SOIL CRC

Performance through collaboration

MILESTONE REPORT

Project 2.3.002

Visualising Australasia's Soils: Systems Architectures

Milestone 14.3

VAS Phase 2 Systems Architectures 2024

VISUALISING AUSTRALASIA'S SOILS: PHASE 2 SYSTEMS ARCHITECTURES

CONTENTS

AUTHORS

Andrew MacLeod, Heath Gillett, Bruce Simons, Megan Wong, Alicia Reynolds, Cedar Martin, Richard Archer and Peter Dahlhaus.

KEYWORDS

Soil database, Soil data information model, Soil data vocabularies, Soil data schema

CITATION

MacLeod A., Gillett H., Simons B., Wong M., Reynolds A., Martin C., Archer R. & Dahlhaus P. (2024). Visualising Australasia's Soils: Phase 2 System Architectures. Milestone Report 14.3. Soil CRC Project 2.3.002. Cooperative Research Centre for High Performance Soils, Callaghan, NSW.

1. INTRODUCTION

The Visualising Australasia's Soil (VAS) project, being undertaken through the Cooperative Research Centre for High Performance Soils (Soil CRC), aspires to create a research data federation, based on agreed governance and data-stewardship frameworks, that allows relevant soil data from the private and public sectors to be discoverable through intuitive-to-use internet portals. One of the principal aims is to motivate Australasian soils data custodians to make their data Findable, Accessible, Interoperable, and Reusable (FAIR), thus providing a range of benefits for research, on-farm decision making and policy development.

The VAS project evolved over the past five years into three distinct phases of development:

- Phase 1: Visualising Australasia's Soils: a Soil CRC interoperable spatial knowledge system.
- Phase 2: Visualising Australasia's Soils: extending the soil data federation.
- Phase 3: Visualising Australasia's Soils: building a legacy.

Phase One implemented most of the technical systems architecture which involved the co-design, development, and implementation of the VAS soil data information model. The chosen model conforms to international data exchange standards and can aggregate a variety of soil (and other) data, value adding to the data by making it FAIR. The information model is fully documented in Appendix B of Dahlhaus et al. (2021) and is publicly available[1].

The second main component was building and implementing the soil data aggregator and VAS web portal as the public-facing component of the project. The public view has a selection of the openly available soil data, and the login allows VAS project participants to view their own data. Some functionality to select and filter data, graph data, and download data were included.

Phase Two has upgraded the technical architecture by largely implementing system improvements that were required to support the current VAS users and new requirements for data interoperability with the Australian National Soil Information System (ANSIS), as well as maintain the currency of the system (technology updates, bug fixes, etc.). The main changes were alterations to the way that data is delivered to other systems (i.e., using application programming interfaces – APIs), improvements to the self-serve system, and improved data reporting. These methods are fully documented in the following sections of this report.

1 https://soilcrc.com.au/technical-reports/visualising-australasias-soils/

2. BACKGROUND

The usual approach when establishing a database is to design it to meet specific user requirements via use-cases, mission statements or such like. Similarly, when designing a data structure to exchange domain specific data, a domain-specific model is usually created. Domain data exchange examples include geology (GeoSciML), mining and exploration (EarthResourceML), groundwater (GroundWaterML2), soils (ANZSoilML), and water and energy supply and consumption (WESCML). In these database designs and models all the features/objects that interest the domain are specified, along with the properties that describe these features, and the relationships and relationship properties that exist between these features. In a relational database, these are hypostasised as tables and columns.

This 'hard-type' approach to design presents a number of challenges. These may include finding the right people with the skills and resources, and specifying scope given most domains have overlap with other domains (e.g., Boreholes). In addition, the resulting domain model, if it accurately reflects the real world, is usually complicated, making it difficult to use by both domain experts and non-domain users.

An alternative is to specify the domain-specific features and relationships between them, but 'soft-type' the properties. That is, populate the features (classes) with attribute:value pairs. An example of this design is the soil/terrain model SoTerML. This approach is more readily extendable, but requires a vocabulary of appropriate attributes for each class. It moves the problem from the data-structure (schema) design to specifying vocabulary or look-up tables.

VAS took this last approach, but to wrap it up in a generic 'Observation' framework. So instead of specifying what all the properties any feature, such as a Soil or Soil Horizon or Landform, may have, VAS collects all the observations, measurements, and interpretations that have been made on these features. The domain-specific feature-types that may be stored in the database are not specified as tables, but rather as a 'feature-type' property that comes from a vocabulary listing. This provides the flexibility of adding new feature-types without any database schema changes. It also means that once the Observation and Measurements (O&M) model schema is understood, the data structure is understood for all the domains where observations and measurements are made.

The 'Observation framework' chosen was based on O&M version 2 (see 'CeRDI Observations System Overview' below), although it is also considered to be compatible with the more recent O&M version 3, and the W3C derivative, 'Sensor-Observation-Sampling-Actuator' (SOSA). Instead of identifying domain-specific features and properties, O&M specifies what features and properties are common to all Observations. For example, one of these O&M features is the "FeatureOfInterest" (FoI). The FoI can be anything about which observations are made, and its type identified via its 'featureType' property, which can come from a vocabulary list.

An O&M property is the 'ObservableProperty', the FoI property that is being observed or measured. Again these Observable Properties do not need to be specified in the database design, but can come from a vocabulary list.

Knowing which 'ObservabelProperties' can be associated with which 'featureTypes' is normally the remit of the domain experts during the database/exchange language design phase. However, VAS has taken a different (and new and previously untried) approach.

All the observations that have been stored in VAS have been made by experts in their domain. So, for example, all the soil-layer laboratory results or the soil-profile descriptions have been made either by soil scientists, or on observed properties specified by soil scientists. So all the 'ObservableProperties' that have been associated with the same 'featureType' (such as a 'Soil Layer') represent a collection of the attributes of that feature. If observers use an inappropriate Observable Property, such as 'happiness quotient' for a Soil Horizon, there is nothing built into VAS to prevent this. However, it is unlikely to occur since all the observations and measurements are provided by domain experts. Even if the 'wrong' observed properties are entered, it is unlikely to influence the data usage, as it is unlikely to be discovered as part of a search, and, if found, the expert user can choose to treat it as bad data if it does not fit their concept of an appropriate Soil Horizon property.

Importantly, this list of Observable Properties is not unique to a specific domain (e.g., 'density' or 'Nitrogen concentration' or 'pH' may be 'ObservableProperties' of features from many domains). So a single list of 'ObservableProperties' can be maintained, rather than a list specific to each domain.

Further, the observation result values that have been assigned to these attributes identify that attribute's appropriate vocabulary. For example, terms such as 'Loam', or 'Loamy sand' specify the vocabulary terms that can be associated with the soil texture observable property, but these terms would not have been associated with other observable properties, such as coarse fragment shape.

VAS captures all the results, observable properties, and procedures that have been made for each of the soil feature types (such as Soil Body, Soil Layer, Soil Horizon, Soil Profile). If VAS captures sufficient breadth of Observations (including Measurements and Interpretations), then it is possible to identify all those properties usually associated with each feature, the procedures used for generating the result values for each property, and the range of appropriate results for each procedure for each observable property. As the number and range of observations increases, this model should approach a model generated from the traditional domain-expert process.

3. CERDI OBSERVATIONS SYSTEM OVERVIEW

The primary purpose of the observations system is to collect and store observation and measurement data and publish this data for researchers, industry, government, and the public. It uses the ISO19156[2] and Open Geospatial Consortium (OGC)[3] Observations and Measurements (O&M) conceptual model to store field and laboratory environmental data in a domain independent structure. The observation model states:

“An **Observation** is an action whose **result** is an estimate of the value of some **property** of the **feature-of-interest**, obtained using a specified **procedure**” (SJD Cox) The key insights are to:

- separate the *observation act* from the procedure, which may be used for other observations.
- separate the *feature-of-interest,* which has many properties, the values of each of which may be estimated more than once, at different times or using different procedures.
- recognise that the outcome of an Observation is a *result* – the value of which constitutes an estimate of a value of a property, and this may be a value or range of values if a measurement, a term, a term range or a description is an assertion

In addition to standardising the data structure, the system makes use of existing domain-specific controlled vocabularies and ontologies to standardise the semantic content.

The system is a custom development by CeRDI, primarily using PHP and PostgreSQL, services are provided via PostGrest.

[2] Cox, Simon Jonathan David (2011). "ISO 19156:2011 Geographic information – Observations and measurements". International Organization for Standardization. doi:10.13140/2.1.1142.3042

[3] "OGC Abstract Specification Topic 20: Observations and measurements". 2010. Retrieved 2010-11-22.

4. VAS CONCEPTUAL OVERVIEW

The VAS technical system is illustrated below (Figure 4.1).

The VAS soil research data is aggregated via the CeRDI Observations System as described in the previous section. The data aggregator allows participants to log into a self-serve system to submit a dataset using a template that collects metadata and the soil data observations. Educational materials in the form of short 'how-to' videos are provided as instructions. These data are then manually checked and mapped into the aggregator using the standard schema.

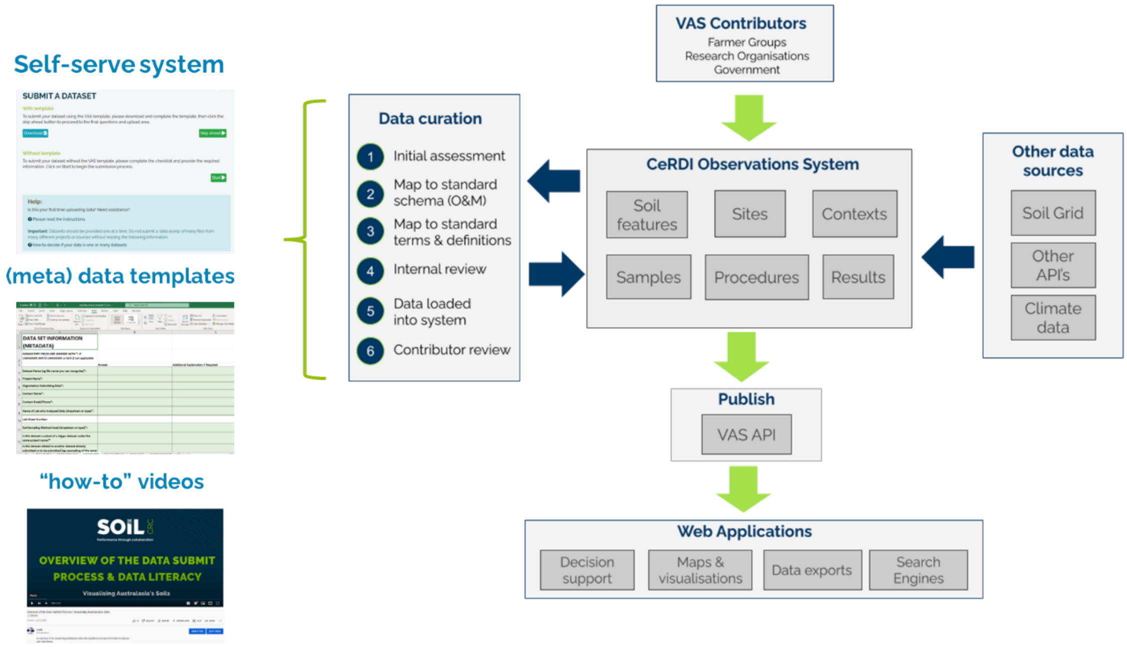

Figure 4.1. The VAS technical system overview.

Soil and ancillary data from other sources that interoperably serve standard datasets, such as ANSIS, New Zealand Soil Classification, Soil and Landscape Grid of Australia (SLGA), or the Bureau of Meteorology, can be interoperably consumed.

The data is published, subject to the data custodians consent, via an API to the VAS portal, other systems, or users.

Future work will include improving the self-serve system to include a data management system to give the data custodians more control, and make the input metadata mapping as seamless as possible.

5. DATABASE SCHEMA OVERVIEW

5.1 PART 1: OBSERVATIONS, PROCEDURES, SPECIMENS, SAMPLING FEATURES, FEATURES OF INTEREST

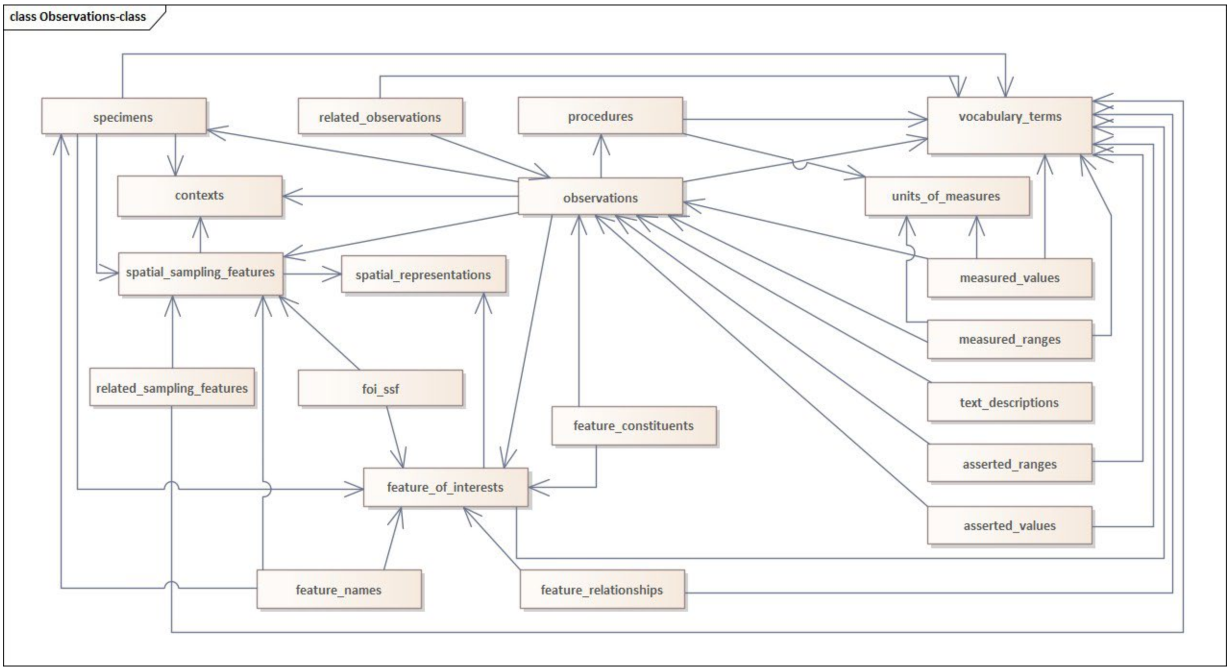

Figure 5.1 Diagram: Database Tables specifically relating to Observations

The 'observations', 'procedures', and 'features_of_interests' database tables relate directly to concepts described in O&M. Specimens (for storing sample data) are also taken directly from the O&M model. Additional tables allow capturing relationships and names of these features. Results are stored depending on the type of result: measured values and ranges (with links to associated units of measure), asserted values (terms) and ranges (terms for upper and lower values), and text descriptions.

The 'spatial_sampling_feature' table stores information about the sampling feature, that is, the human construct that was used to sample the real-world feature whose property is being measured.

The 'vocabulary_terms' table stores required terms (see 'Part 3: Auxiliary Tables: Vocabulary Terms, Units of Measure, Feature Names, Reports'). These may be terms from traditional look-up tables such as terms associated with 'laboratory methods' (e.g., 'pH using 1.5 HCl') or 'drainage' (e.g., 'well-drained', 'poorly drained'). They may also be the terms that are traditionally the look-up table names and appear as database columns (e.g., 'laboratory method', 'pH', 'P concentration'). Vocabularies in the 'vocabulary_terms' table are also used where terms are required elsewhere in the database, such as specifying the 'role' organisations and users play in projects and the type of organisation.

Importantly, in addition to a set of terms the table indicates what schema the term comes from (i.e., what the look-up table name would be that the term came from if from a look-up table), a description, and a persistent identifier to an external resource that contains more information (e.g., images, other relationships, alternative labels or languages).

A separate 'units_of_measures' table (see ‘Part 3: Auxiliary Tables: Vocabulary Terms, Units of Measure, Feature Names, Reports’) is used for the special set of vocabularies with additional properties associated with units of measure, where, in addition to alternative names (e.g., ‘metre’ and ‘meter’) and notation (e.g., ‘m’, ‘cm’), specifying what quantity kind the unit is used for (e.g. ‘length’) is required. The unit of measure identifier provides a link to an external ontology maintained by an appropriate authority.

5.2 OBSERVATIONS

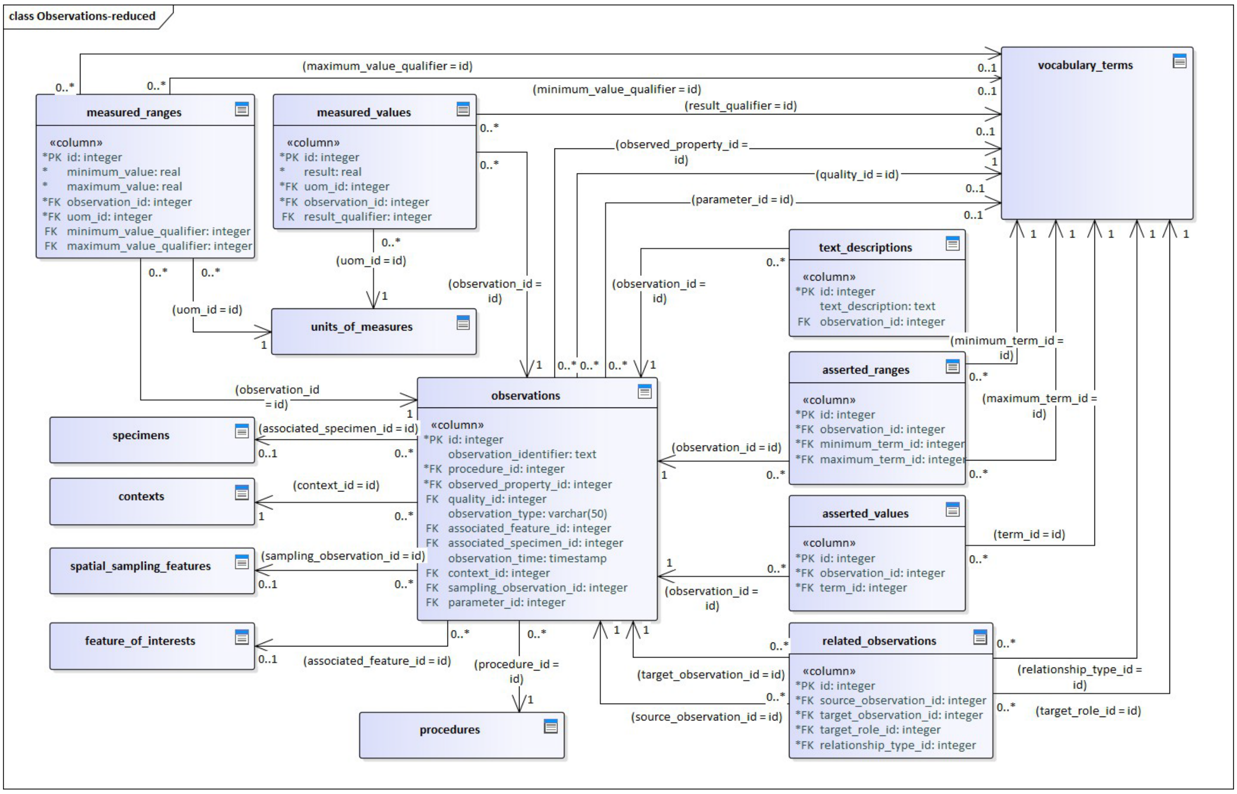

Figure 5.2. ‘observations’ and the results tables. Columns of related tables and associated relationships are not shown

5.2.1 ‘observations’ table description

The **observations** table records information about individual observations (measurement or assertions) made on the soil feature.

1. The type of Observation (e.g., ‘Laboratory Measurement’, ‘Field Observation’). This could potentially be expanded to include ‘Application Rate’, ‘Crop Yield’, etc.
2. The result of the Observation could be a numerical range, single numerical result, a range of terms or a single term. This could be expanded to include raster, time series, image results.

COLUMN NAME	DATA TYPE	NOT NULL	COMMENTS
id	integer	True	The database id
observation_identifier	text	False	The persistent identifier of the individual Observation (URI)
procedure_id	integer	True	The foreign key to the procedures table identifying the procedure used to generate the observation
observed_property_id	integer	True	Foreign key to the vocabulary_terms table identifying the observed property being measured (e.g. "K concentration", "pH').
quality_id	integer	False	Foreign key to the vocabulary_terms table to describe the quality of the observation.
observation_type	varchar(50)	False	The type of Observation (e.g. "Laboratory Measurement", "Field Observation"). This could be expanded to include "Application Rate", "Crop Yield", etc. Future upgrade to a Foreign key to the vocabulary_terms table
associated_feature_id	integer	False	Foreign key to the soil feature the observations were made on (the real world FeatureOfInterest such as a Soil Body, Soil Horizon, Soil Profile, Soil Layer).
associated_specimen_id	integer	False	The foreign key to the (Soil) Specimen that the observation was made on.
observation_time	timestamp	False	The time and date of the observation. The O&M model allows for additional time properties, such as the time the observation was applicable, the time the observation was made, the time the observation is valid for. These variations have not been included here.
context_id	integer	False	A foreign key to the contexts table specifying the context in which the observation was made. Usually a job or set of tasks undertaken by a single observer/machine as part of a broader project. Links Job, Project, Client information
sampling_observation_id	integer	False	A foreign key to the feature that was used to sample the real world environmental feature such as SoilProfile, or Specimen that the observation was made on. SpatialSamplingFeatures may be "Plot", "Site", "Station", "Paddock", etc.

parameter_id	integer	False	Describes an arbitrary event-specific parameter. This might be an environmental parameter, an instrument setting or input, or an event-specific sampling parameter that is not tightly bound to either the feature-of-interest or to the observation procedure. In some contexts the *Observation::procedure* is a generic or standard procedure, rather than an event-specific process. In this context, parameters bound to the observation act, such as instrument settings, calibrations or inputs, local position, detection limits, asset identifer, operator, may augment the description of a standard procedure. EXAMPLE The fraction of the soil ('coarse fraction', 'fine fraction', 'whole soil') from which the particle size proportion was taken.

5.2.2 **'related_observations' table description**

The **related_observations** table records information about relationships between individual observations.

In addition to identifying observations that were made as part of a project, for example, this table could be used to relate previous application or yield results to current results.

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
id	integer	True	The database id
source_observation_id	integer	True	The foreign key to the observation that is the source in the relationship
target_observation_id	integer	True	The foreign key to the Observation that is the target in the relationship
target_role_id	integer	True	The foreign key to the vocabulary_terms table that specifies the role that the target Observation plays in the relationship (e.g. "previous application rate", "related observation", "alternative result").
relationship_type_id	integer	True	The foreign key to the vocabulary_terms table that specifies the type of relationship between the two Observations

The results of an observation may be one of five types:

1. A numeric value resulting from a measurement (**measured_values**)
2. A numeric range resulting from a set of measurements **(measured_ranges**)
3. A term value resulting from an asserted value (**asserted_values**)
4. A range of term values resulting from a set of asserted values (**asserted_ranges**)
5. A description (**text_descriptions**).

Depending on the type of result, these are stored in separate tables.

5.2.2.1 'measured_values' table description

The 'measured_values' table is used to store empirical results from laboratories, field measuring devices, etc., where the result is a single numeric value.

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
id	Integer	True	The database id
result	Real	True	The numerical result of the Observation
uom_id	Integer	True	The Unit Of Measure for the result. This is a foreign key to the units_of_measures Table.
observation_id	Integer	True	The Observation that the result relates to.
result_qualifier	Integer	False	Foreign key to the vocabulary_terms table. Qualifies the result. Examples are '<', '<=', '>', '>=', '~', 'below detection limit', 'above detection limit'. Also to capture the gml:NilReasonTypes ('unknown', 'withheld', 'inapplicable', 'missing', 'template').

5.2.2.2 'measured_ranges' table description

The 'measured_ranges' table is used to store empirical results from laboratories, field measuring devices, etc., where the result consists of a minimum and maximum numeric value each with an optional qualifier.

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
id	integer	True	The database id for the measured range
minimum_value	real	True	The smallest numerical value in the range
maximum_value	real	True	The largest numerical value in the range
observation_id	integer	True	The foreign key to the Observation which created this result
uom_id	integer	True	A foreign key to the units_of_measures table for the Unit Of Measure
minimum_value_qualifier	integer	False	Foreign key to the vocabulary_terms table. Qualifies the minimum value result. Examples are '<', '<=', '>', '>=', '~', 'below detection limit', 'above detection limit'. Also to capture the gml:NilReasonTypes ('unknown', 'withheld', 'inapplicable', 'missing', 'template').
maximum_value_qualifier	integer	False	Foreign key to the vocabulary_terms table. Qualifies the maximum value result. Examples are '<', '<=', '>', '>=', '~', 'below detection limit', 'above detection limit'. Also to capture the gml:NilReasonTypes ('unknown', 'withheld', 'inapplicable', 'missing', 'template').

5.2.2.3 'asserted_values' table description

The 'asserted_values' table is used to store the result of an observation that is a term from a vocabulary.

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
id	integer	True	The database id
observation_id	integer	True	The foreign key to the Observation that generated the term result
term_id	integer	True	The foreign key to the vocabulary_terms table

5.2.2.4 'asserted_ranges' table description

The 'asserted_ranges' table is used to store the result of an observation where the results are a lower term and an upper term.

COLUMN NAME	DATATYPE	NOT NULL	DESCRIPTION
id	integer	True	The database id
observation_id	integer	True	The foreign key to the Observation that generated the term ranges
minimum_term_id	integer	True	The foreign key to the vocabulary_terms table for the minimum term (e.g. "poorly drained").
maximum_term_id	integer	True	The foreign key to the vocabulary_terms table for the maximum term (e.g. "well-drained").

5.2.2.5 'text_descriptions' table description

The 'text_description' table is used to store the result of an observation where the result is a free text description.

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
id	integer	True	database id
text_description	text	False	The free text description result
observation_id	integer	False	The foreign key to the Observation that generated the free text description.

5.3 PROCEDURES

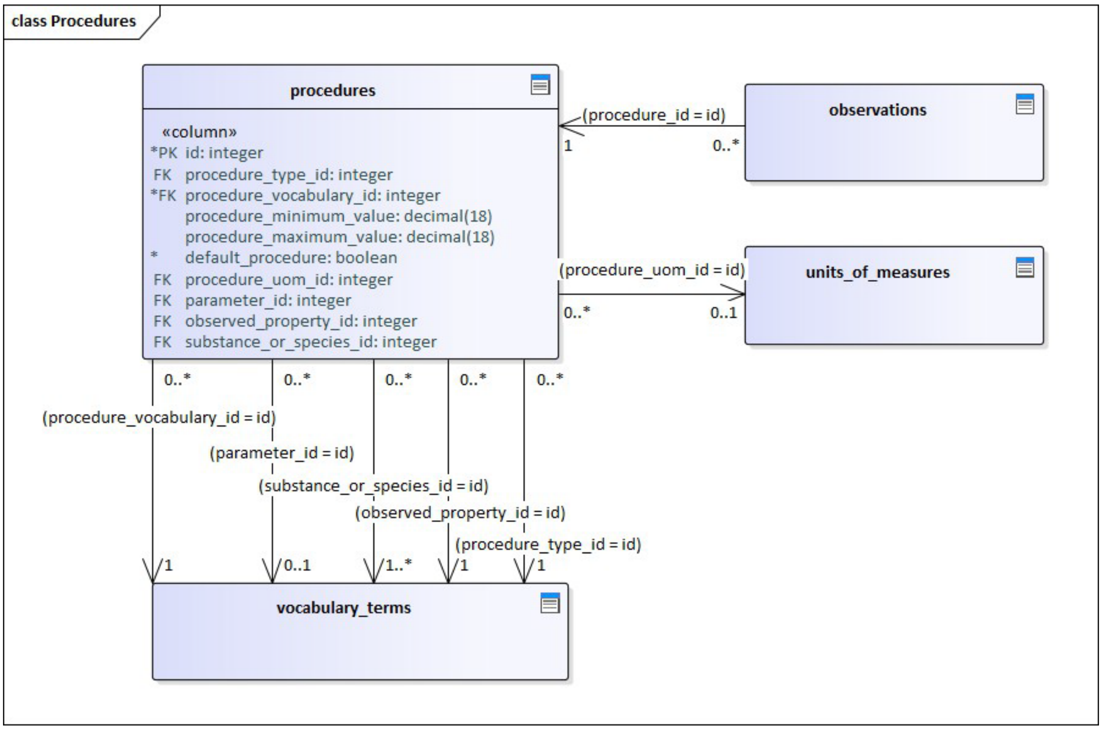

Figure 5.3 'procedures' table. Columns of related tables and associated relationships are not shown

5.3.1 'procedures' table description

This table records information about the procedures or methods used to record the observation. A challenge with recording procedures is that they sometimes need to be treated as a series of activities or steps, sometimes as a term with additional procedure specific information, and sometimes just as a term from a vocabulary. The former can cater for these variations, but comes with added complexity, particularly during data entry. The current database design caters for the second approach by accommodating additional information, such as whether the procedure is the default method, what result unit of measure the procedure uses, and what the data range is.

The purpose of the 'procedures' table is to group the properties of units of measure, observed property, minimum and maximum values, and substance or species that apply to a certain procedure. From an observation perspective the 'procedures' table only provides a link via the procedure_id to the vocabulary term that identifies the procedure. The Units of Measure (UoM) for the actual observation are stored against the result. The observed property for the observation is stored against the observation.

The downside of this is that there is no guarantee that the UoM and observed property associated with the selected procedure will match the UoM stored against the result or the observed property stored against the observation.

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
id	integer	True	The database id
procedure_type_id	integer	False	A foreign key to the vocabulary_terms table specifying the type of procedure ("measurement", "interpretation", "assertion", "calculation").
procedure_vocabulary_id	integer	True	A foreign key to the vocabulary_terms table that identifies the procedure used (e.g. "pH using 1:5 HCl")
procedure_minimum_value	decimal(18)	False	The minimum allowable value for the procedure if the result is expected to be a measurement.
procedure_maximum_value	decimal(18)	False	The maximum allowable value for the procedure if the result is expected to be a measurement.
default_procedure	boolean	True	Whether the Procedure is the default procedure for this Observed Property
procedure_uom_id	integer	False	A foreign key to the units_of_measures table corresponding to the unit of measure for the procedure
parameter_id	integer	False	A foreign key to the vocabulary_terms table to specify a procedure specific parameter term
observed_property_id	integer	False	A foreign key to the vocabulary_terms table that identifies the observed property being measured by the procedure (e.g. "soil texture")
substance_or_species_id	integer	False	A foreign key to the vocabulary_terms table of the term that identifies the object, substance or species that the property is related to (e.g. "Potassium", "Ammonia").

5.4 SPECIMENS

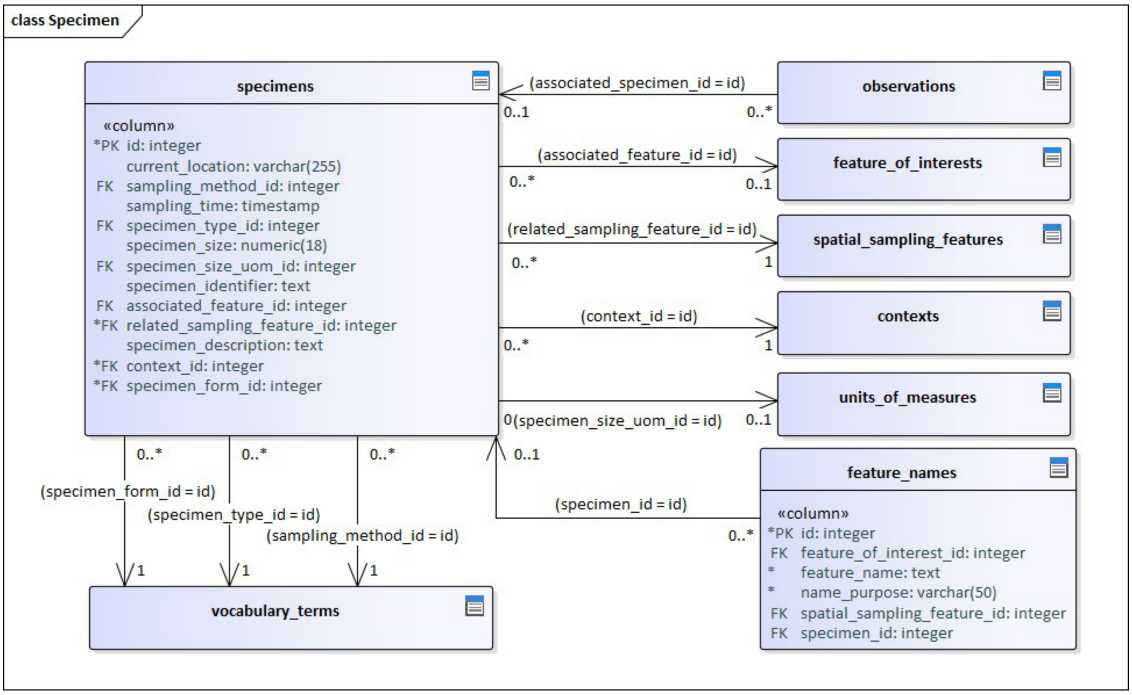

Figure 5.4 'specimens' table. Columns of related tables and associated relationships are not shown

5.4.1 'specimens' table description

The 'specimens' table captures information relating to Specimens (= Soil Samples). These may be temporary field specimens that have been discarded or ones that have been archived. It is analogous to O&M SF_Specimen.

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
id	integer	True	The database id of the Specimen
current_location	varchar(255)	False	The current storage facility and/or shelf and/or container of the Specimen
sampling_method_id	integer	False	A foreign key to the term in the vocabulary_terms table that identifies the sampling method. In O&M this is an SF_Process, i.e. the sampling procedure consists of a series of steps (to describe for instance combining samples from 10 sites along a transect), not just a single term method. This may be required in future developments.

sampling_time	timestamp	False	The date/time that the specimen was obtained in the field.
specimen_type_id	integer	False	A foreign key to the term in the vocabulary_terms table that identifies the specimen type.
specimen_size	numeric(18)	False	The size of the actual specimen.
specimen_size_uom_id	integer	False	A link to the term in the units_of_measures table that identifies the Unit of Measure of the specimen size.
specimen_identifier	text	False	The persistent identifier (URI) of the Specimen
associated_feature_id	integer	False	A foreign key to the associated features_of_interests table (i.e. the Soil Feature)
related_sampling_feature_id	integer	True	A foreign key to the spatial_sampling_features table to the feature that sampled the Specimen.
specimen_label	varchar(255)	False	The name or label of the Specimen.
specimen_description	text	False	A text description or comment associated with the Specimen.
context_id	integer	True	A foreign key to the context in which the specimen was taken. Usually a job or set of tasks undertaken by a single observer/machine as part of a broader project. Links Job, Project, Client information
specimen_form_id	integer	True	A foreign key to the vocabulary_terms table specifying the basic form of the specimen. e.g. "polished section", "core", "pulp", "solution" Corresponds to SF_Specimen/specimenType

5.5 SAMPLING FEATURES

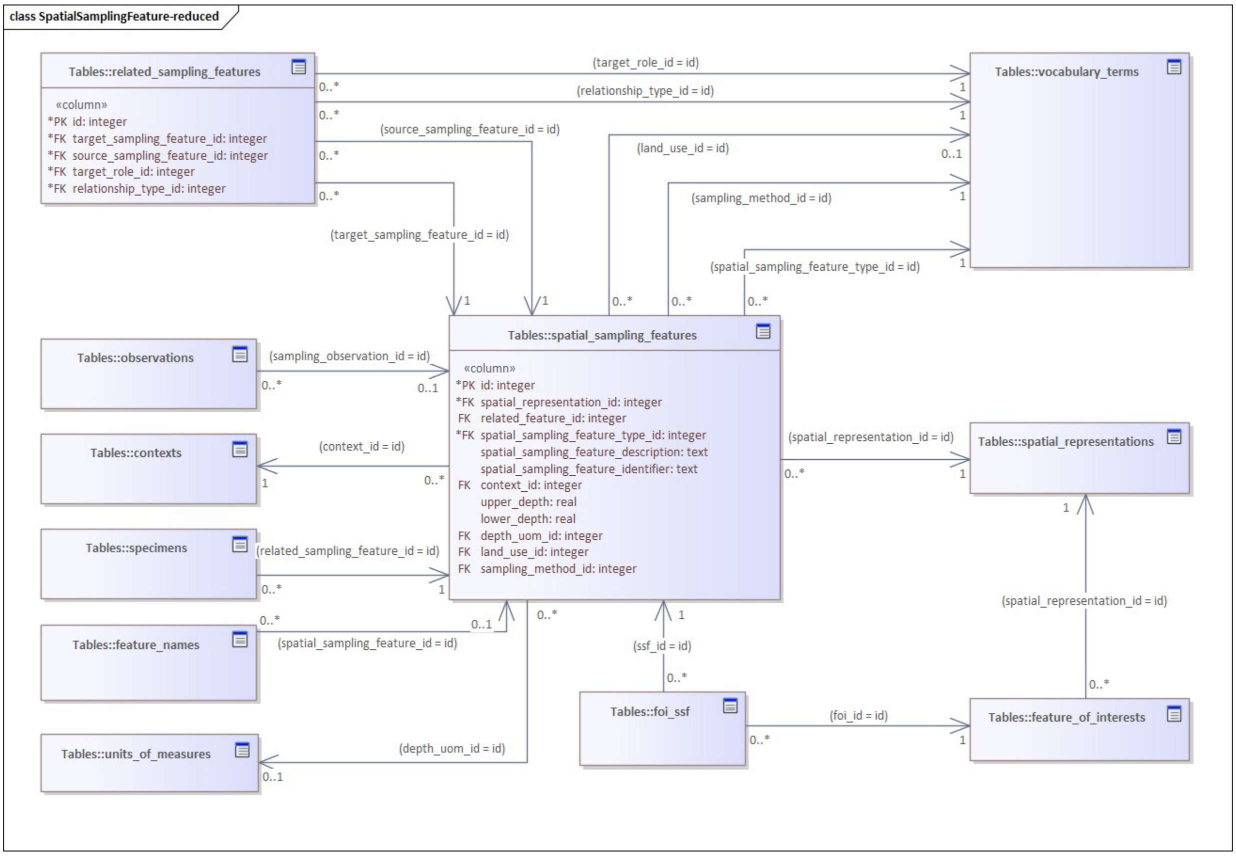

Figure 5.5 'spatial_sampling_features', 'related_sampling_features' and 'spatial_representations' tables.

Columns of related tables and associated relationships are not shown. Together the 'spatial_sampling_features' and 'spatial_representations' tables describe the three-dimensional geometry

5.5.1 'spatial_sampling_features' table description

The 'spatial_sampling_features' and 'spatial_representations' tables store properties analogous to those of the O&M SF_SpatialSamplingFeature. A spatial sampling feature is used when property observations are made on a geospatial feature. This sampling feature may be in one, two or three spatial dimensions. Properties observed on sampling features may be time-dependent, but the temporal axis does not generally contribute to the classification of sampling feature classes. Sampling feature identity is usually less time-dependent than is the property value.

The SF_SpatialSamplingFeature is a type of SF_SamplingFeature. A SF_SamplingFeature is intended to sample some feature of interest in an application domain. They are artefacts of an observational strategy, and have no significant function outside of their role in the observation process. The physical characteristics of the features themselves are of little interest, except perhaps to the manager of a sampling campaign.

SpatialSamplingFeatures may be 'Plot', 'Site', 'Station', 'Paddock', etc. These types are managed through the vocabulary_terms table. A 'station' is essentially an identifiable locality where a sensor system or procedure may be deployed and an observation made. In the context of the observation model, it connotes the 'world in the vicinity of the station', so the observed properties relate to the physical medium at the station, and not to any physical artefact such as a mooring, buoy, benchmark, monument, well, etc. A transient sampling feature, such as a 'ships-track' or 'flight-line', may be identified and described, but is unlikely to be revisited exactly.

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
id	integer	True	The database id
spatial_representation_id	integer	True	A link to the SpatialRepresentation of the SpatialSamplingFeature
related_feature_id	integer	True	The real world feature (e.g.for a SoilFeature the Soil Body, Soil Horizon, Soil Layer, Soil Profile) that the SpatialSamplingFeature samples. (Deprecated 19/10/2022 and replaced with foi_ssf table to allow multiple cardinalities)
spatial_sampling_feature_type_id	integer	True	Identifies the type of SpatialSamplingfeature (e.g. Pit, Plot, Site, Paddock, Borehole, etc). This is a link to a term from a vocabulary in the VocabularyTerm table.
spatial_sampling_feature_description	text	False	A text description or comment related to the SpatialSamplingFeature
context_id	integer	False	A foreign key to the context in which the specimen was taken. Usually a job or set of tasks undertaken by a single observer/machine as part of a broader project. Links Job, Project, Client information
spatial_sampling_feature_identifier	text	False	The persistent identifier (URI) of the SpatialSamplingFeature
upper_depth	real	False	The distance from the surface to the top of the sampling feature
lower_depth	real	False	The distance from the surface to the bottom of the sampling feature

depth_uom_id	integer	False	Identifies the units_of_measures unit that contains the Unit of Measure for the depth values.
land_use_id	integer	False	Identifies the vocabulary term that corresponds to the land use of the spatial sampling feature.
sampling_method_id	integer	False	A link to the term in the VocabularyTerm table that identifies the sampling method. This should be a SF_Process, i.e. the sampling procedure consists of a series of steps (to describe for instance combining samples from 10 sites along a transect), not just a single term method. This may be required in future developments.

5.5.2 **'spatial_representations' table description**

Spatial representation (geometry) of the feature of interest. The depth range and associated unit of measure for true 3D information is stored with the associated spatial_sampling_feature due to database geometry handling considerations.

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
id	integer	True	The database id
geometry	geometry	False	FK to the Spatial geometry of the feature
longitude	real	False	
latitude	real	False	
easting	real	False	
northing	real	False	
mapzone	integer	False	
elevation	real	False	
datum	varchar(50)	False	
spatial_id	varchar(50)	False	A non database, non-unique historical id for the GIS geometry.

5.5.3 **'related_sampling_features' table description**

This table identifies the relationships between the Spatial Sampling Features and the roles each feature plays in the relationship.

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
Id	integer	True	The database id of the relationship
target_sampling_feature_id	integer	True	The target SpatialSamplingFeature
source_sampling_feature_id	integer	True	The source SpatialSamplingFeature
target_role_id	integer	True	A link to a term in the vocabulary_terms table that identifies the role the target SpatialSamplingFeature plays in the relationship.
relationship_type_id	integer	True	A link to a term in the vocabulary_terms table that identifies the type of relationship between the SpatialSamplingFeatures.

5.5.4 **'foi_ssf' table description:**

This table identifies the relationships between the Spatial Sampling Features and the Features of Interest.

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
id	integer	True	The database id of the foi_ssf
foi_id	integer	True	Link to the Feature Of Interests table.
ssf_id	integer	True	Link to the Spatial Sampling Features Table.

5.6 FEATURES OF INTEREST

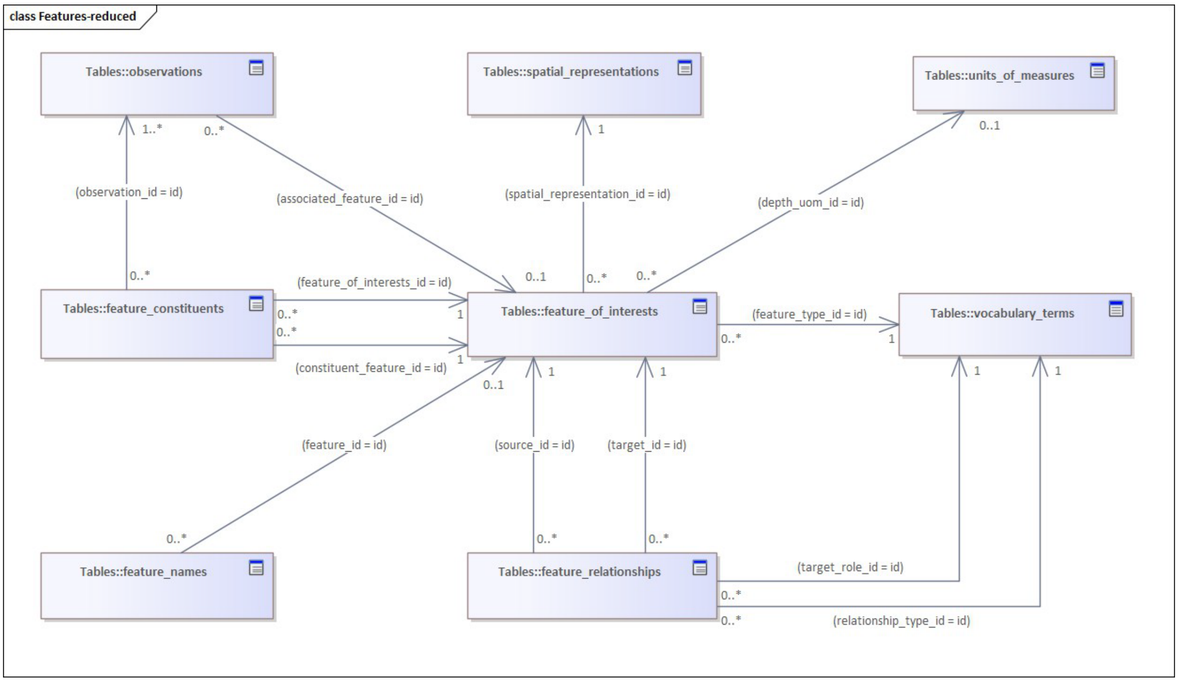

Figure 5.6 'features_of_interests', 'feature_relationships', and 'feature_constituents' tables. Columns of related tables and associated relationships are not shown.

5.6.1 'feature_of_interests' table description

A generic class to handle the real world features being described by the observations. SoilFeature:featureTypes are 'SoilBody', 'SoilHorizon', 'SoilLayer', and 'SoilProfile'. GroundwaterFeature:FeatureTypes are 'Aquifer', 'Borehole', and 'FluidBody.'

The featureTypes are managed through the vocabulary_terms table with values populated from a feature type catalogue.

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
id	integer	True	The database id of the Feature of interest
feature_identifier	text	False	Unique persistent identifier (URI) of the feature
feature_type_id	integer	True	Whether a "SoilBody", "SoilHorizon", "SoilProfile", "SoilLayer", "Aquifer", "EarthMaterial", "SoilConstituent", etc. Values from the vocabulary_terms table.

spatial_representation_id	integer	False	The link to the spatial representation, including depth range, of the Feature.
feature_description	text	False	A free text description of the feature.
upper_depth	float	False	Closest distance from the Earth's surface to the feature.
lower_depth	float	False	Furthest distance from the Earth's surface to the feature.
depth_uom_id	integer	False	Identifies the VocabularyTerm that contains the Unit of Measure for the depth values.

5.6.2 **'feature_relationships' table description**

This table identifies the relationships between the Features of Interests and the roles each feature plays in the relationship. Captures relationships between features such as Soil Horizon to Soil Horizon, Soil Horizon to Soil Body, Soil Horizon to Soil Profile and Soil Layer to Soil Body relationships.

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
id	integer	True	The database id of the relationship
target_role_id	integer	True	Identifies the relationship role the target plays from VocabularyTerm table.
relationship_type_id	integer	True	Identifies the relationship type in VocabularyTerm table
source_id	integer	True	Identifies the source soil feature in SoilFeature table
target_id	integer	True	Identifies the target soil feature in SoilFeature table

5.6.3 **'feature_constituents' table description**

This table identifies the constituent parts of a feature and the proportion that each part comprises.

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
Id	integer	True	The database id of the constituent

constituent_feature_id	integer	True	Foreign Key to the constituent feature of interest. Is part of a composite primary key with the feature_of_interest_id - allows the PK index to group record for the same feature together.
feature_of_interests_id	integer	True	Foreign Key to the undivided feature of interest. Is part of a composite primary key with the constituent_interest_id - allows the PK index to group record for the same feature together.
observation_id	integer	True	The observation that provides the proportion ('asserted values', 'measured ranges', 'measured values')

5.7 PART 2: CONTEXTS (CONTEXTS, PROJECTS, ORGANISATIONS, COLLECTIONS)

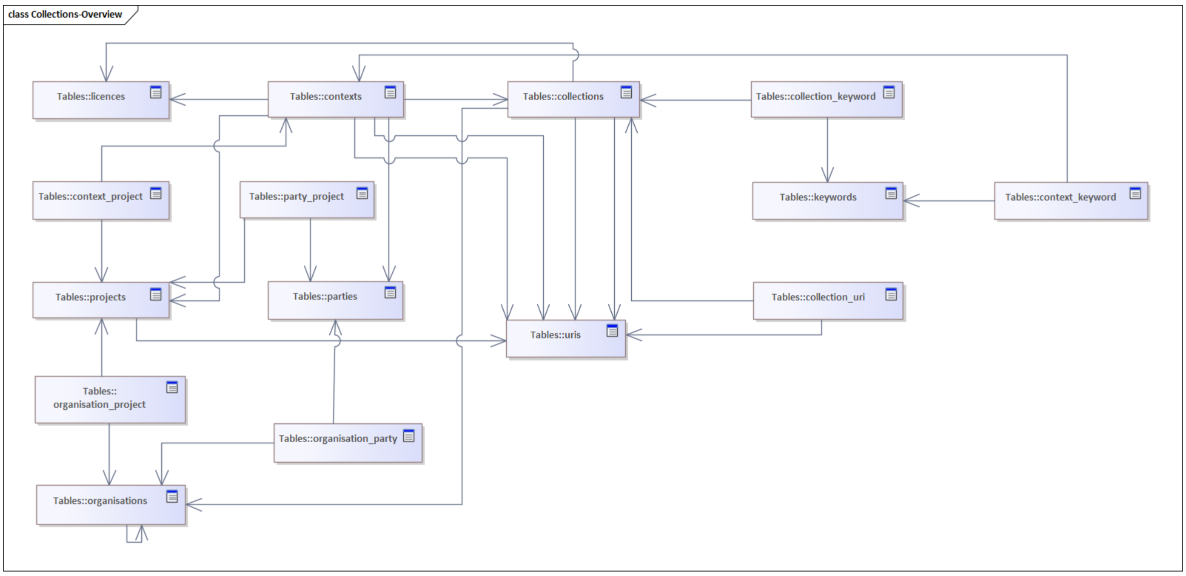

Figure 5.7 Database tables that relate to data contexts

The ‘Contexts’ component stores information about the context of the observational data, its ‘metadata’. This includes job information, project names, organisations and their roles, and users (parties) and their roles, along with licensing information, associated keywords or URI links to further information. The term ‘parties’ has been used to define people who have a specific relationship with the dataset. A separate ‘users’ table defines members that have access to a VAS login and is a system table not defined here.

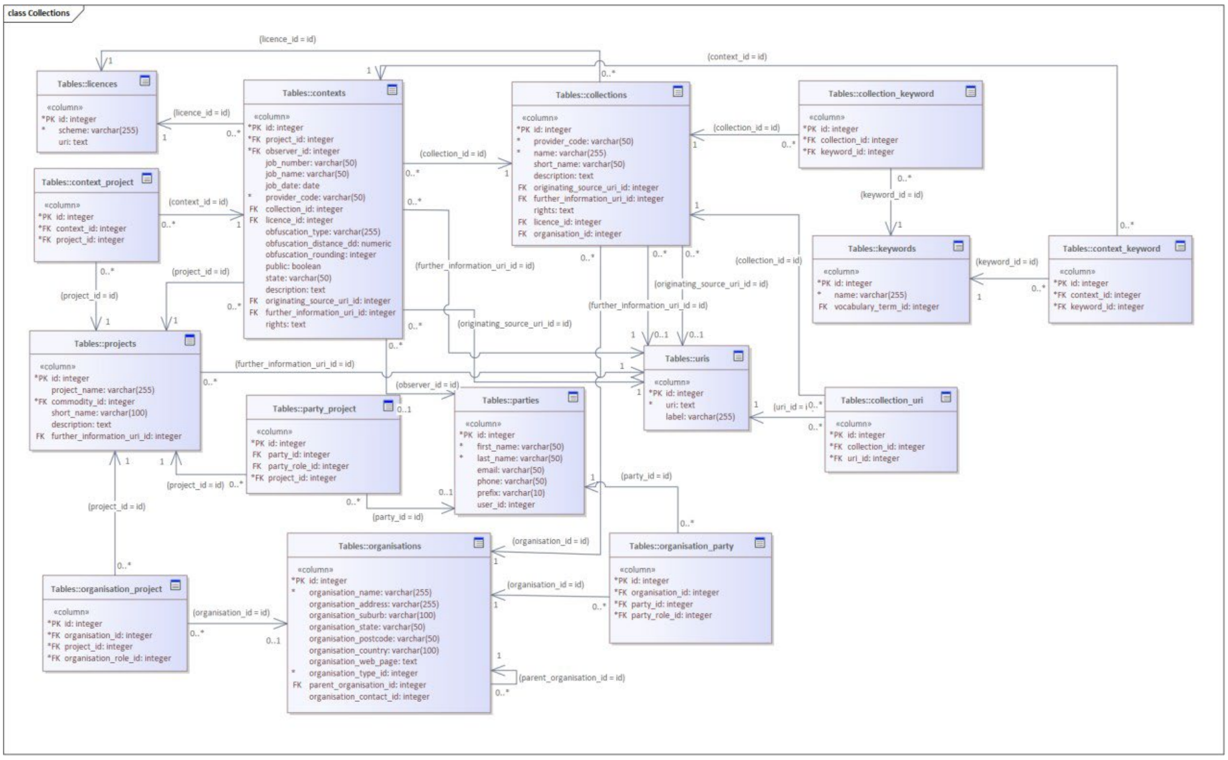

Figure 5.8 Database tables that relate to data contexts: ‘contexts’, ‘projects’, ‘collections’, ‘organisations’, ‘parties’, ‘licenses’, ‘keywords’, and ‘uris’ tables along with linking tables. Columns of further related tables and associated relationships are not shown.

5.7.1 ‘contexts’ table description

This table includes the context in which the observation, specimen, sampling feature was made. Usually a job or set of tasks undertaken by a single observer/machine as part of a broader project (i.e., Links Job, Project, Client information)

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
id	integer	True	The database id
project_id	integer	True	id of the Project associated with the sampling/observing activity
observer_id	integer	False	id of the user who made the observation or took the sample
job_number	varchar(50)	False	The job number
job_name	varchar(50)	False	The job name
job_date	date	False	Date the job was initiated

provider	varchar(50)	True	Provider code used to restrict access to datasets
collection_id	integer	False	A link to the collection
licence_id	integer	False	A link to the license if relevant
obfuscation_type	varchar(255)	False	If not False, specify the type of obfuscation to be applied, e.g. “5km_bbox”
obfuscation_distance_dd	float	False	The distance of obfuscation in decimal degrees
obfuscation_rounding	integer	False	The amount of decimal rounding for location randomisation
public	boolean	False	Is this context intended to be fully publicly available?
state	varchar(50)	False	The Australian state abbreviation most appropriate this dataset
description	text	False	A description of the dataset
originating_source_uri_id	integer	False	Link to the URI containing the original source if relevant
further_information_uri_id	integer	False	Link to the URI containing further information about dataset if relevant
rights	text	False	Any specific rights applicable to the dataset

5.7.2 ‘projects’ table description

This table includes the Project that the observations and/or sampling were part of. Projects are a collection of activities, jobs or tasks that together have a common goal. Although multiple organisations may be associated with a Project, usually only a single organisation has the role ‘client’.

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
id	integer	True	The database id
project_name	varchar(255)	False	The name of the Project
commodity_id	integer	True	The FK to the id in the vocabulary_terms table that corresponds to the commodity that the project is primarily interested in, such as “soil”, “groundwater”.
short_name	varchar(100)	False	An abbreviated name for the project

description	text	False	A description of the project
further_information_uri_id	integer	False	Link to the URI containing further information about project if relevant.

5.7.3 **'parties' table description**

This table includes an individual user's contact details. Multiple users may have multiple roles in a Project.

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
id	integer	True	The database id of the party
first_name	varchar(50)	True	First name of the party
last_name	varchar(50)	True	The parties last name
email	varchar(255)	False	Email address of the party
phone	varchar(20)	False	Phone number of the party
prefix	varchar(10)	False	Prefix or title for the party
user_id	integer	False	Link to user record if relevant (user also has a login to VAS)

5.7.4 **'organisations' table description**

This table includes individual organisation details.

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
id	integer	True	Database id of the organisation
organisation_name	varchar(255)	True	Name of the organisation
organisation_address	varchar(255)	False	Address of the Organisation
organisation_contact_id	integer	False	Link to the individual contact for the organisation's information
organisation_web_page	text	False	The Web home page of the organisation
organisation_type_id	integer	False	A FK to the vocabulary_terms table that corresponds to the term that describes the type of organisation, such as "government", "research", "grower group".
organisation_suburb	varchar(100)	False	Town or suburb of the address of the organisation

organisation_state	varchar(50)	False	State or Territory of the address of the organisation
organisation_postcode	varchar(50)	False	Postcode of the address of the organisation. Needs to be able to handle international postcodes
organisation_country	varchar(100)	False	Country of the address of the organisation
parent_organisation_id	integer	False	The id of the parent organisation, if present.

5.7.5 **'organisation_project' table description**

This table includes multiple organisations that may be associated with a project and multiple projects per organisation.

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
id	integer	True	Database id
organisation_id	integer	True	FK to the Organisation Table
project_id	integer	True	FK to the Project table
organisation_role_id	integer	True	FK to the vocabulary_terms table for the term that corresponds to the role the organisation plays in the Project. A project, such as 'Precision Agriculture soil tests' may have multiple organisations such as 'Farm A', 'Farm B' playing the role 'contributor' with the organisation 'Precision Agriculture' playing the role 'client'. In different projects the same organisations may play different roles.

5.7.6 **'organisation_party' table description**

This table includes the multiple organisations that may be associated with a party and multiple parties per organisation.

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
id	integer	True	Database id
organisation_id	integer	True	FK to the Organisation Table.
party_id	integer	True	FK to the Party Table.

party_role_id	integer	True	FK to the vocabulary_terms table for the term that corresponds to the role the party plays in the organisation. An organisation, such as 'Precision Agriculture' may have multiple parties such as 'John Smith' playing the role 'contributor' with the party 'Jane Smith' playing the role 'point of contact'. In different organisations the same party may play different roles.

5.7.7 **'context_project' table description**

This table includes multiple projects that may be associated with multiple contexts per project.

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
id	integer	True	Database id
context_id	integer	True	FK to the Contexts Table.
project_id	integer	True	FK to the Project table.

5.7.8 **'party_project' table description**

This table includes multiple parties that may be associated with a project and multiple projects per party.

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
id	integer	True	Database id
party_id	integer	True	FK to the Party Table
project_id	integer	True	FK to the Project table
party_role_id	integer	True	FK to the vocabulary_terms table for the term that corresponds to the role the party plays in the Project. A project, such as 'Precision Agriculture soil tests' may have multiple parties such as 'John Smith', 'Jane Smith' playing the role 'contributor'. In different projects the same party may play different roles.

5.7.9 **'collections' table description**

A collection can be used to group a number of contexts together into a related set of data.

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
id	integer	True	The database id of the collection
provider_code	varchar	True	Provider code used to restrict access to datasets
name	varchar	True	The name of the collection
short_name	varchar	False	A short version of the collection name if relevant
description	text	False	A description of the collection
originating_source_uri_id	integer	False	Link to the URI containing original source information if relevant
further_information_source_id	integer	False	Link to the URI containing further information about dataset if relevant
rights	text	False	Any specific rights applicable to the collection
licence_id	integer	False	FK to licence if relevant.
organisation_id	integer	True	FK to the organisation that the collection belongs to

5.7.10 **'keywords' table description:**

Keywords can be used to store terms that are relevant to the dataset.

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
id	integer	True	Database id
name	varchar(255)	True	The name of the keyword
vocabulary_term_id	integer	False	FK to the vocabulary term if relevant.

5.7.11 **'licences' table description:**

Details on the licence applicable to the context.

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
id	integer	True	Database id
scheme	varchar(255)	True	The licence scheme
uri	text	True	The URI for the licence scheme (e.g. 'GPL (General Public License)')

5.7.12 **'uris' table description:**

This table is used to store any collection or dataset related URIs. This could be an external link to a report or to an organisation's website.

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
id	integer	True	Database id
uri	text	True	The URI link
label	varchar(255)	False	A label for the URI

5.7.13 **'collection_keyword' table description:**

Multiple keywords may be associated with a collection and multiple collections per keyword.

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
id	integer	True	Database id
collection_id	integer	True	FK to the Collection Table.
keyword_id	integer	True	FK to the Keyword Table.

5.7.14 **'collection_uri' table description:**

Multiple URIs may be associated with a collection and multiple collections per URI.

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
id	integer	True	Database id
collection_id	integer	True	FK to the Collection Table.
uri_id	integer	True	FK to the URI Table.

5.7.15 **'context_keyword' table description:**

Multiple keywords may be associated with multiple contexts per keyword.

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
id	integer	True	Database id
context_id	integer	True	FK to the Contexts Table.
keyword_id	integer	True	FK to the Keyword Table.

5.8 PART 3: AUXILIARY TABLES (VOCABULARY TERMS, UNITS OF MEASURE, FEATURE NAMES, REPORTS)

The 'vocabulary_terms' and 'units_of_measures' tables are central to providing controlled terms to the concepts stored in the database. These two tables play the role of 'look-up' tables, with additional associated information and links to external references and ontologies for the concepts. The intention is that external parties manage these terms using external applications that then make the vocabularies available via web services. The CeRDI Observations Database harvests these concepts and stores them as a local cache.

5.9 VOCABULARY TERMS

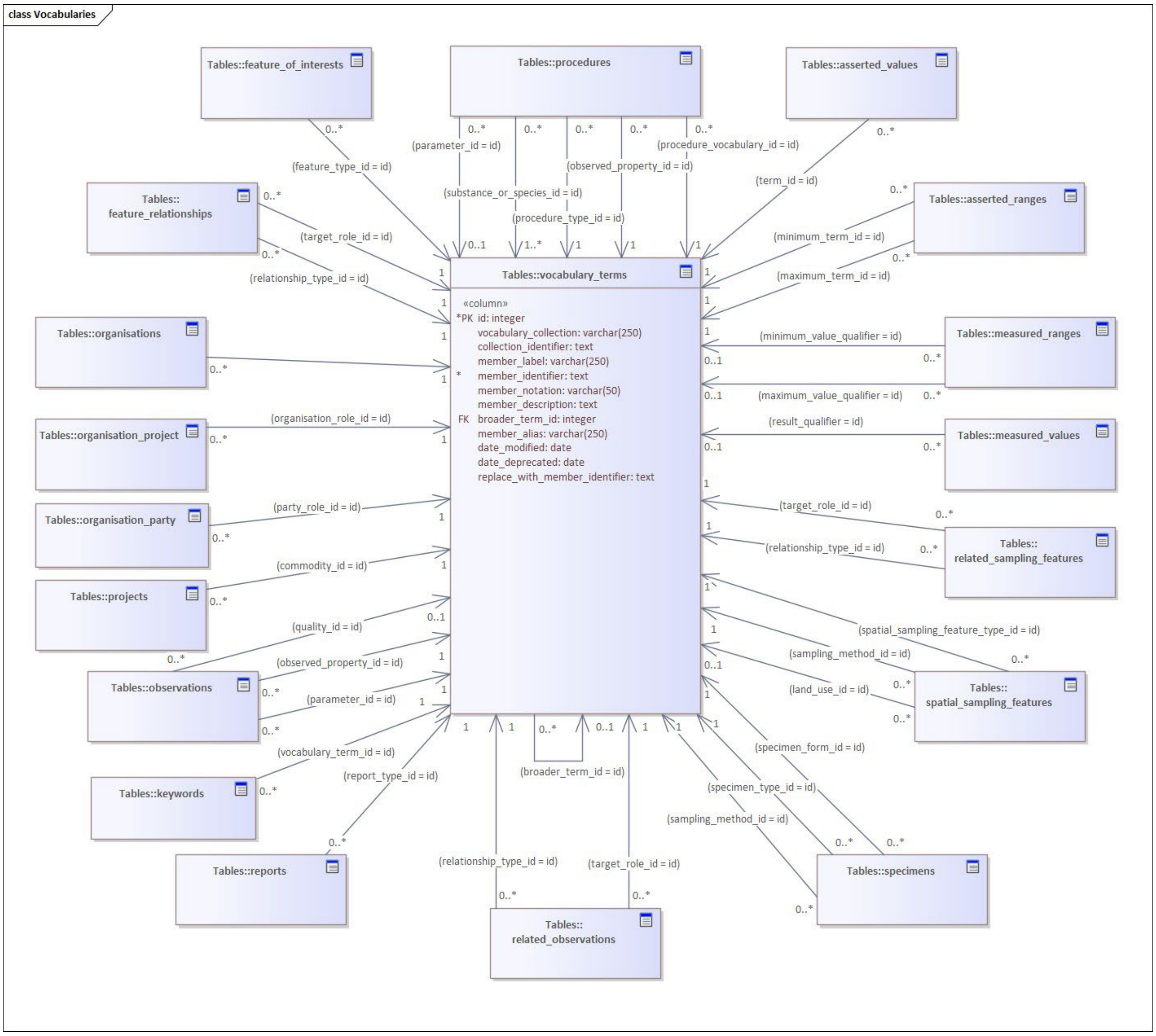

Figure 5.9 The 'vocabulary_terms' table and the associated tables that use it. Columns of related tables and associated relationships are not shown.

5.9.1 **'vocabulary_terms' table description**

This table is used to store required terms. These may be terms from traditional look-up tables such as terms associated with 'laboratory methods' (e.g., 'pH using 1.5 HCl'), or with 'drainage' (e.g., 'well-drained', 'poorly drained'). They may also be the terms that are traditionally the look-up table names and appear as database columns (e.g., 'laboratory method', 'pH', 'P concentration').

Importantly, in addition to a set of terms the table indicates what schema the term comes from (i.e., what the look-up table name would be that the term came from if from a look-up table), a description, and a persistent identifier to an external resource that contains more information (e.g., images, other relationships, conversion values, alternative labels or languages).

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
id	integer	True	The database column id
vocabulary_collection	varchar(255)	False	The label of the collection that the vocabulary member belongs to (e.g. "drainage", "laboratory method", "observed property")
collection_identifier	text	False	Unique persistent id (URI) of the collection the vocabulary term belongs to
member_label	varchar(255)	False	Preferred label of the vocabulary member (e.g. "poorly drained"). This is what is normally stored in the look-up table.
member_identifier	text	True	Unique external persistent identifier (URI) of the vocabulary term
member_notation	varchar(50)	False	Code/abbreviation for the vocabulary member
member_description	text	False	Description of the vocabulary term
broader_term_id	integer	False	The id of the broader term in a vocabulary hierarchy, if present
member_alias	varchar(255)	False	An alternative member label
date_modified	date	False	Recorded database date of modification of vocabulary term
date_deprecated	date	False	Date this record was deprecated (if applicable)
replace_with_member_identifier	text	False	Member identifier this term has been replaced by (if applicable)

5.10 UNITS OF MEASURE

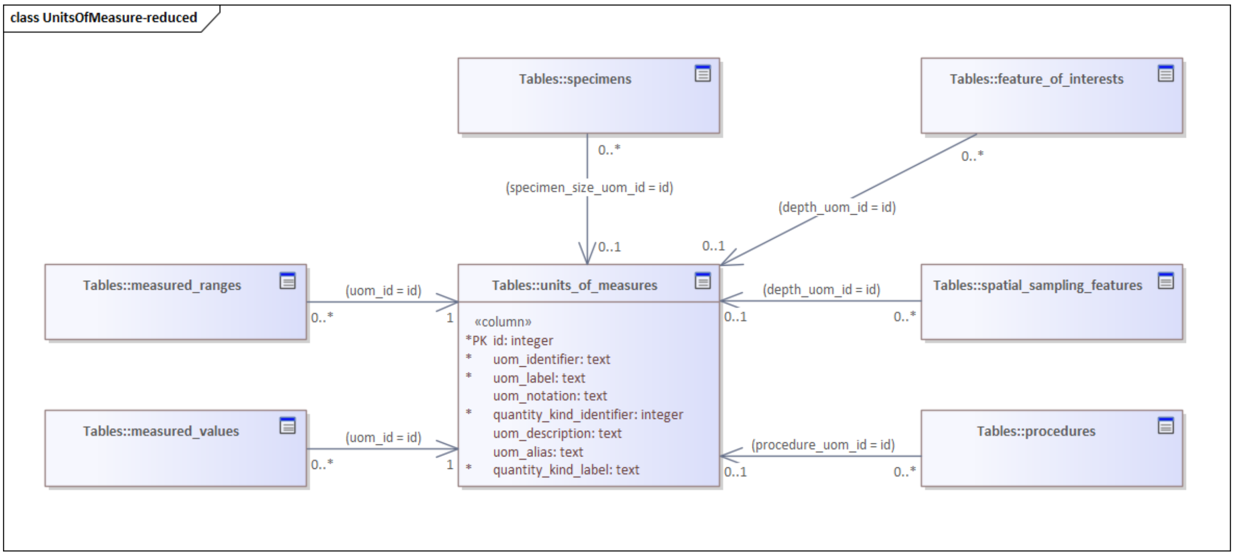

Figure 5.10 The 'units_of_measures' table and the associated tables that use it. Columns of related tables and associated relationships are not shown.

5.10.1 'units_of_measures' table description

The units of measure is based on the Quantities, Units, Dimensions and Types Ontology (QUDT). It allows specifying the label (e.g., 'metre'), its abbreviation (e.g., 'm'), alternative labels (e.g. 'meter'), and the kind of measure it relates to (e.g., 'Length'). The table also caters for the identifiers (URIs) for the unit of measure and its quantity kind.

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
id	integer	True	Database column id
uom_identifier	text	True	URI of the unit of measure term. From Linked Data Registry = rdf:about
uom_label	text	True	The text label of the UoM term. From Linked Data Registry = rdfs:label
uom_notation	text	False	The abbreviation for the UoM term. From Linked Data Registry = qudt:abbreviation or qudt:symbol or skos:notation

quantity_kind_identifier	integer	True	The URI for the type of quantity that the UoM is (e.g. 'metre' is a 'Length' 'QuantityKind') From Linked Data Registry = https://qudt.org/schema/qudt/QuantityKind Because QuantityKinds are hierarchical a Unit of Measure may have multiple QuantityKinds: e.g. '**Atoms per Litre'** is a unit of measure for QuantityKinds of '**Concentration**' as well as its specialization '**Amount of Substance Per Unit Volume**'. The database design only allows for a Unit of Measure to be assigned to a single QuantityKind, preferably the most specialized.
uom_description	text	False	Text to describe the UoM. From Linked Data Registry = qudt:description or dct:description or skos:definition
uom_alias	text	False	An alternative label or spelling of the UoM term (e.g. 'metre' rather than 'meter') From Linked Data Registry = skos:altLabel
quantity_kind_label	text	True	The label of the QuantityKind (e.g. “Length”). From Linked Data Registry = https://www.w3.org/2000/01/rdf-schema#label

5.11 FEATURE NAMES

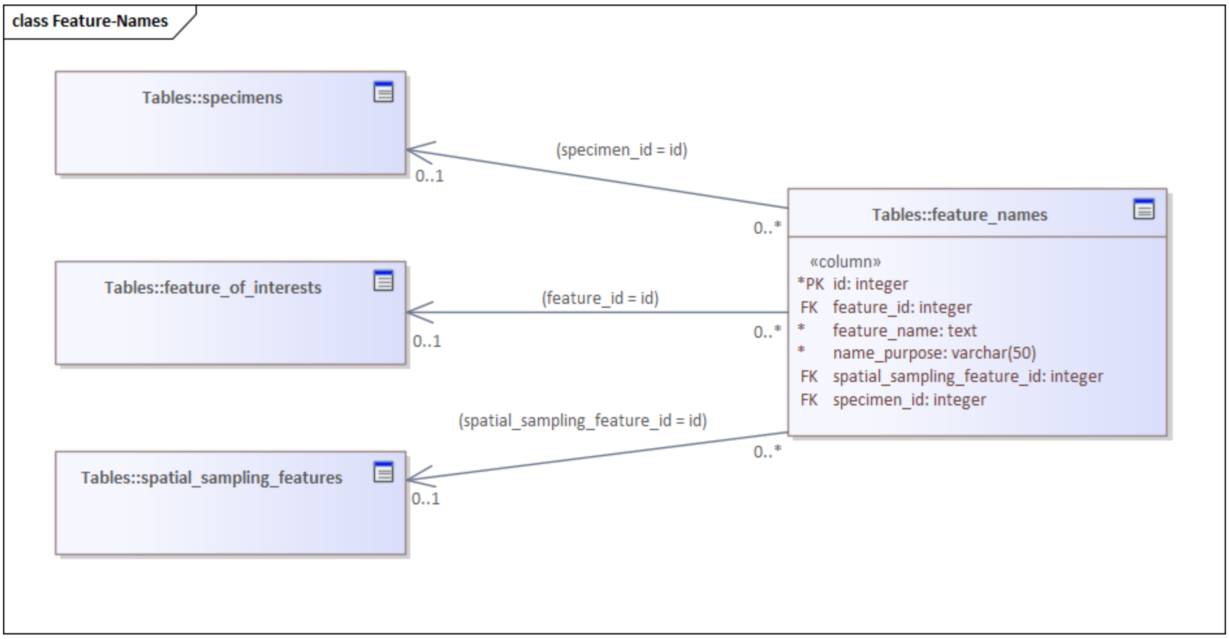

Figure 5.11 The 'feature_names' table allows specifying multiple names (identity) to individual features, whether they are features of interest, specimens or spatial sampling features. Columns of related tables and associated relationships are not shown.

5.11.1 'feature_names' table description

This table allows for distinguishing multiple names of any single feature.

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
id	integer	True	The database column id
feature_id	integer	False	The identity of the feature of interest that the name applies to
feature_name	text	True	The name of the feature (e.g. "Lismore Grey Clays" and/or "33", or "Burt's Paddock", or "Specimen Barcode 222222", "Borehole M124"etc.)
name_purpose	varchar(50)	True	Allows distinguishing the reason for multiple names
spatial_sampling_feature_id	integer	False	The identity of the spatial sampling feature that the name applies to
specimen_id	integer	False	The identity of the specimen that the name applies to

6. VAS SYSTEM APPLICATION PROGRAMMING INTERFACE (API)

As shown in Figure 4.1 of this report, soil data stored in the VAS aggregator and other data consumed from external sources is published from the VAS system via an application programming interface (API). The VAS API strives to implement the best practice set out by the World Wide Web Consortium (W3C) for publishing data on the web.

- Data on the Web Best Practices https://www.w3.org/TR/dwbp/
- Best Practices for Publishing Linked Data https://www.w3.org/TR/ld-bp/
- Spatial Data on the Web Best Practices https://www.w3.org/TR/sdw-bp/

To achieve these goals, the project chose to implement a REST-full API, using persistent identifiers and data primarily delivered in JSON-LD format. These payloads were, in turn, based on a set of best practices for Building JSON-LD APIs https://json-ld.org/spec/latest/json-ld-api-best-practices/

JSON-LD is an ideal data format for programming environments familiar with the pervasive JSON format, being fully backwards-compatible with JSON libraries and parsers. It adds interoperability by providing the ability to bind JSON elements, attributes, and values to formal data structures, standards, and vocabularies. To this end, the VAS API has attempted to re-use existing international standards, vocabularies, and common patterns when defining its data structures.

In particular, the Endpoints described here have been based on the Sensor, Observation, Sample, and Actuator (SOSA) ontology https://www.w3.org/TR/vocab-ssn/ but should not be considered fully compliant with that specification. Other standards and vocabularies utilised include:

- Schema.org [https://schema.org/]
- Quantities, Units, Dimensions, and Types (QUDT) [https://www.qudt.org/]
- Provenance ontology (PROV) [https://www.w3.org/TR/prov-dm/]
- Simple Knowledge Organization System (SKOS) [https://www.w3.org/TR/skos-reference/]
- GeoSPARQL [https://www.ogc.org/standard/geosparql/]
- ANZSoilML & ANSIS controlled vocabularies for soil data

6.1 API SPECIFICATION

An OpenAPI specification document describing the VAS API is available at https://app.swaggerhub.com/apis-docs/FedUniCeRDI/vas-soils-api/1.0# but an abbreviated version is presented here as an introduction.

Base URI:

https://id.cerdi.edu.au Main

Endpoints:

Name	URI
Feature	/{provider}/data/sosa/feature /{provider}/data/sosa/feature/{id} Examples: /ccma/data/sosa/feature /ccma/data/sosa/feature/ccma.soil.feature.7174
Observation	/{provider}/data/sosa/observation /{provider}/data/sosa/observation/{id} Examples: /ccma/data/sosa/observation /ccma/data/sosa/observation/ccma.soil.observation.10840
Dataset	/{provider}/data/schema/dataset /{provider}/data/schema/dataset/{contextJobNumber} Examples: /ccma/data/schema/dataset /ccma/data/schema/dataset/000001
Observed Properties	/query/observedproperties
Procedures	/query/procedures
PropertiesProcedures	/query/propertyprocedures /query/propertyprocedures/{provider}

6.1.1 Example response for a single observation


```
{
    "@graph": {
        "@id": "https://id.cerdi.edu.au/ccma/data/sosa/observation/ccma.soil.observation.10840",
        "@type": "Observation",
        "hasFeatureOfInterest": {
            "@id": "https://id.cerdi.edu.au/ccma/data/sosa/sample/ccma.soil.sample.1659.1659"
        },
        "hasResult": {
            "numericValue":
            5.2, "unit": {
                "@id": "https://qudt.org/vocab/unit/PH",
                "abbreviation": "pH",
                "label": "Acidity"
            }
        },
        "hasSimpleResult": 5.2,
        "hasUltimateFeatureOfInterest": {
            "@id": "https://id.cerdi.edu.au/ccma/data/sosa/feature/ccma.soil.feature.7174"
        },
        "inDataset": "https://id.cerdi.edu.au/ccma/data/schema/dataset/ccma.schema.dataset.000001",
        "name": "ccma.soil.observation.10840",
        "observedProperty": {
            "@id": "http://environment.data.gov.au/def/property/PH",
            "@type": "Concept",
            "description": "negative logarithm of hydrogen ion concentration in ph units",
            "label": "pH",
            "name": "pH",
            "notation": null
        },
        "phenomenonTime": {
            "@type": "http://www.w3.org/2001/XMLSchema#dateTimeStamp",
            "@value": "2018-05-15T00:00:00"
        },
        "provider": "ccma",
        "resultTime": {
            "@type": "http://www.w3.org/2001/XMLSchema#dateTimeStamp",
            "@value": "2018-05-15T00:00:00"
        },
        "soDepth": {
            "soMaximumValue": {
                "@value": 20
            },
            "soMinimumValue": {
                "@value": 10
            },
            "unit": {
                "@id":
                "https://qudt.org/vocab/unit/CentiM",
                "abbreviation": "cm"
            }
        },
        "ufoi_id": 7174,
        "usedProcedure": {
            "@id": "http://anzsoil.org/def/au/scm/4B4",
            "@type": "Concept",
            "description": "pH of 1:5 soil/0.01 M calcium chloride extract - following method 4A1
(with stirring during measurement)",
            "label": "pH of 1:5 soil/0.01 M calcium chloride extract - 4B4",
            "name": "pH of 1:5 soil/0.01 M calcium chloride extract - following method 4A1 (with
stirring during measurement)",
            "notation": "4B4"
        }
    }
```

6.1.2 Other example requests

In the case of the Corangamite Catchment Management Authority (CCMA) dataset, a suite of soil chemistry analyses - 'observations' - is available for each of the features of type *SoilLayer*. These observations can be retrieved using the query syntax below (i.e., return all the observations that relate to these two features). In this case, these two features represent 0-10cm (soil layer depth) at a specific location in 2014 and 0-10cm at the same location in 2018.

```
https://id.cerdi.edu.au/ccma/data/sosa/observation/?or=(
 hasUltimateFeatureOfInterest-->"@id".eq.https://id.cerdi…/feature/ccma.soil.feature.6868,
 hasUltimateFeatureOfInterest->>"@id".eq.https://id.cerdi…/feature/ccma.soil.feature.14
)
```

To go the other way (from observation up to feature) is also possible (e.g., return all the observations that have Observed Property = Bulk Density).

```
https://id.cerdi.edu.au/ccma/data/sosa/observation/?
 observedProperty->>"@id"=eq.http://…cerdi.edu.au/property/bulk density soil
```

The response to the request above is a standard SOSA pattern, where the value URI of `hasUltimateFeatureOfInterest` for each observation would need to be resolved to get more information about the feature (including spatial information).

There is a convenience version of the observation end-point which includes the *f_geom* in each observation for use in spatial visualisation by property value.

```
https://id.cerdi.edu.au/ccma/data/sosa/observationsearch/?
 observedProperty->>"@id".=eq.http://placeholder.cerdi.edu.au/property/bulk density soil
```

Observations matching specific observedProperty and usedProcedure

```
https://id.cerdi.edu.au/ccma/data/sosa/observation/?or=(
 observedProperty->>"@id".eq. http://environment.../def/property/phosphorus_concentration,
 usedProcedure->>"@id".eq.http://www.anzsoil.org/def/au/scma/9B)
```

As the API is based on PostgREST middleware, it should support all the features of a typical ODATA [https://www.odata.org/documentation/] API. This includes attribute selection, pagination, ordering etc. Most of the example queries documented at the URL below should apply.

https://postgrest.org/en/stable/references/api/tables_views.html

For example, the observations end point example for bulk density with a limit and attribute selection added.

```
https://id.cerdi.edu.au/ccma/data/sosa/observationsearch/?observedProperty
 -->”@id”.=eq.http://placeholder.cerdi.edu.au/property/bulk_density_soil
 &limit=20
 &select=provider,f geom,hasResult
```

6.2 AUTHENTICATION

Since the majority of datasets are non-public at the present time, access is provided to authorised datasets via an API key. **An authentication header** with a valid API Key is required for many of the REST endpoints when requesting non-public data. The *Authorization* field in the HTTP header is used to pass the API key (or token).

7. SOIL PROBE DATA: ARCHITECTURE DEVELOPMENT

VAS Phase Two aspired to include data collected by soil probes available via the VAS portal. Since most of the participating farmer groups in the VAS project have soil moisture and temperature probes installed, methods were explored to include these data in the soil data federation. Probe data are usually telemetered to a digital repository run by a third party and then graphically displayed on the farmer group website, via a member login.

An initial use-case to include Southern Farming Systems (SFS) probe data in the VAS portal was undertaken. The methods built on a previous project undertaken by CeRDI for SFS, which established a portal known as Probetrax[4] in which the public can view the current plant available water (PAW) as a percentage value (%) at each probe site as a coloured dot. In the Probetrax portal, SFS members hosting a probe can log into the portal and view their time-series soil moisture and temperature down profile as graphs, together with the rainfall, if a rain gauge was fitted.

Displaying the SFS soil moisture probes in the VAS portal was achieved (Figure 7.1). Additional methods were implemented include data from Agriculture Victoria's soil moisture probe network in the VAS portal (Figure 7.2).

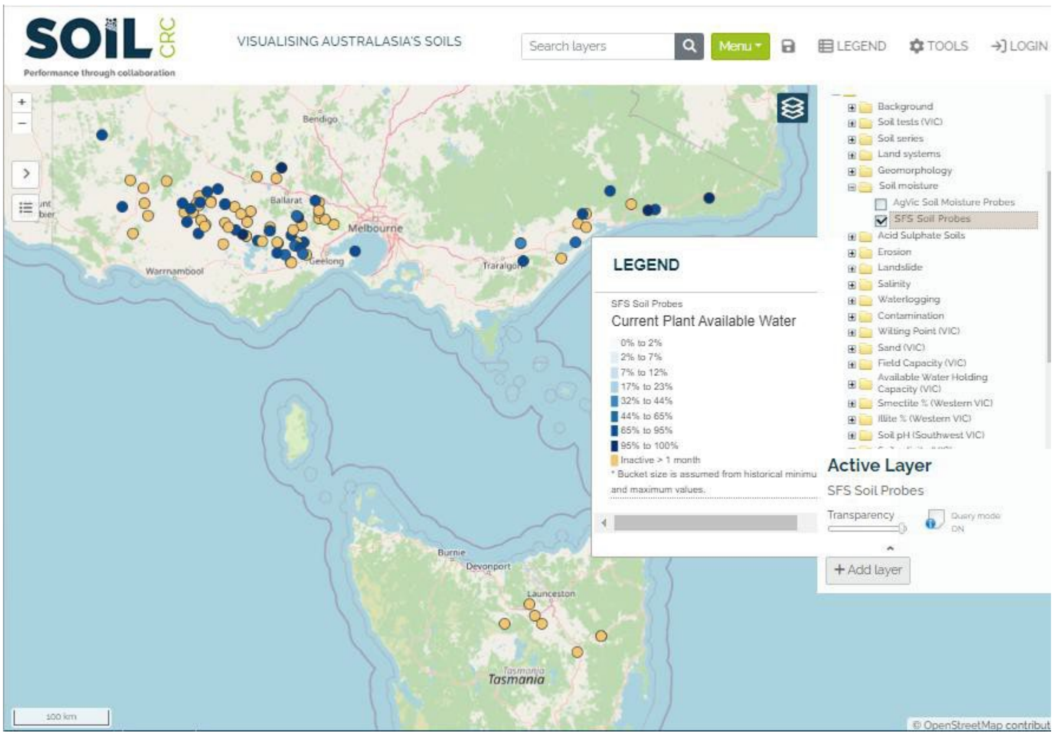

Figure 7.1 SFS soil moisture probes displayed in the VAS portal (8/8/2024).

---

[4] https://probetrax.sfs.org.au/

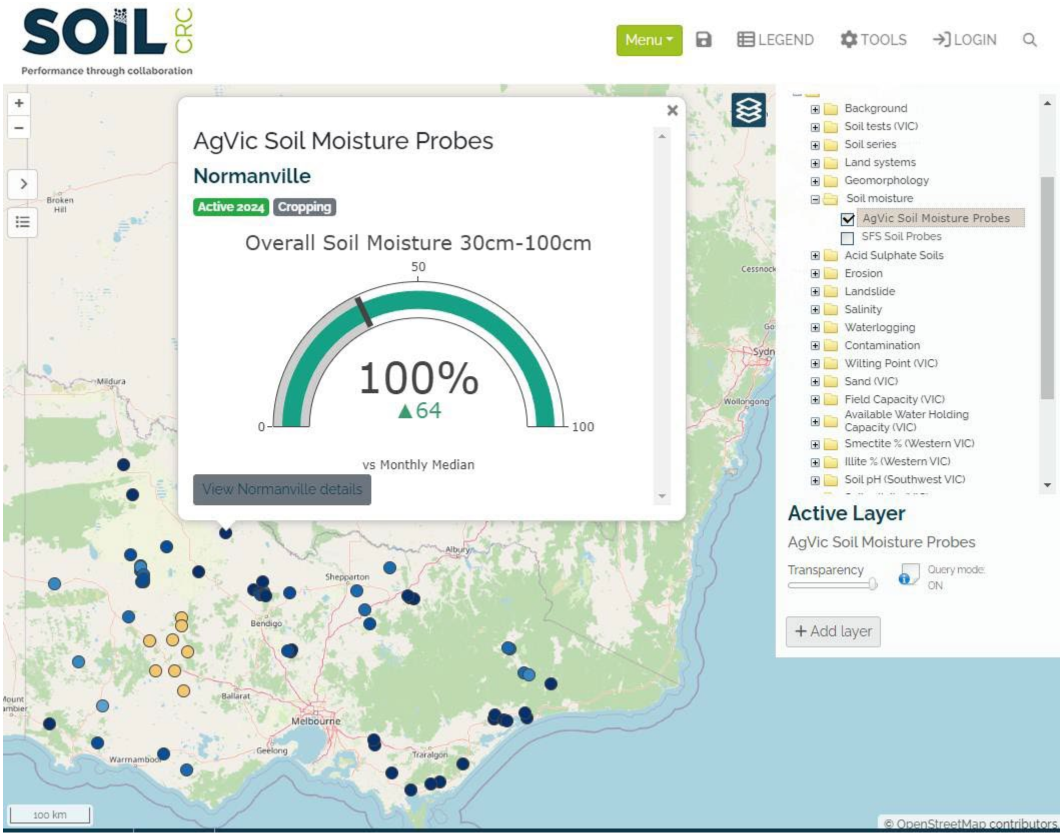

Figure 7.2 Agriculture Victoria's soil moisture probes in the VAS portal (8/8/2024).

Research effort has continued throughout 2024 on integrating soil probes from more providers across Australia within VAS. A key objective is to be able to provide 'on-demand' access for a user to request 'all the data' (i.e., include any sensor value that is reported, not just soil moisture). This request springs from researchers, especially PhD students, who are undertaking research projects that require access to time-series soil moisture, and/or temperature throughout a soil profile across a wide area. Among the many challenges this creates is the fact that these data for one provider (SFS) amounts to well over 600 million observations.

A new VAS soil moisture probe interface was developed in early 2024 to allow access to 'Verified' (Soil Moisture) sensors and 'Unverified' sensors (All other sensors), as well as selecting an appropriate data frequency (Figure 7.3). However this interface upgrade has remained on hold pending other VAS upgrades and a resolution to the probe data issues.

After lengthy development and testing, the technical team made the decision to abandon the existing SensorThings-based Frost service, deployed for the Probetrax architecture, as it has proven to be a major barrier in this objective. While being OGC compliant and achieving some FAIR data objectives, the lack of aggregate data queries in the standard makes the end-use very developer-unfriendly, requiring thousands of API calls to build the dataset.

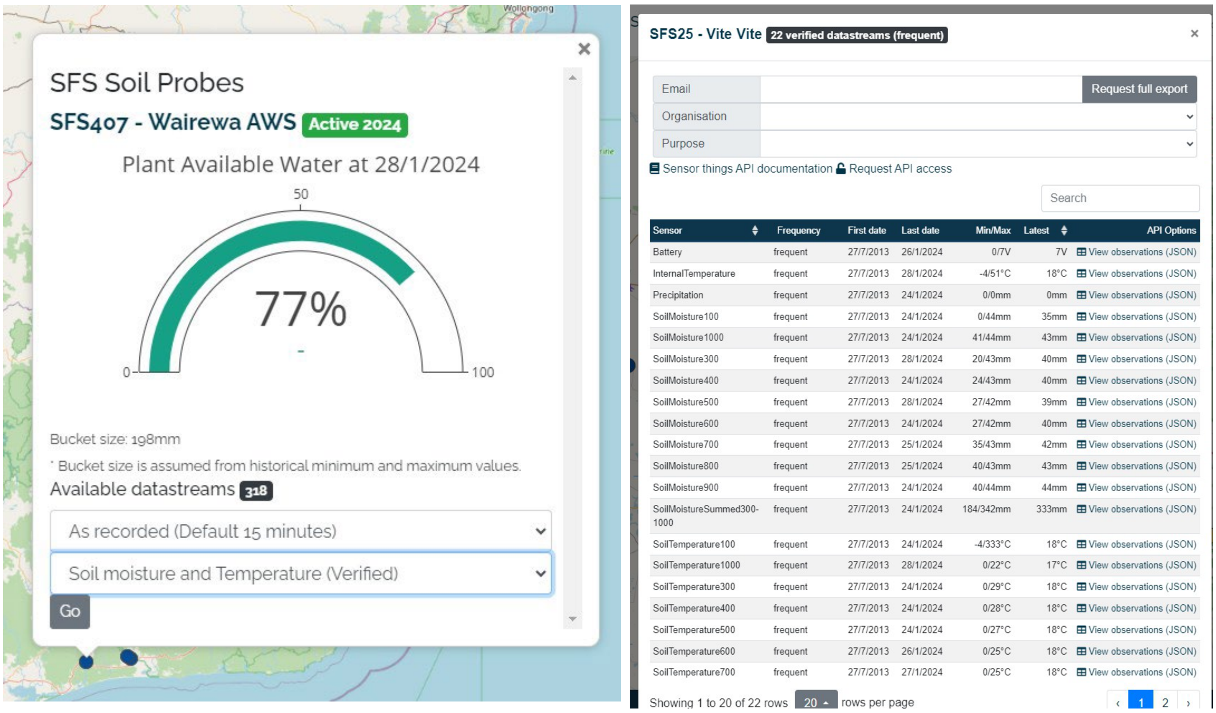

Figure 7.3 An example of the interface using verified data.

The background processing for importing multi-year backlogs of data via SensorThings Frost Server is very lengthy (i.e., weeks/days), and much difficulty was encountered in verifying the imports had completed successfully, although it is acknowledged that some of these difficulties may be particular to CeRDI's combination of infrastructure and SensorThings, and not specific to the Frost Server itself. Nevertheless, the process introduced a significant risk, since if a central Frost SensorThings service was used for all providers, and a data issue was uncovered, it may take months to wipe and re-ingest the data for all providers. On the flip side, a dedicated Frost SensorThings service for each provider brought other issues when it comes to the goal of providing all the data together, as well as for maintainability.

A secondary goal in the VAS soil data federation is to have a central system ready to connect other soil probe data providers across Australasia. This development has been hampered by the fact that no project participants have yet provided API access to their probe data in 2023/2024. At present only API connections from SFS data stream provider, Adcon, and Mackillop Farm Management Group (MFMG) data stream provider, Wildeye, have been tested. Given the number of issues uncovered from these two experiments, it is expected that many more issues will be uncovered as more providers come on board.

As a result of the above research, a decision was made to develop a new technical architecture and system to overcome these challenges.

7.1 CERDI MULTI PROVIDER PROBE COLLECTOR

In 2024 development has started on a new probe collector platform for CeRDI's purposes. This collector will:

- replace existing standalone import scripts for Wildeye and Adcon data stream providers.
- replace custom frameworks and legacy code with a modern Laravel framework.
- utilise TimescaleDB for performance and efficient live data aggregates.
- provide a straightforward token-based API service to request data instead of a SensorThings datastream service.

7.1.1 API Routes

The API routes in development are as follows:

Create Sanctum Token

User Required

`POST /api/login`

NOTE: Username and login must be created via Artisan Tinker until a registration process exists.

Get Device List (All providers)

Token Required

`GET /api/devices/list/devices/list`

Get Sensor List

Token Required

`GET /{provider}/sensors/{device}/list`

Get Observed Properties

Token Required

`GET /observed-properties/list`

Get Units of Measure

Token Required

`GET /unit-of-measure/list`

Get Observations

Token Required

Paginated at 5000 per page

Optional parameters:

start_date YYYY-MM-DD 00:00:00

end_date YYYY-MM-DD 00:00:00

By Sensor

Token Required

```
GET /{provider}/observations/sensor/{sensor}/list
```

By Device

Token Required

```
GET /{provider}/observations/device/{device}/list
```

By Frequency

Token Required

```
/{provider}/observations/frequency/hourly/list  /{provider}/observations/frequency/daily/list
/{provider}/observations/frequency/weekly/list
/{provider}/observations/frequency/monthly/list
```

Export Observations

Generate CSV Request

Token Required

```
/{provider}/observations/export
```

Download generated CSV*

Token Required

```
/{provider}/observations/export/{download_number}
```

*Maximum of 500k observations returned

7.1.2 Progress and research challenges

The development of this API has raised many research challenges with soil probe data not previously considered by the VAS technical team. Aside from VAS, the Multi Provider Probe Collector service is required for other significant CeRDI research projects such as a Future Drought Fund Small Farm Dams Project, and the SFS My Farm Dashboard project. Some of the same issues have been encountered working on probe data for these projects, and the learnings and solutions are transferrable VAS. These issues are discussed below.

7.1.2.1 Inactive probe issues

There are a growing number of probes from some project participants that are no longer operational. The upcoming impact of the Telstra 3G Mobile Service shutdown will also make this issue larger unless replacement probe SIM cards are funded. The number of inactive probes can result in misleading maps. That is, the map may display what appears to be the current state of soil moisture in a region, but inactive probes may be showing the last operational value (i.e., stale data).

7.1.2.2 Ingesting data without sufficient context

The original SensorThings API was mapped carefully to observed properties and manual descriptions of the soil moisture probe sensors so that the data made sense to the end-user. Only sensors with an established definition and vocabulary were imported into the system. However, when ingesting all API data without such a vetting process, the quality of information is mixed and context is lacking. Issues include:

- inconsistent probe naming conventions, both across providers and within the same provider
- misleading sensor configurations, for example:
  - ‘10cm soil moisture’ is actually ‘20cm soil moisture’ at a certain location, as the 10cm probe was moved to a different location, and a 10-90cm probe was installed 20cm deep.
  - An extension called ‘NW 1’ is actually ‘Rainfall since 9am at NW1’, but the name is only in the parent node.
  - Sensors close together that have different soil and crop types (Lucerne and Veldt).
  - A user preference to display one of those probes, and not the other.
- exposing data that should not be exposed, for example:
  - CeRDI’s API access can be more elevated than required. Without careful checking, partners not participating in the VAS project may be exposed.
  - There have been instances where probe data was included via the API, but this data was never intended for the project partner, since the probe provider was merely testing a different brand of sensor in parallel, which was not made available via the partner’s dashboard.
- unannounced changes in naming convention or hierarchy. This can mean probe data ceases to import, as the corresponding identifier is no longer valid. This potentially allows for CeRDI to ingest data it is not permitted to use if the identifier is changed to one thought to be permitted.

- new probes can be included without warning. In many cases the system would work as intended when a new probe was installed. In one case, at the request of the project partner, new probes were included in the API feed from a local council by the data stream provider. This automatically and prematurely fed into the live site before any notification had been received. This occurred before permission to use the data was secured, and before any review of the incoming data.
- misrepresenting sensor data. The VAS project partners typically access their data via a dashboard configured by their data-stream provider (e.g., Adcon, Wildeye, or Outpost Central). This dashboard is carefully curated by the data stream provider and only contains the information the provider thinks is useful and accurate for the project partner. It may not include all sensors, and may display particular sensors in a different way. The partner dashboard typically includes clear descriptions and units, but this information is not supplied via the API. The data-stream provider may have their own bespoke logic around how summed soil moisture is calculated at different locations. This logic is not communicated via the API, and will automatically attempt to sum all sensors from 300mm to 1000mm as summed soil moisture. As a result, access via the API has no context of what has been curated by the data stream provider and can present all the data in a ‘dumb’ manner. While technically functioning correctly, the values presented to the user in the VAS system can be very different to those they viewed in their own dashboard, eroding trust in the data sharing.