19510:2013, Information technology ― Object Management Group Business Process Model and Notation

[7] QUDT (<http://www.qudt.org/>) Referenced 25 Feb 2024.

[8] Unified Code for Units of Measure, 2017, (<https://ucum.org/ucum>) Referenced 25 Feb 2024.

[9] UN/CEFACT, 2017. *Recommendation 20* (<https://unece.org/trade/uncefact/cl-recommendations>) Referenced 25 Feb 2024.

[10] WMO No. 8. Guide to Meteorological Instruments and Methods of Observation, 7th Edition. 2008. Geneva