Sintetic
SINGLE ITEM IDENTIFICATION FOR FOREST PRODUCTION, PROTECTION AND MANAGEMENT

D1.5
Geospatial and platform data model, conceptual schema

Project Acronym: SINTETIC

Project name: Single item identification for forest production, protection, and management Europe

Call ID: HORIZON-CL6-2022-CIRCBIO-02-06-two-stage
(Harnessing the digital revolution in the forest-based sector)

Duration: 48 months

Starting date: 01/06/2023

Work Package: WP1

Task Number: T1.3

Lead beneficiary: LAMMA (Environmental Monitoring and Modeling Laboratory for the Sustainable Development Consortium)

Contributing beneficiary(ies): All Partners

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101082051

Sintetic

Dissemination level

[ ] PU- Public: must be available in the website

[X] CO- Confidential: Only for members of the Consortium and the Commission Services

Document history

Edition	Date	Status	Author
Version 1	15/03/2024	Draft	Manuela Corongiu (LAMMA) Stefano Romanelli (LAMMA) Raffaella Ferrari (LAMMA) Lorenzo Bottai (LAMMA) Tiziana De Filippis (CNR) Leandro Rocchi (CNR) Lorenzo Arcidiaco (CNR) Gianni Picchi (CNR) Alex Poveda (TREE) Antonio Ruano (CTFC)
Version 2	06/04/2024	Revision	Manuela Corongiu (LAMMA) Raffaella Ferrari (LAMMA) Stefano Romanelli (LAMMA) Tiziana De Filippis (CNR) Lorenzo Arcidiaco (CNR) Alex Poveda (TREE)

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101082051

Sintetic

Table of Contents

Sintetic

List of Figures

Sintetic

List of Tables

Sintetic

List of abbreviations

Acronym / Abbreviation	Meaning / Full text
API	Application Programming Interface
DBMS	DataBase Management System
DBH	Diameter at Breast Height
DDL	Data Definition Language
ETL	Extract Transform Load
FK	Foreign Key
GeoDB	GeoDataBase
GIS	Geographic Information Systems
GML	Geography Markup Language
HTTP	HyperText Transfer Protocol
INSPIRE	Infrastructure for Spatial Information in Europe
ISO	International Organization for Standardization
LiDAR	Light Detection and Ranging
JSON	JavaScript Object Notation
OGC	Open Geospatial Consortium
PK	Primary Key
PIM	Platform Independent Model
PSM	Platform Specific Model
RFID	Radio Frequency Identification
RS	Remote Sensing
SDI	Spatial Data Infrastructure
SQL	Structured Query Language

TLS	Terrestrial Laser Scanning
UML	Unified Modeling Language
UUID	Universally Unique Identifier
W3C	World Wide Web Consortium
WCS	Web Coverage Service
WFS	Web Feature Service
WMS	Web Map Service
WPS	Web Processing Service
XML	Extensible Markup Language

Sintetic

INTRODUCTION

SINTETIC aims to create a GeoDataBase that can manage data across the forest value chain while also ensuring its protection. This database will be able to connect every tree with data generated throughout the supply chain, allowing for a complete overview of the entire process, from forest inventory to finished sawn wood products. By incorporating silvicultural interventions, forest stand characteristics, and historical climatic data, the database will enable a correlation of yield and quality outcomes at any stage of the value chain with the preceding steps.

The current deliverable “D1.5: Geospatial and platform data model, conceptual schema”, is focused on geospatial data modelling that enables traceability of the entire wood value chain. This task builds on the results of Task 1.1 and Task 1.2, with the goal of establishing the structure and specific criteria for the information platform that will be deployed in WP3. The project's data model has been formalized into machine-readable language to ensure clarity and accessibility. This includes defining encodings for newly introduced machinery, sensors, data acquisition systems, and custom-developed applications within WP2. This deliverable also outlines technical specifications for the traceability system to ensure seamless continuity across the entire timber supply chain.

Standards references.

This deliverable refers to ISO TC/211 Geographic information/Geomatics Standard series and INSPIRE Directive (inspire directive 2007/2/ec) Implementing Rules and Data Specifications

ISO 19101, Geographic information/Geomatics — Reference Model

ISO 19103, Geographic information/Geomatics — Conceptual schema language

ISO 19107, Geographic information/Geomatics — Spatial Schema

ISO 19109, Geographic information/Geomatics — Rules for application schema

ISO 19110, Geographic information/Geomatics — Methodology for feature cataloguing

ISO 19115-1:2014, Geographic information/Geomatics — Metadata — Part 1: Fundamentals

INSPIRE Generic Conceptual Model Title D2.5: Generic Conceptual Model, Version 3.4

Terms and definitions

Definition from ISO Standards

1.1
application
manipulation and processing of data in support of user requirements [ISO 19101]

1.2
application schema
conceptual schema for data required by one or more applications [ISO 19101]

1.3
complex feature
feature composed of other features

1.4
conceptual formalism
set of modelling concepts used to describe a conceptual model (1.5) EXAMPLES UML meta model Note: One conceptual formalism can be expressed in several conceptual schema languages (1.7).

1.5
conceptual model
model that defines concepts of a universe of discourse [ISO 19101]

1.6
conceptual schema
formal description of a conceptual model [ISO 19101]

1.7
conceptual schema language
formal language based on a conceptual formalism (1.4) for the purpose of representing conceptual schemas (1.6) EXAMPLES UML,

Note: A conceptual schema language may be lexical or graphical. Several conceptual schema languages can be based on the same conceptual formalism.

1.8
dataset
identifiable collection of data [ISO 19115]

1.9
domain
well-defined set [ISO 19107]

NOTE Well-defined means that the definition is both necessary and sufficient, as everything that satisfies the definition is in the set and everything that does not satisfy the definition is necessarily outside the set.

1.10
feature
abstraction of real-world phenomena

NOTE A feature may occur as a type or an instance. Feature type or feature instance should be used when only one is meant. [ISO 19101]

1.11
feature association
relationship that links instances of one feature type with instances of the same or a different feature type

[ISO 19110]

1.12
feature attribute
characteristic of a feature

NOTE 1 A feature attribute may occur as a type or an instance. Feature attribute type or feature attribute instance is used when only one is meant.

NOTE 2 A feature attribute type has a name, a data type and a domain associated to it. A feature attribute instance has an attribute value taken from the domain of the feature attribute type. [adapted from ISO 19101]

1.13
feature catalogue
catalogue containing definitions and descriptions of the feature types (1.15), feature attributes (1.12) and feature relationships occurring in one or more sets of geographic data

1.14
feature instance
individual of a given feature type (1.15) having specified feature attribute (1.12) values

1.15
feature type
class of features (1.10) having common characteristics [SOURCE: ISO 19156:2011, 4.7]

1.16
geographic data/information
data with implicit or explicit reference to a location relative to the earth

NOTE Geographic information is also used as a term for information concerning phenomena implicitly or explicitly associated with a location relative to the earth.

1.17
geographic information service
service (1.27) that transforms, manages, or presents geographic information (1.16) to users

1.18
geographic information system
information system (1.19) dealing with information concerning phenomena associated with location relative to the Earth

1.19
information system
information processing system, together with associated organizational resources such as human, technical, and financial resources, that provides and distributes information

1.20
metadata
data about data [ISO 19115]

1.21

model
abstraction of some aspects of reality

1.22

ontology
formal representation of phenomena of a universe of discourse (1.28) with an underlying vocabulary including definitions and axioms that make the intended meaning explicit and describe phenomena and their interrelationships

1.23
Profile
set of one or more base standards or subsets of base standards, and, where applicable, the identification of chosen clauses, classes, options and parameters of those base standards, that are necessary for accomplishing a particular function

1.24
quality
totality of characteristics of a product that bear on its ability to satisfy stated and implied needs [ISO 19101]

1.25
reference model
framework for understanding significant relationships among the entities of some environment, and for the development of consistent standards or specifications supporting that environment

1.26
schema
formal description of a model

1.27
service
distinct part of the functionality that is provided by an entity through interfaces [SOURCE: ISO 19119:2005, 4.1]

1.28
universe of discourse
view of the real or hypothetical world that includes everything of interest [ISO 19101]

1.29
Web service
service (1.27) that is made available through the Web

Note: A Web service usually includes some combination of programming and data. It may also include human resources.

According to reference standards [ISO 19101] the Sintetic GeoDB conceptual model has been transposed from the Real Word as follow (Figure 1)

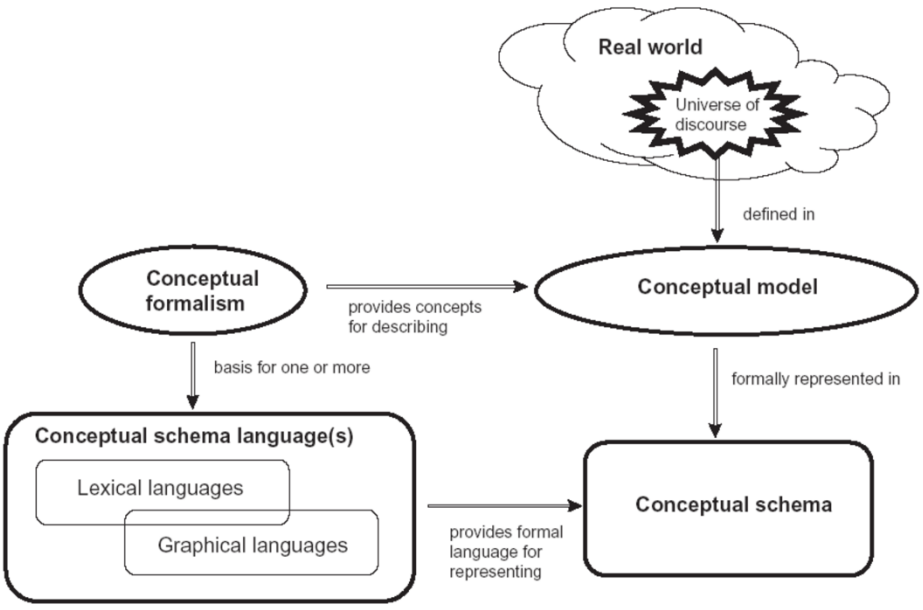


Figure 1 : From reality to conceptual schema [ISO 19101]

Then, according to reference standards [ISO 19109], the formalisation of the Sintetic GeoDB Application Schema has been designed as follow (Figure 2)

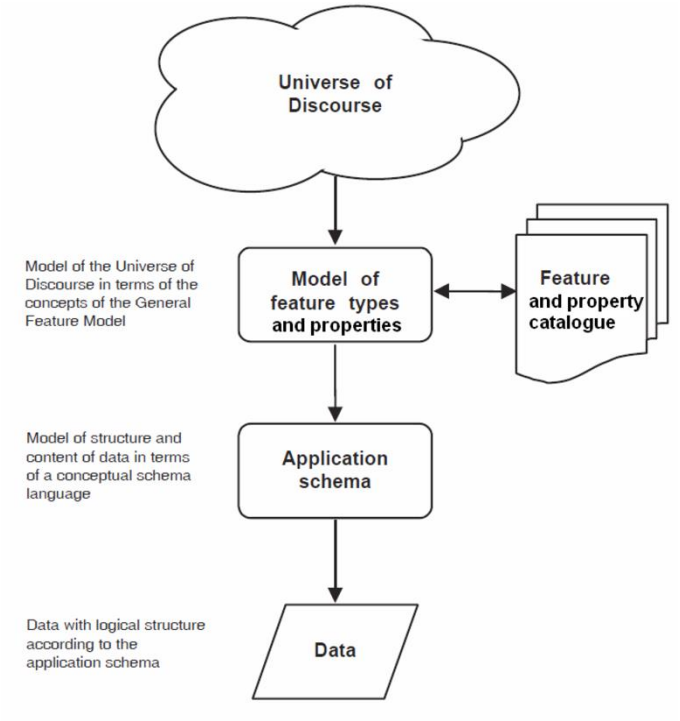


Figure 2: From reality to geographic data [ISO 19109]

The process from the conceptual to the physical Sintetic GeoDB and the use cases in Enterprise Architect

The SINTETIC GeoDB application schema was designed using Enterprise Architect (https://sparxsystems.com/), which is a powerful software tool for modelling, designing, and managing complex systems and architectures. It offers several modelling techniques, such as UML (Unified Modeling Language) and allows users to create diagrams representing different views of a system, including use case diagrams, class diagrams, sequence diagrams, and activity diagrams.

Below is the process of creating the SINTETIC GeoDB conceptual schema and its typical components.

1. **Identify Requirements**: Understand the requirements of SINTETIC database application. This involves analysing what data needs to be stored, how it will be organized, and what operations will be performed on it.
2. **Conceptual Data Model**: Start by creating a conceptual data model to represent the entities and relationships in the system. This model is independent of any specific database management system (DBMS). Information is organised in UML Structural Diagrams depicting elements of a system that are independent of time and that convey the concepts of a system and how they relate to each other. The elements in these diagrams resemble the nouns in a natural language and the relationships that connect them always show structural or semantic relationships. Contents are organised in Class of Objects and their Attributes. The class is the basic logical entity in the UML. As a structural unit of the model, it is defined by attributes. Attributes represent the properties or characteristics of the entities. Attributes provide detailed information about classes and are essential for data storage and retrieval. The model approach has been aimed to normalize our data model to reduce redundancy and improve data integrity. This involves organizing the data into multiple tables and establishing relationships between them to eliminate data duplication.
3. **GML Application Schema:** model-based engineering of spatial data and geodatabase designs to aid development of geographic information systems (GIS). It's the interoperable spatial data model using Open Geospatial Consortium's Geography Markup Language (GML) and according to INSPIRE data specifications. Contents are described through Feature Types, Data Types, a Code List and other geographic information. Once you have created your GML model, you can generate the GML Application Schema to be used in other third-party applications. Structural objects are here represented by **FeatureType**, in turn defined by attributes which Type could be default data type (String, Integer, boolean...), **DataType** (stereotype of class in Sintetic GeoDB with prefix "**dt**_", aggregation of one or more properties in turn detailed), **CodeList** (stereotype of class in Sintetic GeoDB with prefix "**dm**_"), list of encoded domain values).
4. **Model transformation**: A model transformation is a user-initiated function that starts the process of transforming one or more Platform Independent Model (PIM) elements into their corresponding Platform Specific Model (PSM) elements. The DDL (Database Definition Language) transformation has been selected considering the PostgreSQL database.
5. **Code Engineering DDL generation**: Once a physical model has been defined and the objects modeled, the Database Definition Language (DDL) for a variety of objects including database Tables, Views, Functions, Sequences and Procedures has been generated. This is a time saving mechanism and reduces the errors that can be introduced by doing this by hand in other tools. Forward engineering is governed by a set of templates that define how UML constructs are converted to the objects in the targeted DBMS (PostgreSQL). Standard templates are provided for all supported DBMSs, and these can be edited to customize the way the DDL is generated.
6. **Physical Data Model**: After creating a normalized data model, we developed a physical data model that outlines how the data will be implemented in a specific DBMS. This includes defining data types, constraints, indexes, and other database-specific details.
7. **Database Schema Generation**: Enterprise Architect's features generated the SQL code for creating the database schema based on our physical data model. This code can then be executed on PostsreSQL to create the actual database.
8. **Use case Model and Actors:** The Use Case model is a catalogue of system functionality described using UML Use Cases. Each Use Case represents a single, repeatable interaction that a user or "actor" experiences when using the system. A Use Case typically includes one or more "scenarios" which describe the interactions that go on between the Actor and the System, documenting the results and exceptions that occur from the user's perspective. Use Cases may include other Use Cases as part of a larger pattern of interaction and may also be extended by other use cases to handle exceptional conditions. **Actors** are the users of the system being modeled. Each Actor will have a well-defined role, and in the context of that role have useful interactions with the system. A person may perform the role of more than one Actor, although they will only assume one role during one use case interaction.

An Actor role may be performed by a non-human system, such as another computer program. **The primary use cases** have been defined considering Actors that interact with the proposed system. Each interaction may be specified using scenarios, sequence diagrams, communication diagrams and other dynamic diagrams or textual descriptions which together describe how the system, when viewed as a "black-box", interacts with a user.

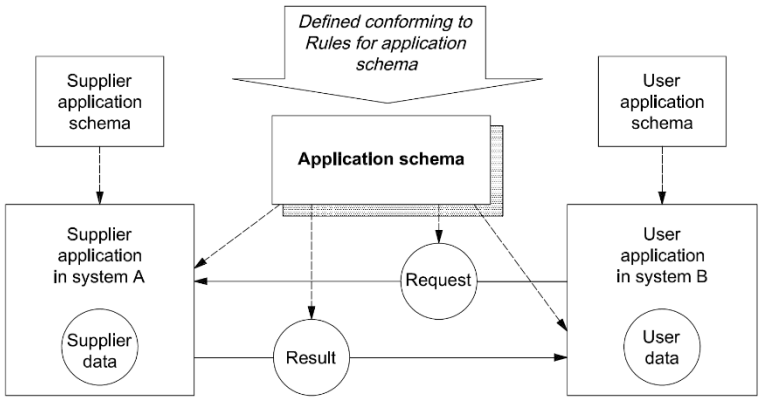
9. **Review and Refinement**: Ensure the database schema meets SINTETIC application requirements; Ensure the database schema meets SINTETIC application requirements; Refine database schema based on stakeholder feedback or requirement changes.
10. **Documentation**: The database schema documentation has been thoroughly edited using Enterprise Architect's documentation features. This documentation includes descriptions of the tables, columns, relationships, and other relevant details to help developers understand the structure of the database.

By following these steps and leveraging Enterprise Architect's features, a comprehensive GeoDB application schema that meets the needs of the SINTETIC project has been created.

Sintetic_DB: Diagram, DDL, Use Cases

Starting from the INSPIRE Consolidated UML Model (https://inspire.ec.europa.eu/data-model/approved/r4618-ir/html/, available on https://github.com/INSPIRE-MIF/uml-models), within Enterprise Architect a new package to define the application schema namely “Sintetic_DB” has been created as a specialisation of the Land Cover theme of the Inspire Directive (Annex II) to inherit the approach and functionalities of the Inspire Data Specifications.

The Sintetic GeoDB conceptual design aim to be a liaison application schema between supplier application schema and user Application schema, according to ISO 19109 suggestions (Figure 3).



NOTE The unbroken lines show the flow of data. Broken lines denote the role of the application schema on the data interchange.

Figure 3: Data interchange by transactions [ISO 19109]

This is the reason why some requirements of users appear implicitly described even though they are structured in the supplier application schema, as it happens for data coming from different phases. The following approach has been addressed to design and normalise information into the application schema to be independent to specific supplier data or user data (services on input and output data), toward sustainability during time of the Sintetic GeoDB.

The Sintetic_DB has been structured in three main sections: **GML_ApplicationSchema diagram, DDL,** and **Use Case Model (Actors and Primary Use Case)**.

GML_ApplicationSchema diagram

GML diagram in package 'Sintetic_DB'

According to the INSPIRE Consolidated UML Model, the ApplicationSchema is based on the GML geospatial model (in turn related to ISO 19136 GML specifications) so that information has been structured on

1. **Feature Type** (Stereotype <<**FeatureType**>>): basic class of objects used in the model. Relationships are formalised at conceptual level only among these (Figure 4).
2. **Attribute Type**: "**Boolean**", "**integer**", "**Real**", "**String**" default types, or customized types such as that with a "**dt**_" prefix that refers to aggregation of properties formalised in a specific **DataType** class (Stereotype <<**DataType**>>) or that a with "**dm**_" prefix that refers to domain or dictionaries formalise in a specific **CodeList** class (Stereotype <<C**odeList**>>) (Figure 4, 5, 6).

Moreover "String" attribute Type could represent Array which dimension expressed through "[]" symbol, e.g. "String [] []" indicates that the attribute is made of a 2D dimension Array (Figure 4, 5).

3. **Data Type** (Stereotype <<**DataType**>>): with a "**dt**_" prefix, as a class that aggregate more than one property (Figure 5).
4. **Code List** (Stereotype <<**CodeLit**>>): with a "**dm**_" prefix, as a class that describes dictionaries a domain of a specific attributes (Figure 6).

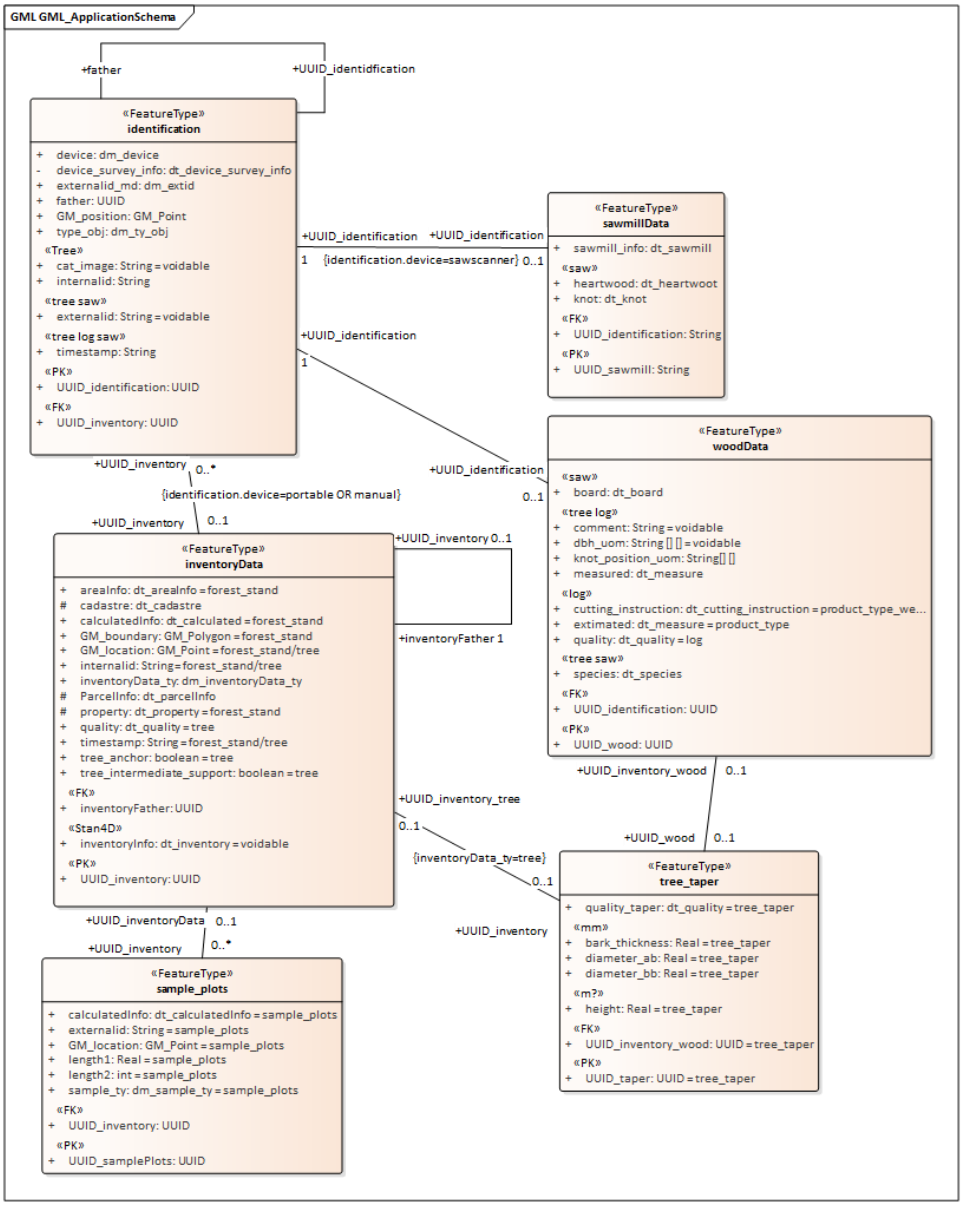


Figure 4: GML_ApplicationSchema - Feature Type Classes and Relationships

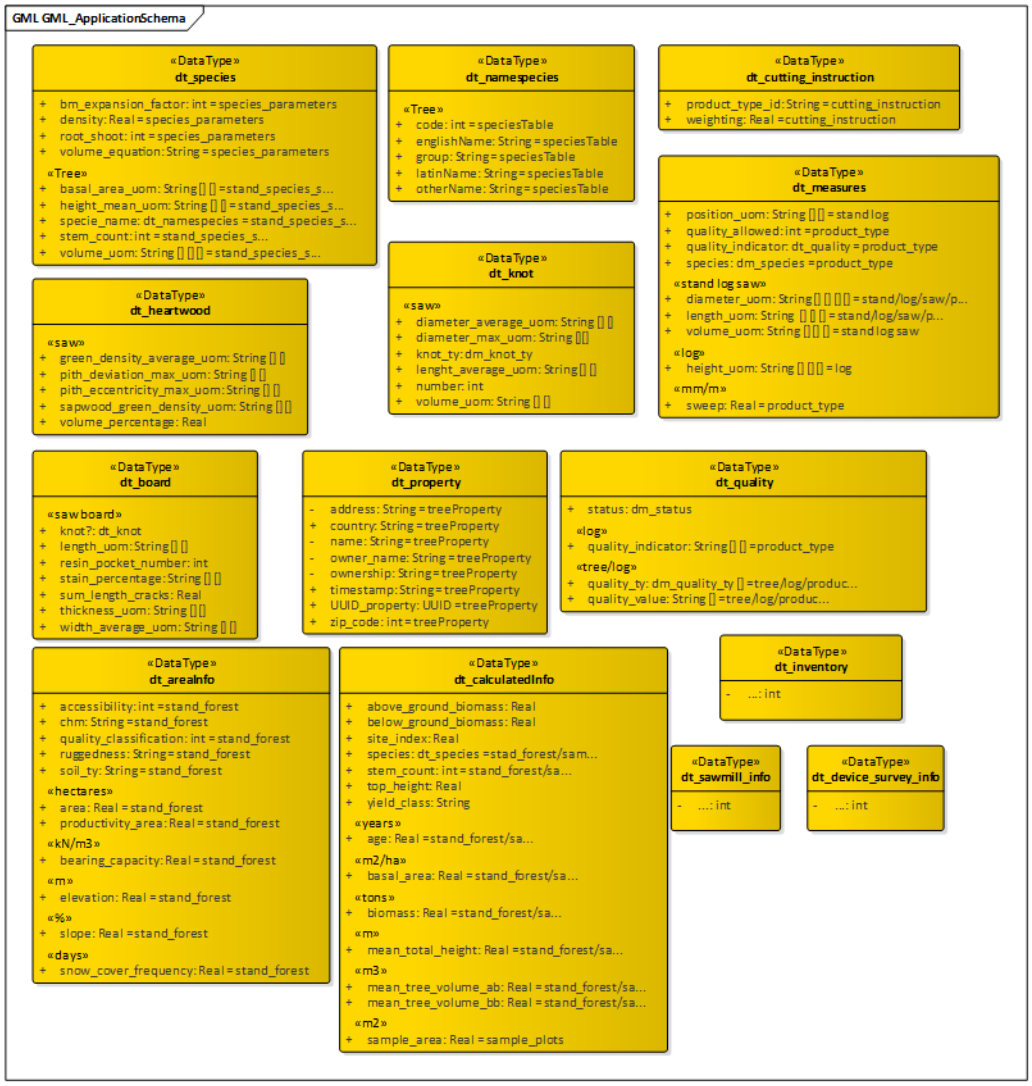


Figure 5: GML_ApplicationSchema - Data Type Classes

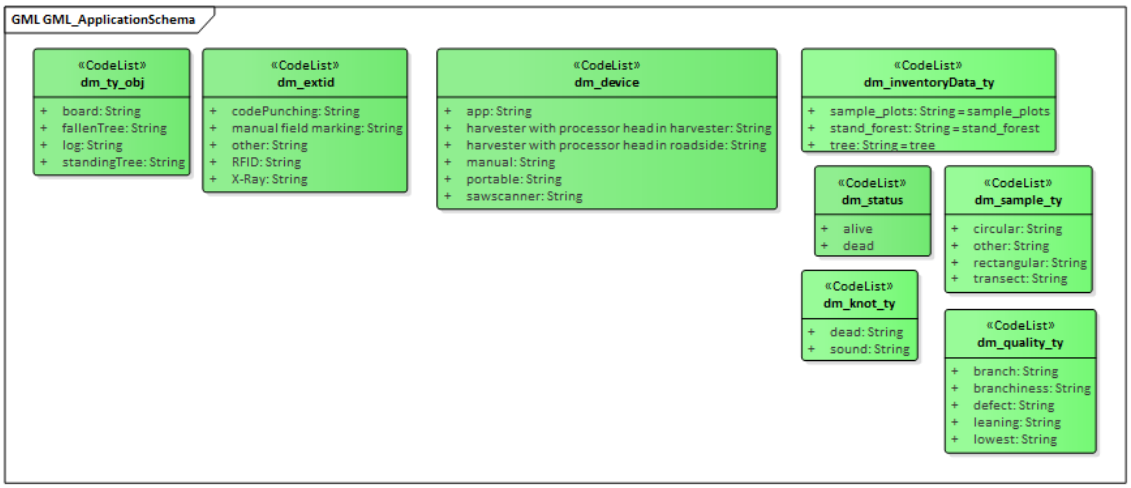


Figure 6: GML_ApplicationSchema – Code List Classes

GML_ApplicationSchema Specifications

In this paragraph a reporting in natural language of the GeoDB application schema specification has been formally detailed.

CodeList Classes

Code Lists will be physically implemented as enumerations, dictionaries, domains/subdomains of fields, or explicitly as table m:1 related to the object instance that details.

The Code List Classes are:

dm_device

Class «CodeList» in package 'Sintetic_DB'

This class provides an enumeration of possible devices based on both WP2-Prototypes and WP1-Requirements collected information.

dm_device
Version 1.0 Phase 1.0 Proposed
corongiu created on 01/02/2024. Last modified 01/02/2024

ATTRIBUTES	
app: String Public	[Is static False. Containment is Not Specified.]
harvester with processor head in harvester: String Public	[Is static False. Containment is Not Specified.]
harvester with processor head in roadside: String Public	[Is static False. Containment is Not Specified.]
manual: String Public	[Is static False. Containment is Not Specified.]
portable: String Public	[Is static False. Containment is Not Specified.]
sawscanner: String Public	[Is static False. Containment is Not Specified.]

dm_extid

Class «CodeList» in package 'Sintetic_DB'

This class provides an enumeration of possible types of identification considering both for different phases (inventory of sample trees, standing trees; harvest with fallen trees or logs, sawmill with logs or boards) and devices (App, harvester machine, manual, punching, scanner, etc.) of the supply chain, based on both WP2-Prototypes and WP1-Requirements collected information.

dm_extid
Version 1.0 Phase 1.0 Proposed
corongiu created on 01/02/2024. Last modified 07/02/2024

ATTRIBUTES	
codePunching: String Public	[Is static False. Containment is Not Specified.]
manual field marking: String Public	[Is static False. Containment is Not Specified.]
other: String Public	[Is static False. Containment is Not Specified.]
RFID: String Public	[Is static False. Containment is Not Specified.]
X-Ray: String Public	[Is static False. Containment is Not Specified.]

dm_inventoryData_ty

Class «CodeList» in package 'Sintetic_DB'

This class specify which type of inventory area refers to.

dm_inventoryData_ty
Version 1.0 Phase 1.0 Proposed
corongiu created on 27/02/2024. Last modified 28/02/2024

ATTRIBUTES	
sample_plots: String Public = sample_plots	[Sub area of a stand_forest area. Is static False. Containment is Not Specified.]
stand_forest: String Public = stand_forest	[Forest area unit or stratum. Is static False. Containment is Not Specified.]
tree: String Public = tree	[single tree inventory information. Is static False. Containment is Not Specified.]

dm_knot_ty

Class «CodeList» in package 'Sintetic_DB'

This class refers to the type of knot coming from features we could export per each product on each scanner belonging to the sawmill phase.

dm_knot_ty
Version 1.0 Phase 1.0 Proposed
corongiu created on 19/02/2024. Last modified 19/02/2024

ATTRIBUTES	
dead : String Public	[Is static False. Containment is Not Specified.]
sound : String Public	[Is static False. Containment is Not Specified.]

dm_quality_ty

Class «CodeList» in package 'Sintetic_DB'

This class enumerates some quality indicators, according to suggestions on Tree Data definition_machingGeoDB (Annex 2), the list this list should be extended.

dm_quality_ty
Version 1.0 Phase 1.0 Proposed
corongiu created on 28/02/2024. Last modified 28/02/2024

ATTRIBUTES	
branch : String Public	[Is static False. Containment is Not Specified.]
branchiness : String Public	[Is static False. Containment is Not Specified.]
defect : String Public	[Is static False. Containment is Not Specified.]
leaning : String Public	[Is static False. Containment is Not Specified.]
lowest : String Public	[Is static False. Containment is Not Specified.]

dm_sample_ty

Class «CodeList» in package 'Sintetic_DB'

This class define the type of the sample used in the inventory phase.

dm_sample_ty
Version 1.0 Phase 1.0 Proposed
corongiu created on 27/02/2024. Last modified 28/02/2024

ATTRIBUTES	
circular : String Public	[Is static False. Containment is Not Specified.]
other : String Public	[Is static False. Containment is Not Specified.]
rectangular : String Public	[Is static False. Containment is Not Specified.]
transect : String Public	[Is static False. Containment is Not Specified.]

dm_species

Class «CodeList» in package 'Sintetic_DB'

The specie domain will be specified as dictionaries directly on the implementation level (WP3) according to real surveyed data. To be defined.

dm_species
Version 1.0 Phase 1.0 Proposed
corongiu created on 01/02/2024. Last modified 01/02/2024

dm_status

Class «CodeList» in package 'Sintetic_DB'

This class defines the status of a single tree (basically on the inventory phase).

dm_status
Version 1.0 Phase 1.0 Proposed
corongiu created on 17/03/2024. Last modified 17/03/2024

ATTRIBUTES	
alive : Public	[Is static False. Containment is Not Specified.]
dead : Public	[Is static False. Containment is Not Specified.]

dm_ty_obj

Class «CodeList» in package 'Sintetic_DB'

This class define the type of object that is considered according to the phase of the supply chain.

dm_ty_obj
Version 1.0 Phase 1.0 Proposed
corongiu created on 16/02/2024. Last modified 16/02/2024

ATTRIBUTES
board : String Public [Is static False. Containment is Not Specified.]
fallenTree : String Public [Is static False. Containment is Not Specified.]
log : String Public [Is static False. Containment is Not Specified.]
standingTree : String Public [Is static False. Containment is Not Specified.]

DataType Classes

DataTypes will be physically implemented as multi-dimentional arrays, explicitly declared as a set of attributes in the reference FeatureType (or in turn into a “father” DataType), or explicitly as table 1:1 related to the FeatureType (or in turn into a “father” DataType) of object instance that details.

The Data Type Classes are:

dt_areaInfo

Class «DataType» in package 'Sintetic_DB'

This class details a set of attributes that refers to an inventory area.

For each attribute, after the “=” the initial value is detailed according to the table of Tree Data definition_machingGeoDB (Annex 2).

Moreover, stereotypes declare the used uom (Unit of Measure)

dt_areaInfo
Version 1.0 Phase 1.0 Proposed
corongiu created on 21/02/2024. Last modified 21/02/2024

ATTRIBUTES
accessibility : int Public = stand_forest accessibility code [Is static False. Containment is Not Specified.]

ATTRIBUTES

area : Real Public = stand_forest

stand area

[Stereotype is «hectares». Is static False. Containment is Not Specified.]

bearing_capacity : Real Public = stand_forest

average pressure between the harvesting machines and the soil

[Stereotype is «kN/m3». Is static False. Containment is Not Specified.]

chm : String Public = stand_forest

reference to TIFF file

[Is static False. Containment is Not Specified.]

elevation : Real Public = stand_forest

stand average/relevant elevation above sea level

[Stereotype is «m». Is static False. Containment is Not Specified.]

productivity_area : Real Public = stand_forest

Planted area (usually adjustment from maps or % of the total area)

[Stereotype is «hectares». Is static False. Containment is Not Specified.]

quality_classification : int Public = stand_forest

quality classification (to be defined)

[Is static False. Containment is Not Specified.]

ruggedness : String Public = stand_forest

stand ruggedness description

[Is static False. Containment is Not Specified.]

slope : Real Public = stand_forest

stand average/relevant slope

[Stereotype is «%». Is static False. Containment is Not Specified.]

snow_cover_frequency : Real Public = stand_forest

minimum height of snow that hampers harvesting activities

[Stereotype is «days». Is static False. Containment is Not Specified.]

soil_ty : String Public = stand_forest

ATTRIBUTES

soil type list to be defined.

[Is static False. Containment is Not Specified.]

dt_board

Class «DataType» in package 'Sintetic_DB'

This class details a set of attributes that refers to board parameters catchable during the sawmill phase.

Moreover, stereotypes detail the sawmill source of information according to intermediate meeting minutes (Annex 1 - Sawmill property quality information and link to identification - 09th of February 2024).

dt_board
Version 1.0 Phase 1.0 Proposed
corongiu created on 19/02/2024. Last modified 19/02/2024

ATTRIBUTES

knot : dt_knot Public

[DataType of a Datatype. Stereotype is «saw board». Is static False. Containment is Not Specified.]

length_uom : String [] [] Public

[attribute defined by a 2D array: First [] defines the value, the Second [] defines the uom. Stereotype is «saw board». Is static False. Containment is Not Specified.]

resin_pocket_number : int Public

[Stereotype is «saw board». Is static False. Containment is Not Specified.]

stain_percentage : String [] [] Public

Alias: brown/blue stain percentage
[attribute defined by a 2D array: First [] defines the brown stain percentage, the Second [] defines the blue stain percentage Stereotype is «saw board». Is static False. Containment is Not Specified.]

sum_length_cracks : Real Public

[Stereotype is «saw board». Is static False. Containment is Not Specified.]

thickness_uom : String [] [] Public

[attribute defined by a 2D array: First [] defines the value, the Second [] defines the uom. Stereotype is «saw board». Is static False. Containment is Not Specified.]

width_average_uom : String [] [] Public

[attribute defined by a 2D array: First [] defines the value, the Second [] defines the uom. Stereotype is «saw board». Is static False. Containment is Not Specified.]

dt_calculatedInfo

Class «DataType» in package 'Sintetic_DB'

This class details a set of attributes about parameters calculated by a post-processing phase of the achieved information.

For each attribute, after the "=" the initial value is detailed according to the table of Tree Data definition_machingGeoDB (Annex 2).

Moreover, stereotypes declare the used uom (Unit of Measure)

dt_calculatedInfo
Version 1.0 Phase 1.0 Proposed
corongiu created on 27/02/2024. Last modified 27/02/2024

ATTRIBUTES
above_ground_biomass : Real Public [Is static False. Containment is Not Specified.]
age : Real Public = stand_forest/sample_plots/tree User provided or the Average age from lower levels -in years- (e.g. 85) [Stereotype is «years». Is static False. Containment is Not Specified.]
basal_area : Real Public = stand_forest/sample_plots stand basal area [Stereotype is «m2/ha». Is static False. Containment is Not Specified.]
below_ground_biomass : Real Public [Is static False. Containment is Not Specified.]
biomass : Real Public = stand_forest/sample-plots/tree total biomass [Stereotype is «tons». Is static False. Containment is Not Specified.]
mean_total_height : Real Public = stand_forest/sample_plots/tree Mean tree height in m (e.g. 25.8) for stand_forest or sample_plots, total_height for tree [Stereotype is «m». Is static False. Containment is Not Specified.]
mean_tree_volume_ab : Real Public = stand_forest/sample_plots/tree Mean tree volume above bark in m3 (e.g. 0.82) [Stereotype is «m3». Is static False. Containment is Not Specified.]

ATTRIBUTES
mean_tree_volume_bb : Real Public = stand_forest/sample_plots/tree mean tree volume below the bark in m3 (e.g. 0.82) [Stereotype is «m3». Is static False. Containment is Not Specified.]
sample_area : Real Public = sample_plots Sample area can be calculated from plot radius/length or directly provided [Stereotype is «m2». Is static False. Containment is Not Specified.]
site_index : Real Public [Is static False. Containment is Not Specified.]
species : String Public = stad_forest/sample_plots/tree List of species and Number of stems (e.g. AA, SS, LP) [Is static False. Containment is Not Specified.]
stem_count : int Public = stand_forest/sample_plots number of stems within the stand [Is static False. Containment is Not Specified.]
top_height : Real Public [Is static False. Containment is Not Specified.]
yield_class : String Public [Is static False. Containment is Not Specified.]

dt_cutting_instruction

Class «DataType» in package 'Sintetic_DB'

This class details a set of attributes about cutting instructions.

A cutting instruction contain multiple products weighting.

For each attribute, after the "=" the initial value is detailed according to the table of Tree Data definition_machingGeoDB (Annex 2).

dt_cutting_instruction
Version 1.0 Phase 1.0 Proposed
corongiu created on 29/02/2024. Last modified 29/02/2024

ATTRIBUTES
product_type_id : String Public = cutting_instruction "product_type" object [Is static False. Containment is Not Specified.]

ATTRIBUTES
weighting : Real Public = cutting_instruction product type weighting values [Is static False. Containment is Not Specified.]

dt_device_survey_info

Class «DataType» in package 'Sintetic_DB'

This class details a set of attributes about device used to survey. To be defined directly on the implementation phase (WP3) with real data.

dt_device_survey_info
Version 1.0 Phase 1.0 Proposed
corongiu created on 17/03/2024. Last modified 17/03/2024

ATTRIBUTES
... : int Private [Is static False. Containment is Not Specified.]

dt_heartwood

Class «DataType» in package 'Sintetic_DB'.

This class details a set of attributes about Sawmill phase surveyed parameters.

Moreover, stereotypes detail the sawmill source of information according to minutes (Annex 1 - Sawmill property quality information and link to identification - 09th of February 2024). Alias defines the usual uom.

dt_heartwood
Version 1.0 Phase 1.0 Proposed
corongiu created on 19/02/2024. Last modified 19/02/2024

ATTRIBUTES
green_density_average_uom : String [] [] Public Alias: g/dm3 [attribute defined by a 2D array: First [] defines the value, the Second [] defines the uom. Stereotype is «saw». Is static False. Containment is Not Specified.]
pith_deviation_max_uom : String [] [] Public Alias: mm [attribute defined by a 2D array: First [] defines the value, the Second [] defines the uom. Stereotype is «saw». Is static False. Containment is Not Specified.]
pith_eccentricity_max_uom : String [] [] Public Alias: mm [attribute defined by a 2D array: First [] defines the value, the Second [] defines the uom. Stereotype is «saw». Is static False. Containment is Not Specified.]

ATTRIBUTES
sapwood_green_density_uom : String [] [] Public Alias: g/dm3 [attribute defined by a 2D array: First [] defines the value, the Second [] defines the uom. Stereotype is «saw». Is static False. Containment is Not Specified.]
volume_percentage : Real Public [Stereotype is «saw». Is static False. Containment is Not Specified.]

dt_inventory

Class «DataType» in package 'Sintetic_DB'

This class details a set of attributes about the inventory phase. To be defined directly on the implementation phase (WP3) with real data.

dt_inventory
Version 1.0 Phase 1.0 Proposed
corongiu created on 21/02/2024. Last modified 21/02/2024

ATTRIBUTES
... : int Private [Is static False. Containment is Not Specified.]

dt_knot

Class «DataType» in package 'Sintetic_DB'

This class details a set of attributes about Sawmill phase surveyed parameters.

Moreover, stereotypes detail the sawmill source of information according to minutes (Annex 1 - Sawmill property quality information and link to identification - 09th of February 2024).

dt_knot
Version 1.0 Phase 1.0 Proposed
corongiu created on 19/02/2024. Last modified 19/02/2024

ATTRIBUTES
diameter_average_uom : String [] [] Public [attribute defined by a 2D array: First [] defines the value, the Second [] defines the uom. Stereotype is «saw». Is static False. Containment is Not Specified.]
diameter_max_uom : String [] [] Public [attribute defined by a 2D array: First [] defines the value, the Second [] defines the uom. Stereotype is «saw». Is static False. Containment is Not Specified.]
knot_ty : dm_knot_ty Public [Stereotype is «saw». Is static False. Containment is Not Specified.]

ATTRIBUTES
lenght_average_uom : String [] [] Public [attribute defined by a 2D array: First [] defines the value, the Second [] defines the uom. Stereotype is «saw». Is static False. Containment is Not Specified.]
number : int Public [Stereotype is «saw». Is static False. Containment is Not Specified.]
volume_uom : String [] [] Public [attribute defined by a 2D array: First [] defines the value, the Second [] defines the uom. Stereotype is «saw». Is static False. Containment is Not Specified.]

dt_measures

Class «DataType» in package 'Sintetic_DB'

This class details a set of attributes about measures both observed or estimated. Stereotypes describe the origin phases of the information (stand=stand_forest inventory info, log=log of a tree, saw=sawmill) measures or its UoM. Because of the use of more than two dimensions, notes indicate each array dimension and what information refers to.

For each attribute, after the "=" the initial value is detailed according to the table of Tree Data definition_machingGeoDB (Annex 2).

dt_measures
Version 1.0 Phase 1.0 Proposed
corongiu created on 20/02/2024. Last modified 20/02/2024

ATTRIBUTES
diameter_uom : String [] [] [] [] Public = stand/log/saw/product_type [value] [uom] [min/max] [LED/SED] [Stereotype is «stand log saw». Is static False. Containment is Not Specified.]
height_uom : String [] [] [] Public = log [value] [uom] [top/bottom] [Stereotype is «log». Is static False. Containment is Not Specified.]
length_uom : String [] [] [] Public = stand/log/saw/product_type [value] [uom] [min/max] [Stereotype is «stand log saw». Is static False. Containment is Not Specified.]
max_sweep : Real Public = product_type Maximum sweep in the log product

ATTRIBUTES
Constraints: woodData.eximated : [Stereotype is «mm/m». Is static False. Containment is Not Specified.]
position_uom : String [] [] Public = stand log [Is static False. Containment is Not Specified.]
quality_allowed : int Public = product_type Type of defect allowed for his type of product (1,2,3) [Is static False. Containment is Not Specified.]
quality_indicator : dt_quality Public = product_type [Is static False. Containment is Not Specified.]
species : dm_species Public = product_type Constraints: woodData.eximated : OCL [Is static False. Containment is Not Specified.]
volume_uom : String [] [] [] Public = stand log saw [value] [uom] [ab/bb] [Stereotype is «stand log saw». Is static False. Containment is Not Specified.]

dt_namespecies

Class «DataType» in package 'Sintetic_DB'

This class details a set of attributes about namespace in different languages related to a unique code identification.

For each attribute, after the "=" the initial value is detailed according to the table of Tree Data definition_machingGeoDB (Annex 2)

dt_namespecies
Version 1.0 Phase 1.0 Proposed
corongiu created on 20/02/2024. Last modified 20/02/2024

ATTRIBUTES
code : int Public = speciesTable [Stereotype is «Tree». Is static False. Containment is Not Specified.]
englishName : String Public = speciesTable [Stereotype is «Tree». Is static False. Containment is Not Specified.]
group : String Public = speciesTable [Stereotype is «Tree». Is static False. Containment is Not Specified.]

ATTRIBUTES
latinName : String Public = speciesTable [Stereotype is «Tree». Is static False. Containment is Not Specified.]
otherName : String Public = speciesTable [Stereotype is «Tree». Is static False. Containment is Not Specified.]

dt_property

Class «DataType» in package 'Sintetic_DB'

This class details a set of attributes about the location and owner information. Sometimes attributes refer to “Private” instead of “Public” information to must be blinded according to the GDPR 2016/679 regulation.

dt_property
Version 1.0 Phase 1.0 Proposed
corongiu created on 21/02/2024. Last modified 21/02/2024

ATTRIBUTES
address : String Private = treeProperty address of the forest [Is static False. Containment is Not Specified.]
country : String Public = treeProperty country where the forest is located [Is static False. Containment is Not Specified.]
name : String Private = treeProperty [Is static False. Containment is Not Specified.]
owner_name : String Private = treeProperty Name of the owner [Is static False. Containment is Not Specified.]
ownership : String Private = treeProperty Type of ownership [Is static False. Containment is Not Specified.]
timestamp : String Public = treeProperty time and date when the property is created [Is static False. Containment is Not Specified.]

ATTRIBUTES
UUID_property : UUID Public = treeProperty Universally Unique IDentifier created when information is uploaded into DB [Is static False. Containment is Not Specified.]
zip_code : int Public = treeProperty zip code of the forest [Is static False. Containment is Not Specified.]

dt_quality

Class «DataType» in package 'Sintetic_DB'

This class details a set of attributes about quality that are defined in inventory tree data as well as log product definition. In the implementation phase, it should evaluate if distinguishing between two different domains (for inventory or log product scopes) is better.

Stereotypes identify the object phase to be associated with.

For each attribute, after the "=" the initial value is detailed according to the table of Tree Data definition_machingGeoDB (Annex 2)

dt_quality
Version 1.0 Phase 1.0 Proposed
corongiu created on 28/02/2024. Last modified 28/02/2024

ATTRIBUTES
quality_indicator : String [] [] Public = product_type [Stereotype is «log». Is static False. Containment is Not Specified.]
quality_ty : dm_quality_ty [] Public = tree/log/product_type Allow for multiple quality parameter per log [Stereotype is «tree/log». Is static False. Containment is Not Specified.]
quality_value : String [] [] Public = tree/log/product_type [value] [min/max] [Stereotype is «tree/log». Is static False. Containment is Not Specified.]
status : dm_status Public [Is static False. Containment is Not Specified.]

dt_sawmill_info

Class «DataType» in package 'Sintetic_DB'

To be completed directly on the implementation level with the real additional information will be supplied by the sawmill in WP4 - demonstrations.

dt_sawmill_info
Version 1.0 Phase 1.0 Proposed
corongiu created on 17/03/2024. Last modified 17/03/2024

ATTRIBUTES

- ... : int Private
  [Is static False. Containment is Not Specified.]

dt_species

Class «DataType» in package 'Sintetic_DB'

This class details a set of attributes about quality parameters associated to each specie as reference in inventory as well as suggestion in harvesting phase. In the implementation phase, it should evaluate if distinguishing between two different domains (for inventory or harvesting scopes) is better.

This table refers to calculated info even when used out of Dt_calculatedinfo.

Stereotypes identify the object phase to be associated with. The “Alias” sometimes Identifies the UoM.

For each attribute, after the “=” the initial value is detailed according to the table of Tree Data definition_machingGeoDB (Annex 2)

dt_species
Version 1.0 Phase 1.0 Proposed
corongiu created on 20/02/2024. Last modified 20/02/2024

ATTRIBUTES

- basal_area_uom : String [] [] Public = stand_species_summaryTable
  Alias: m2/ha
  [Stereotype is «Tree». Is static False. Containment is Not Specified.]

- bm_expansion_factor : int Public = species_parameters

  Biomass expansion factor (used for biomass calculation)
  [Is static False. Containment is Not Specified.]

- density : Real Public = species_parameters

  Wood density used for quality and biomass calculations
  [Is static False. Containment is Not Specified.]

- height_mean_uom : String [] [] Public = stand_species_summaryTable
  Alias: m
  [Stereotype is «Tree». Is static False. Containment is Not Specified.]

ATTRIBUTES
root_shoot : int Public = species_parameters Root to shoot ratio (used for biomass calculation) [Is static False. Containment is Not Specified.]
specie_name : dt_namespecies Public = stand_species_summaryTable [Stereotype is «Tree». Is static False. Containment is Not Specified.]
stem_count : int Public = stand_species_summaryTable [Stereotype is «Tree». Is static False. Containment is Not Specified.]
volume_equation : String Public = species_parameters Taper equation [Is static False. Containment is Not Specified.]
volume_uom : String [] [] [] Public = stand_species_summaryTable Alias: m3 [value] [uom] [ab/bb/mean] [Stereotype is «Tree». Is static False. Containment is Not Specified.]

FeatureType Classes

FeatureTypes are classes of features having common characteristics [SOURCE: ISO 19156:2011, 4.7].

A Feature class is the main structural element for describing an application schema. In this context, the FeatureTypes represent the main phases of the wood supply chain: inventory phase, wood harvesting phase, wood processing phase in the sawmill. All these phases are associated with a single FeatureType which uniquely identifies the instance and locates it at a certain time and in a certain place in order to guarantee traceability.

Stereotypes have been used to specify the key types (PK for Primary Key, FK for Foreign Key) enrolled in associations.

The Data Type Classes are:

identification

Class «FeatureType» in package 'Sintetic_DB'

This class describes the properties associated with the unique identification of the "woodData" object that you want to describe. Identification occurs externally to the database with specific phase-dependent methods (via RFID, manual identification, via scanner etc.), then in this class there is the association of each phase-specific

identification system with the Universal Unique ID (UUID)[1]. The UUID is system-independent identification carried out when the information is inserted into the DB. Furthermore, in this class a recursive relationship is formalized which allows the wood traceability chain to be reconstructed. For this reason, the identification phase is associated with the three main phases of the wood value chain: the inventory phase, the wood harvesting phase, the sawmill processing phase.

Stereotypes identify the object phase to be associated with.

For each attribute, after the "=" the initial value is detailed when the attribute could be "voidable"

identification
Version 1.0 Phase 1.0 Proposed
corongiu created on 01/02/2024. Last modified 01/02/2024

ATTRIBUTES
cat_image : String Public = voidable [Stereotype is «Tree». Is static False. Containment is Not Specified.]
device : dm_device Public [Is static False. Containment is Not Specified.]
device_survey_info : dt_device_survey_info Private [Is static False. Containment is Not Specified.]
externalid : String Public = voidable Alias: RFID, XRay, Punching [Stereotype is «tree saw». Is static False. Containment is Not Specified.]
externalid_md : dm_extid Public [Is static False. Containment is Not Specified.]
father : UUID Public [Is static False. Containment is Not Specified.]
GM_position : GM_Point Public [Is static False. Containment is Not Specified.]
internalid : String Public Alias: GUID [Stereotype is «Tree». Is static False. Containment is Not Specified.]

1 ISO/IEC 9834-8:2014 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 https://www.iso.org/standard/62795.html

ATTRIBUTES

timestamp : String Public

on standing tree consider only the last one related to the RFID positioning

[Stereotype is «tree log saw». Is static False. Containment is Not Specified.]

type_obj : dm_ty_obj Public

[Is static False. Containment is Not Specified.]

UUID_identification : UUID Public

[Stereotype is «PK». Is static False. Containment is Not Specified.]

UUID_inventory : UUID Public

[Stereotype is «FK». Is static False. Containment is Not Specified.]

ASSOCIATIONS

Association (direction: Unspecified)

Constraints:
identification.device=portable OR manual : Invariant

Source: Public UUID_inventory (Class) identification «FeatureType»
Cardinality: [0..*]

Target: Public UUID_inventory (Class) inventoryData «FeatureType»
Cardinality: [0..1]

Association (direction: Unspecified)

Source: Public UUID_identidfication (Class) identification «FeatureType»

Target: Public father (Class) identification «FeatureType»

Association (direction: Unspecified)

Source: Public UUID_identification (Class) identification «FeatureType»
Cardinality: [1]

Target: Public UUID_identification (Class) woodData «FeatureType»
Cardinality: [0..1]

Association (direction: Unspecified)

Constraints:
identification.device=sawscanner : Invariant

Source: Public UUID_identification (Class) identification «FeatureType»
Cardinality: [1]

Target: Public UUID_identification (Class) sawmillData «FeatureType»
Cardinality: [0..1]

Association (direction: Unspecified)

Source: Public UUID_identidfication (Class) identification «FeatureType»

Target: Public father (Class) identification «FeatureType»

inventoryData

Class «FeatureType» in package 'Sintetic_DB'

This class describes the information relating to the inventory creation phase at a specific time and in a specific area or relating to each individual tree.

The association between forest areas, sample areas or even of a single tree are obtained through a recursive relationship that allows us to know, for example in which sample area a certain tree is positioned or in which forest property a certain area is geolocated.

Furthermore, some attributes relating to forest owners are defined here. In compliance with the GDPR 2016/679 privacy regulation, this information may have restricted access.

Finally, some information is suggestions of the standard characteristics referring to each forest species considered to facilitate the inventory editing phase.

For each attribute, after the "=" the initial value is detailed according to the table of Tree Data definition_machingGeoDB (Annex 2)

inventoryData
Version 1.0 Phase 1.0 Proposed
corongiu created on 01/02/2024. Last modified 07/02/2024

ATTRIBUTES
areaInfo : dt_areaInfo Public = forest_stand [Is static False. Containment is Not Specified.]
cadastre : dt_cadastre Protected [Is static False. Containment is Not Specified.]
calculatedInfo : dt_calculated Public = forest_stand [Is static False. Containment is Not Specified.]
GM_boundary : GM_Polygon Public = forest_stand stand boundary [Is static False. Containment is Not Specified.]
GM_location : GM_Point Public = forest_stand/tree stand location, POINT object [Is static False. Containment is Not Specified.]
internalid : String Public = forest_stand/tree [Is static False. Containment is Not Specified.]
inventoryData_ty : dm_inventoryData_ty Public [Is static False. Containment is Not Specified.]

ATTRIBUTES

inventoryFather : UUID Public
[Stereotype is «FK». Is static False. Containment is Not Specified.]

inventoryInfo : dt_inventory Public = voidable
[Stereotype is «Stan4D». Is static False. Containment is Not Specified.]

ParcelInfo : dt_parcelInfo Protected
[Is static False. Containment is Not Specified.]

property : dt_property Protected = forest_stand
[Is static False. Containment is Not Specified.]

quality : dt_quality Public = tree

Allow for multiple quality parameter per tree
[Is static False. Containment is Not Specified.]

timestamp : String Public = forest_stand/tree

timestamp when the stan_forest or tree is created
[Is static False. Containment is Not Specified.]

tree_anchor : boolean Public = tree
[Is static False. Containment is Not Specified.]

tree_intermediate_support : boolean Public = tree
[Is static False. Containment is Not Specified.]

UUID_inventory : UUID Public
[Stereotype is «PK». Is static False. Containment is Not Specified.]

ASSOCIATIONS

Association (direction: Unspecified)

Source: Public UUID_inventoryData (Class) inventoryData «FeatureType»
Cardinality: [0..1]

Target: Public UUID_inventory (Class) sample_plots «FeatureType»
Cardinality: [0..*]

Association (direction: Unspecified)

Constraints:
inventoryData_ty=tree : Invariant

ASSOCIATIONS	
Source: Public UUID_inventory_tree (Class) inventoryData «FeatureType» Cardinality: [0..1]	Target: Public UUID_inventory (Class) tree_taper «FeatureType» Cardinality: [0..1]
Association (direction: Unspecified) Source: Public inventoryFather (Class) inventoryData «FeatureType» Cardinality: [1]	 Target: Public UUID_inventory (Class) inventoryData «FeatureType» Cardinality: [0..1]
Association (direction: Unspecified) Constraints: identification.device=portable OR manual : Invariant Source: Public UUID_inventory (Class) identification «FeatureType» Cardinality: [0..*]	 Target: Public UUID_inventory (Class) inventoryData «FeatureType» Cardinality: [0..1]
Association (direction: Unspecified) Source: Public inventoryFather (Class) inventoryData «FeatureType» Cardinality: [1]	 Target: Public UUID_inventory (Class) inventoryData «FeatureType» Cardinality: [0..1]

sample_plots

Class «FeatureType» in package 'Sintetic_DB'

This class describes the additional information of sample-plots that could be associate to inventory data just to compare parameters between sample and inventory. For this reason, an association is defined between them.

For each attribute, after the "=" the initial value is detailed according to the table of Tree Data definition_machingGeoDB (Annex 2)

sample_plots
Version 1.0 Phase 1.0 Proposed
corongiu created on 27/02/2024. Last modified 27/02/2024

ATTRIBUTES	
calculatedInfo : dt_calculatedInfo Public = sample_plots	[Is static False. Containment is Not Specified.]
externalid : String Public = sample_plots	[Is static False. Containment is Not Specified.]
GM_location : GM_Point Public = sample_plots	[Is static False. Containment is Not Specified.]

ATTRIBUTES

length1 : Real Public = sample_plots

Radius or length of the plot

[Is static False. Containment is Not Specified.]

length2 : int Public = sample_plots

Second side length for rectangular plot

[Is static False. Containment is Not Specified.]

sample_ty : dm_sample_ty Public = sample_plots

[Is static False. Containment is Not Specified.]

UUID_inventory : UUID Public

[Stereotype is «FK». Is static False. Containment is Not Specified.]

UUID_samplePlots : UUID Public

[Stereotype is «PK». Is static False. Containment is Not Specified.]

ASSOCIATIONS

Association (direction: Unspecified)

Source: Public UUID_inventoryData (Class) inventoryData «FeatureType»
Cardinality: [0..1]

Target: Public UUID_inventory (Class) sample_plots «FeatureType»
Cardinality: [0..*]

sawmillData

Class «FeatureType» in package 'Sintetic_DB'

This class describes properties of data coming from Sawmill phase. first requirements have been derived from the sawmill source of information according to intermediate meeting minutes (Annex 1 - Sawmill property quality information and link to identification - 09th of February 2024) and identified as stereotypes.

To be completed directly on the implementation level with the real additional information will be supplied by WP2-Prototypes and WP4 - demonstrations.

sawmillData
Version 1.0 Phase 1.0 Proposed
corongiu created on 01/02/2024. Last modified 07/02/2024

ATTRIBUTES

heartwood : dt_heartwoot Public

[Stereotype is «saw». Is static False. Containment is Not Specified.]

knot : dt_knot Public

ATTRIBUTES

[Stereotype is «saw». Is static False. Containment is Not Specified.]

sawmill_info : dt_sawmill Public

[Is static False. Containment is Not Specified.]

UUID_identification : String Public

[Stereotype is «FK». Is static False. Containment is Not Specified.]

UUID_sawmill : String Public

[Stereotype is «PK». Is static False. Containment is Not Specified.]

ASSOCIATIONS

Association (direction: Unspecified)

Constraints:
identification.device=sawscanner : Invariant

Source: Public UUID_identification (Class) identification «FeatureType»
Cardinality: [1]

Target: Public UUID_identification (Class) sawmillData «FeatureType»
Cardinality: [0..1]

tree_taper

Class «FeatureType» in package 'Sintetic_DB'

This Class details additional properties that could be inserted during both inventory and WoodData achievements.
For this reason, the respective associations have been defined.

Sterotypes define UoM or the type of keys (Primary or Foreign) to manage related associations.

(DBH-Height description)

tree_taper
Version 1.0 Phase 1.0 Proposed
corongiu created on 28/02/2024. Last modified 28/02/2024

ATTRIBUTES

bark_thickness : Real Public = tree_taper

Bark thickness for given diameter
Needed to estimate without bark values
calculated or provided

[Stereotype is «mm». Is static False. Containment is Not Specified.]

diameter_ab : Real Public = tree_taper

Diameter above bark

[Stereotype is «mm». Is static False. Containment is Not Specified.]

ATTRIBUTES

diameter_bb : Real Public = tree_taper

Diameter below bark

[Stereotype is «mm». Is static False. Containment is Not Specified.]

height : Real Public = tree_taper

[Stereotype is «m?». Is static False. Containment is Not Specified.]

quality_taper : dt_quality Public = tree_taper

[Is static False. Containment is Not Specified.]

UUID_inventory_wood : UUID Public = tree_taper

from inventoyData or woodData featureType

[Stereotype is «FK». Is static False. Containment is Not Specified.]

UUID_taper : UUID Public = tree_taper

[Stereotype is «PK». Is static False. Containment is Not Specified.]

ASSOCIATIONS

Association (direction: Unspecified)

Source: Public UUID_inventory_wood (Class) woodData «FeatureType»
Cardinality: [0..1]

Target: Public UUID_wood (Class) tree_taper «FeatureType»
Cardinality: [0..1]

Association (direction: Unspecified)

Constraints:
inventoryData_ty=tree : Invariant

Source: Public UUID_inventory_tree (Class) inventoryData «FeatureType»
Cardinality: [0..1]

Target: Public UUID_inventory (Class) tree_taper «FeatureType»
Cardinality: [0..1]

woodData

Class «FeatureType» in package 'Sintetic_DB'

This class specifies the parameters that can be associated with the wood both in the harvesting phase and in the sawmill processing phase.

Measured or estimated quality parameters can be detailed, comparing them in real time and in the same table. The estimated measurements refer to the productType you want to obtain.

The associations with the FeatureTypes "identification" and "tree_taper" are defined.

For each attribute, after the "=" the initial value is detailed according to the table of Tree Data definition_machingGeoDB (Annex 2).

The stereotype indicates the phase of the wood to which the instance can refer or type of keys (Primary or Foreign) to manage related associations.

woodData
Version 1.0 Phase 1.0 Proposed
corongiu created on 31/01/2024. Last modified 01/02/2024

ATTRIBUTES
board : dt_board Public [Stereotype is «saw». Is static False. Containment is Not Specified.]
comment : String Public = voidable [Stereotype is «tree log». Is static False. Containment is Not Specified.]
cutting_instruction : dt_cutting_instruction Public = product_type_weightings cutting instruction name [Stereotype is «log». Is static False. Containment is Not Specified.]
dbh_uom : String [] [] Public = voidable [attribute defined by a 2D array: First [] defines the value, the Second [] defines the uom. Stereotype is «tree log». Is static False. Containment is Not Specified.]
extimated : dt_measure Public = product_type [Stereotype is «log». Is static False. Containment is Not Specified.]
knot_position_uom : String[] [] Public [attribute defined by a 2D array: First [] defines the value, the Second [] defines the uom. Stereotype is «tree log». Is static False. Containment is Not Specified.]
measured : dt_measure Public [Stereotype is «tree log». Is static False. Containment is Not Specified.]
quality : dt_quality Public = log [Stereotype is «log». Is static False. Containment is Not Specified.]
species : dt_species Public [Stereotype is «tree saw». Is static False. Containment is Not Specified.]
UUID_identification : UUID Public [Stereotype is «FK». Is static False. Containment is Not Specified.]
UUID_wood : UUID Public [Stereotype is «PK». Is static False. Containment is Not Specified.]

ATTRIBUTES

ASSOCIATIONS

Association (direction: Unspecified)

Source: Public UUID_inventory_wood (Class) woodData «FeatureType»
Cardinality: [0..1]

Target: Public UUID_wood (Class) tree_taper «FeatureType»
Cardinality: [0..1]

Association (direction: Unspecified)

Source: Public UUID_identification (Class) identification «FeatureType»
Cardinality: [1]

Target: Public UUID_identification (Class) woodData «FeatureType»
Cardinality: [0..1]

DDL

Package in package 'Sintetic_DB'

A Model Transformation is done by initiated function that starts the process of transforming one or more Platform Independent Model (PIM) elements into their corresponding Platform Specific Model (PSM) elements. This process takes place by the rules that have been codified in the Transformation Templatesmation Templates, considering the PostgreSQL Platform. The transformation initiated by selecting a Package named “DDL” where tform-specific table elements represent the translation of the GML ApplicationSchema class elements of the platform-independent model. Because of there is a direct correspondence between UML classes and DDL tables, the general descriptions of the tables have not been reported here, as they can be considered the same as the classes from which they were derived.

The structure of the DDL package is always made by a diagram and a list of tables within.

DDL
Version 1.0 Phase 1.0 Proposed
corongiu created on 29/02/2024. Last modified 29/02/2024

DDL diagram

Class diagram in package 'DDL'

It’s the class that includes a list of the translated tables.

DDL
Version 1.0
corongiu created on 29/02/2024. Last modified 17/03/2024

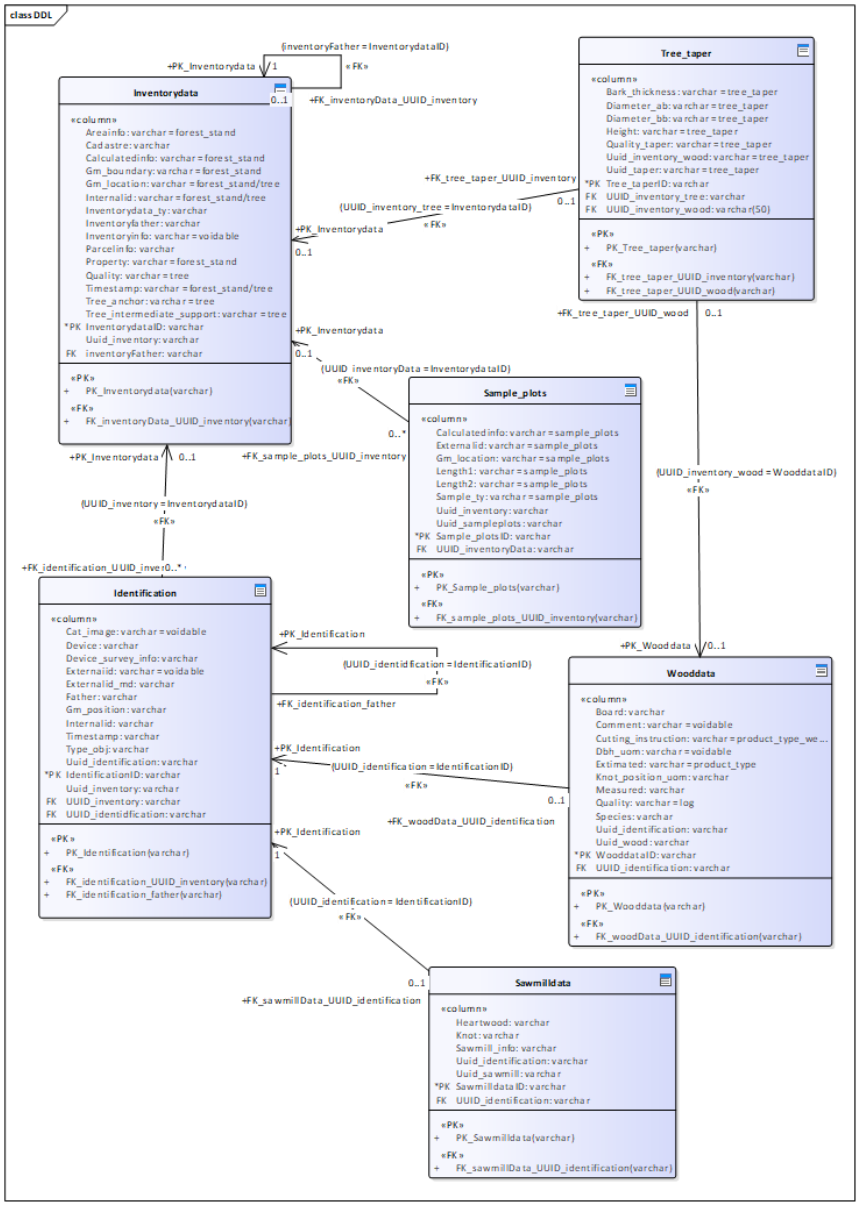


Figure 3: DDL Diagram – FeatureType transformed tables

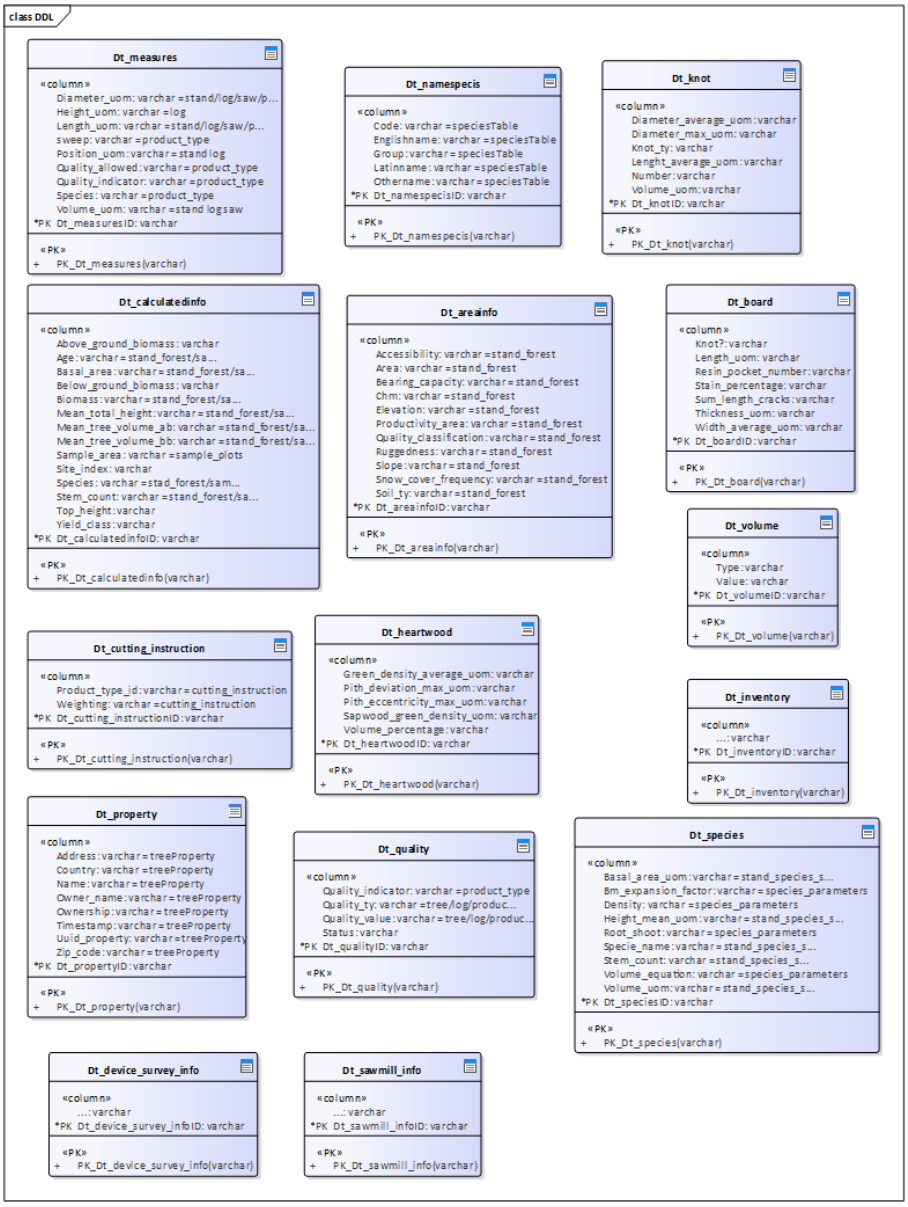


Figure 4: DDL Diagram – DataType transformed tables

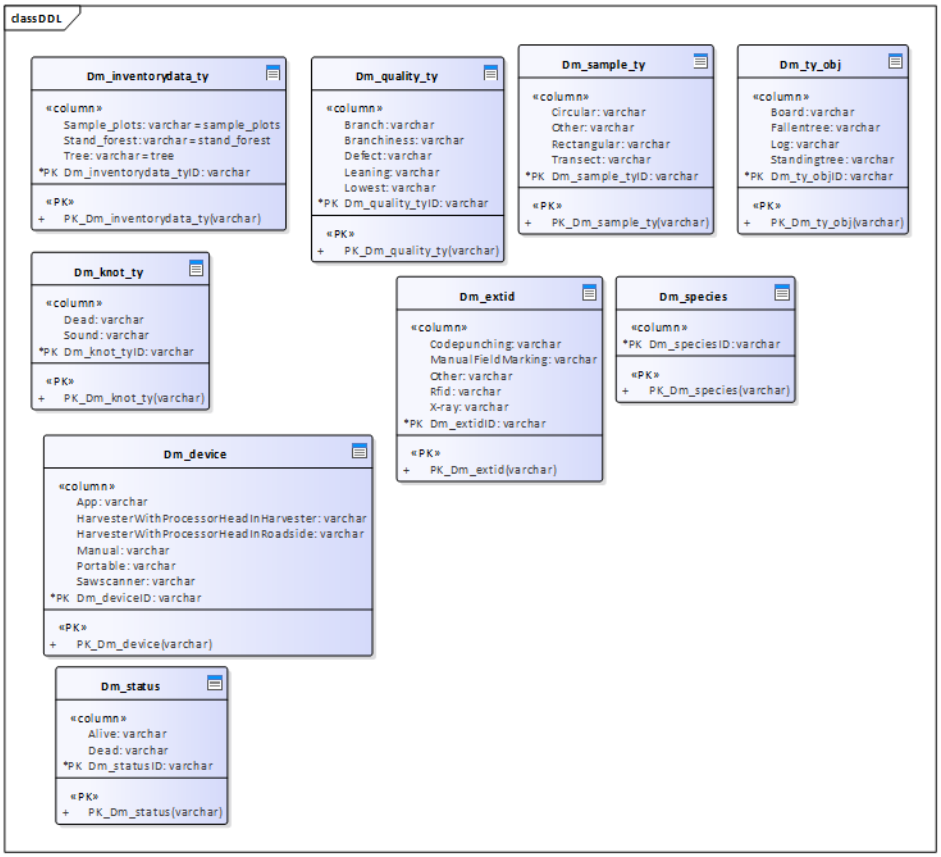


Figure 5: DDL Diagram – CodeList transformed tables

Dm_device

Database table in package 'DDL'

Dm_device
Version 1.0 Phase 1.0 Proposed
corongiu created on 29/02/2024. Last modified 17/03/2024
DBMS PostgreSQL

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
App	varchar	False	
HarvesterWithProcesorHeadInHarvester	varchar	False	
HarvesterWithProcesorHeadInRoadside	varchar	False	
Manual	varchar	False	
Portable	varchar	False	

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
Sawscanner	varchar	False	
Dm_deviceID	varchar	True	

PRIMARY KEY NAME	COLUMNS	COMMENTS
PK_Dm_device	Dm_deviceID	

Dm_extid

Database table in package 'DDL'

Dm_extid
Version 1.0 Phase 1.0 Proposed
corongiu created on 29/02/2024. Last modified 17/03/2024
DBMS PostgreSQL

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
Codepunching	varchar	False	
ManualFieldMarking	varchar	False	
Other	varchar	False	
Rfid	varchar	False	
X-ray	varchar	False	
Dm_extidID	varchar	True	

PRIMARY KEY NAME	COLUMNS	COMMENTS
PK_Dm_extid	Dm_extidID	

Dm_inventorydata_ty

Database table in package 'DDL'

Dm_inventorydata_ty
Version 1.0 Phase 1.0 Proposed
corongiu created on 29/02/2024. Last modified 17/03/2024
DBMS PostgreSQL

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
Sample_plots	varchar	False	Initial value: sample_plots
Stand_forest	varchar	False	Initial value: stand_forest
Tree	varchar	False	Initial value: tree
Dm_inventorydata_tyID	varchar	True	

PRIMARY KEY NAME	COLUMNS	COMMENTS
PK_Dm_inventorydata_ty	Dm_inventorydata_tyID	

Dm_knot_ty

Database table in package 'DDL'

Dm_knot_ty
Version 1.0 Phase 1.0 Proposed
corongiu created on 29/02/2024. Last modified 17/03/2024
DBMS PostgreSQL

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
Dead	varchar	False	
Sound	varchar	False	
Dm_knot_tyID	varchar	True	

PRIMARY KEY NAME	COLUMNS	COMMENTS
PK_Dm_knot_ty	Dm_knot_tyID	

Dm_quality_ty

Database table in package 'DDL'

Dm_quality_ty
Version 1.0 Phase 1.0 Proposed
corongiu created on 29/02/2024. Last modified 17/03/2024
DBMS PostgreSQL

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
Branch	varchar	False	

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
Branchiness	varchar	False	
Defect	varchar	False	
Leaning	varchar	False	
Lowest	varchar	False	
Dm_quality_tyID	varchar	True	

PRIMARY KEY NAME	COLUMNS	COMMENTS
PK_Dm_quality_ty	Dm_quality_tyID	

Dm_sample_ty

Database table in package 'DDL'

Dm_sample_ty
Version 1.0 Phase 1.0 Proposed
corongiu created on 29/02/2024. Last modified 17/03/2024
DBMS PostgreSQL

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
Circular	varchar	False	
Other	varchar	False	
Rectangular	varchar	False	
Transect	varchar	False	
Dm_sample_tyID	varchar	True	

PRIMARY KEY NAME	COLUMNS	COMMENTS
PK_Dm_sample_ty	Dm_sample_tyID	

Dm_species

Database table in package 'DDL'

Dm_species
Version 1.0 Phase 1.0 Proposed
corongiu created on 29/02/2024. Last modified 17/03/2024
DBMS PostgreSQL

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
Dm_speciesID	varchar	True	

PRIMARY KEY NAME	COLUMNS	COMMENTS
PK_Dm_species	Dm_speciesID	

Dm_status

Database table in package 'DDL'

Dm_status
Version 1.0 Phase 1.0 Proposed
corongiu created on 17/03/2024. Last modified 17/03/2024
DBMS PostgreSQL

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
Alive	varchar	False	
Dead	varchar	False	
Dm_statusID	varchar	True	

PRIMARY KEY NAME	COLUMNS	COMMENTS
PK_Dm_status	Dm_statusID	

Dm_ty_obj

Database table in package 'DDL'

Dm_ty_obj
Version 1.0 Phase 1.0 Proposed
corongiu created on 29/02/2024. Last modified 17/03/2024
DBMS PostgreSQL

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
Board	varchar	False	

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
Fallentree	varchar	False	
Log	varchar	False	
Standingtree	varchar	False	
Dm_ty_objID	varchar	True	

PRIMARY KEY NAME	COLUMNS	COMMENTS
PK_Dm_ty_obj	Dm_ty_objID	

Dt_areainfo

Database table in package 'DDL'

Dt_areainfo
Version 1.0 Phase 1.0 Proposed
corongiu created on 29/02/2024. Last modified 17/03/2024
DBMS PostgreSQL

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
Accessibility	varchar	False	Initial value: stand_forest accessibility code
Area	varchar	False	Initial value: stand_forest stand area
Bearing_capacity	varchar	False	Initial value: stand_forest average pressure between the harvesting machines and the soil
Chm	varchar	False	Initial value: stand_forest reference to TIFF file
Elevation	varchar	False	Initial value: stand_forest stand average/relevant elevation above see level
Productivity_area	varchar	False	Initial value: stand_forest Planted area (usually adjustment from maps or % of the total area)
Quality_classification	varchar	False	Initial value: stand_forest quality classification (to be defined)

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
Ruggedness	varchar	False	Initial value: stand_forest stand ruggedness description
Slope	varchar	False	Initial value: stand_forest stand average/relevant slope
Snow_cover_frequency	varchar	False	Initial value: stand_forest minimum height of snow that hampers harvesting activities
Soil_ty	varchar	False	Initial value: stand_forest soil type list?
Dt_areainfoID	varchar	True	

PRIMARY KEY NAME	COLUMNS	COMMENTS
PK_Dt_areainfo	Dt_areainfoID	

Dt_board

Database table in package 'DDL'

Dt_board
Version 1.0 Phase 1.0 Proposed
corongiu created on 29/02/2024. Last modified 17/03/2024
DBMS PostgreSQL

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
Knot?	varchar	False	
Length_uom	varchar	False	Alias: mm
Resin_pocket_number	varchar	False	
Stain_percentage	varchar	False	Alias: brrown/blue stain percentage
Sum_length_cracks	varchar	False	
Thickness_uom	varchar	False	Alias: mm

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
Width_average_uom	varchar	False	Alias: mm
Dt_boardID	varchar	True	

PRIMARY KEY NAME	COLUMNS	COMMENTS
PK_Dt_board	Dt_boardID	

Dt_calculatedinfo

Database table in package 'DDL'

Dt_calculatedinfo
Version 1.0 Phase 1.0 Proposed
corongiu created on 29/02/2024. Last modified 17/03/2024
DBMS PostgreSQL

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
Above_ground_biomass	varchar	False	
Age	varchar	False	Initial value: stand_forest/sample_plots/tree User provided or the Average age from lower levels -in years- (e.g. 85)
Basal_area	varchar	False	Initial value: stand_forest/sample_plots stand basal area
Below_ground_biomass	varchar	False	
Biomass	varchar	False	Initial value: stand_forest/sample-plots/tree total biomass
Mean_total_height	varchar	False	Initial value: stand_forest/sample_plots/tree Mean tree height in m (e.g. 25.8) for stand_forest or sample_plots, total_height for tree
Mean_tree_volume_ab	varchar	False	Initial value: stand_forest/sample_plots/tree Mean tree volume above bark in m3 (e.g. 0.82)
	varchar	False	Initial value: stand_forest/sample_plots/tree

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
Mean_tree_volume_bb			mean tree volume below the bark in m3 (e.g. 0.82)
Sample_area	varchar	False	Initial value: sample_plots Sample area can be calculated from plot radius/length or directly provided
Site_index	varchar	False	
Species	varchar	False	Initial value: stad_forest/sample_plots/tree List of species and Number of stems (e.g. AA, SS, LP)
Stem_count	varchar	False	Initial value: stand_forest/sample_plots number of stems within the stand
Top_height	varchar	False	
Yield_class	varchar	False	
Dt_calculatedinfoID	varchar	True	

PRIMARY KEY NAME	COLUMNS	COMMENTS
PK_Dt_calculatedinfo	Dt_calculatedinfoID	

Dt_cutting_instruction

Database table in package 'DDL'

A cutting instrouction contain multiple products weighting

Dt_cutting_instruction
Version 1.0 Phase 1.0 Proposed
corongiu created on 29/02/2024. Last modified 17/03/2024
DBMS PostgreSQL

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
Product_type_id	varchar	False	Initial value: cutting_instruction "product_type" object
Weighting	varchar	False	Initial value: cutting_instruction product type weting values

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
Dt_cutting_instructionID	varchar	True	

PRIMARY KEY NAME	COLUMNS	COMMENTS
PK_Dt_cutting_instruction	Dt_cutting_instructionID	

Dt_device_survey_info

Database table in package 'DDL'

Dt_device_survey_info
Version 1.0 Phase 1.0 Proposed
corongiu created on 17/03/2024. Last modified 17/03/2024
DBMS PostgreSQL

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
...	varchar	False	
Dt_device_survey_infoID	varchar	True	

PRIMARY KEY NAME	COLUMNS	COMMENTS
PK_Dt_device_survey_info	Dt_device_survey_infoID	

Dt_heartwood

Database table in package 'DDL'

Dt_heartwood
Version 1.0 Phase 1.0 Proposed
corongiu created on 29/02/2024. Last modified 17/03/2024
DBMS PostgreSQL

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
Green_density_average_uom	varchar	False	Alias: g/dm3
Pith_deviation_max_uom	varchar	False	Alias: mm
Pith_eccentricity_max_uom	varchar	False	Alias: mm

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
Sapwood_green_density_uom	varchar	False	Alias: g/dm3
Volume_percentage	varchar	False	
Dt_heartwoodID	varchar	True	

PRIMARY KEY NAME	COLUMNS	COMMENTS
PK_Dt_heartwood	Dt_heartwoodID	

Dt_inventory

Database table in package 'DDL'

Dt_inventory
Version 1.0 Phase 1.0 Proposed
corongiu created on 29/02/2024. Last modified 17/03/2024
DBMS PostgreSQL

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
...	varchar	False	
Dt_inventoryID	varchar	True	

PRIMARY KEY NAME	COLUMNS	COMMENTS
PK_Dt_inventory	Dt_inventoryID	

Dt_knot

Database table in package 'DDL'

Dt_knot
Version 1.0 Phase 1.0 Proposed
corongiu created on 29/02/2024. Last modified 17/03/2024
DBMS PostgreSQL

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
Diameter_average_uom	varchar	False	
Diameter_max_uom	varchar	False	

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
Knot_ty	varchar	False	
Lenght_average_uom	varchar	False	
Number	varchar	False	
Volume_uom	varchar	False	
Dt_knotID	varchar	True	

PRIMARY KEY NAME	COLUMNS	COMMENTS
PK_Dt_knot	Dt_knotID	

Dt_measures

Database table in package 'DDL'

Dt_measures
Version 1.0 Phase 1.0 Proposed
corongiu created on 29/02/2024. Last modified 17/03/2024
DBMS PostgreSQL

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
Diameter_uom	varchar	False	Initial value: stand/log/saw/product_type [value] [uom] [min/max] [LED/SED]
Height_uom	varchar	False	Initial value: log [value] [uom] [top/bottom]
Length_uom	varchar	False	Initial value: stand/log/saw/product_type [value] [uom] [min/max]
Sweep	varchar	False	Initial value: product_type Maximum sweep in the log product
Position_uom	varchar	False	Initial value: stand log
Quality_allowed	varchar	False	Initial value: product_type Type of defect allowed for his type of product (1,2,3)
Quality_indicator	varchar	False	Initial value: product_type

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
Species	varchar	False	Initial value: product_type
Volume_uom	varchar	False	Initial value: stand log saw [value] [uom] [ab/bb]
Dt_measuresID	varchar	True	

PRIMARY KEY NAME	COLUMNS	COMMENTS
PK_Dt_measures	Dt_measuresID	

Dt_namespecis

Database table in package 'DDL'

Dt_namespecis
Version 1.0 Phase 1.0 Proposed
corongiu created on 29/02/2024. Last modified 17/03/2024
DBMS PostgreSQL

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
Code	varchar	False	Initial value: speciesTable
Englishname	varchar	False	Initial value: speciesTable
Group	varchar	False	Initial value: speciesTable
Latinname	varchar	False	Initial value: speciesTable
Othername	varchar	False	Initial value: speciesTable
Dt_namespecisID	varchar	True	

PRIMARY KEY NAME	COLUMNS	COMMENTS
PK_Dt_namespecis	Dt_namespecisID	

Dt_property

Database table in package 'DDL'

Dt_property
Version 1.0 Phase 1.0 Proposed
corongiu created on 29/02/2024. Last modified 17/03/2024
DBMS PostgreSQL

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
Address	varchar	False	Initial value: treeProperty address of the forest
Country	varchar	False	Initial value: treeProperty country where the forest is located
Name	varchar	False	Initial value: treeProperty
Owner_name	varchar	False	Initial value: treeProperty Name of the owner
Ownership	varchar	False	Initial value: treeProperty Type of ownership
Timestamp	varchar	False	Initial value: treeProperty time and date when the property is created
Uuid_property	varchar	False	Initial value: treeProperty
Zip_code	varchar	False	Initial value: treeProperty zip code of the forest
Dt_propertyID	varchar	True	

PRIMARY KEY NAME	COLUMNS	COMMENTS
PK_Dt_property	Dt_propertyID	

Dt_quality

Database table in package 'DDL'

Dt_quality
Version 1.0 Phase 1.0 Proposed
corongiu created on 29/02/2024. Last modified 17/03/2024
DBMS PostgreSQL

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
Quality_indicator	varchar	False	Initial value: product_type
Quality_ty	varchar	False	Initial value: tree/log/product_type Allow for multiple quality parameter per log
Quality_value	varchar	False	Initial value: tree/log/product_type **[value] [min/max]**
Status	varchar	False	
Dt_qualityID	varchar	True	

PRIMARY KEY NAME	COLUMNS	COMMENTS
PK_Dt_quality	Dt_qualityID	

Dt_sawmill_info

Database table in package 'DDL'

Dt_sawmill_info
Version 1.0 Phase 1.0 Proposed
corongiu created on 17/03/2024. Last modified 17/03/2024
DBMS PostgreSQL

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
...	varchar	False	
Dt_sawmill_infoID	varchar	True	

PRIMARY KEY NAME	COLUMNS	COMMENTS
PK_Dt_sawmill_info	Dt_sawmill_infoID	

Dt_species

Database table in package 'DDL'

This table refers to calculated info even when used out of Dt_calculatedinfo.

Dt_species
Version 1.0 Phase 1.0 Proposed
corongiu created on 29/02/2024. Last modified 17/03/2024
DBMS PostgreSQL

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
	varchar	False	Alias: m2/ha

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
Basal_area_uom			Initial value: stand_species_summaryTable
Bm_expansion_factor	varchar	False	Initial value: species_parameters Biomass expansion factor (used for biomass calculation)
Density	varchar	False	Initial value: species_parameters Wood density used for quality and biomass calculations
Height_mean_uom	varchar	False	Alias: m Initial value: stand_species_summaryTable
Root_shoot	varchar	False	Initial value: species_parameters Root to shoot ratio (used for biomass calculation)
Specie_name	varchar	False	Initial value: stand_species_summaryTable
Stem_count	varchar	False	Initial value: stand_species_summaryTable
Volume_equation	varchar	False	Initial value: species_parameters Taper equation
Volume_uom	varchar	False	Alias: m3 Initial value: stand_species_summaryTable [value] [uom] [ab/bb/mean]
Dt_speciesID	varchar	True	

PRIMARY KEY NAME	COLUMNS	COMMENTS
PK_Dt_species	Dt_speciesID	

Dt_volume

Database table in package 'DDL'

Dt_volume
Version 1.0 Phase 1.0 Proposed
corongiu created on 29/02/2024. Last modified 17/03/2024
DBMS PostgreSQL

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
Type	varchar	False	

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
Value	varchar	False	
Dt_volumeID	varchar	True	

PRIMARY KEY NAME	COLUMNS	COMMENTS
PK_Dt_volume	Dt_volumeID	

Identification

Database table in package 'DDL'

Identification
Version 1.0 Phase 1.0 Proposed
corongiu created on 29/02/2024. Last modified 17/03/2024
DBMS PostgreSQL

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
Cat_image	varchar	False	Initial value: voidable
Device	varchar	False	
Device_survey_info	varchar	False	
Externaiid	varchar	False	Alias: RFID, XRay, Punching Initial value: voidable
Externalid_md	varchar	False	
Father	varchar	False	
Gm_position	varchar	False	
Internalid	varchar	False	Alias: GUID
Tagposition	varchar	False	
Timestamp	varchar	False	on standing tree consider only the lastOne related to the RFID positioning
Type_obj	varchar	False	

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
Uuid_identification	varchar	False	
Uuid_inventory	varchar	False	
IdentificationID	varchar	True	
UUID_inventory	varchar	False	
UUID_identidfication	varchar	False	

PRIMARY KEY NAME	COLUMNS	COMMENTS
PK_Identification	IdentificationID	

FOREIGN KEY NAME	COLUMNS	REFERENCES
FK_identification_UUID_inventory	UUID_inventory	Inventorydata(InventorydataID)
FK_identification_father	UUID_identidfication	Identification(IdentificationID)

Inventorydata

Database table in package 'DDL'

Inventorydata
Version 1.0 Phase 1.0 Proposed
corongiu created on 29/02/2024. Last modified 17/03/2024
DBMS PostgreSQL

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
Areainfo	varchar	False	Initial value: forest_stand
Cadastre	varchar	False	
Calculatedinfo	varchar	False	Initial value: forest_stand
Gm_boundary	varchar	False	Initial value: forest_stand stand boundary
Gm_location	varchar	False	Initial value: forest_stand/tree stand location, POINT object

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
Internalid	varchar	False	Initial value: forest_stand/tree
Inventorydata_ty	varchar	False	
Inventoryfather	varchar	False	
Inventoryinfo	varchar	False	Initial value: voidable
Parcelinfo	varchar	False	
Property	varchar	False	Initial value: forest_stand
Quality	varchar	False	Initial value: tree Allow for multiple quality parameter per tree
Timestamp	varchar	False	Initial value: forest_stand/tree timestamp when the stan_forest or tree is created
Tree_anchor	varchar	False	Initial value: tree
Tree_intermediate_support	varchar	False	Initial value: tree
Uuid_inventory	varchar	False	
InventorydataID	varchar	True	
inventoryFather	varchar	False	

PRIMARY KEY NAME	COLUMNS	COMMENTS
PK_Inventorydata	InventorydataID	

FOREIGN KEY NAME	COLUMNS	REFERENCES
FK_inventoryData_UUID_inventory	inventoryFather	Inventorydata(InventorydataID)

Sample_plots

Database table in package 'DDL'

Sample_plots
Version 1.0 Phase 1.0 Proposed
corongiu created on 29/02/2024. Last modified 17/03/2024
DBMS PostgreSQL

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
Calculatedinfo	varchar	False	Initial value: sample_plots
Externalid	varchar	False	Initial value: sample_plots
Gm_location	varchar	False	Initial value: sample_plots
Length1	varchar	False	Initial value: sample_plots Radius or length of the plot
Length2	varchar	False	Initial value: sample_plots Second side length for rectangular plot
Sample_ty	varchar	False	Initial value: sample_plots
Uuid_inventory	varchar	False	
Uuid_sampleplots	varchar	False	
Sample_plotsID	varchar	True	
UUID_inventoryData	varchar	False	

PRIMARY KEY NAME	COLUMNS	COMMENTS
PK_Sample_plots	Sample_plotsID	

FOREIGN KEY NAME	COLUMNS	REFERENCES
FK_sample_plots_UUID_inventory	UUID_inventoryData	Inventorydata(InventorydataID)

Sawmilldata

Database table in package 'DDL'

Sawmilldata
Version 1.0 Phase 1.0 Proposed
corongiu created on 29/02/2024. Last modified 17/03/2024
DBMS PostgreSQL

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
Heartwood	varchar	False	
Knot	varchar	False	
Sawmill_info	varchar	False	
Uuid_identification	varchar	False	
Uuid_sawmill	varchar	False	
SawmilldataID	varchar	True	
UUID_identification	varchar	False	

PRIMARY KEY NAME	COLUMNS	COMMENTS
PK_Sawmilldata	SawmilldataID	

FOREIGN KEY NAME	COLUMNS	REFERENCES
FK_sawmillData_UUID_identification	UUID_identification	Identification(IdentificationID)

Tree_taper

Database table in package 'DDL'

(DBH-Height description)

Tree_taper
Version 1.0 Phase 1.0 Proposed
corongiu created on 29/02/2024. Last modified 17/03/2024
DBMS PostgreSQL

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
Bark_thickness	varchar	False	Initial value: tree_taper Bark thickness for given diameter Needed to estimate without bark values calculated or provided

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
Diameter_ab	varchar	False	Initial value: tree_taper Diameter above bark
Diameter_bb	varchar	False	Initial value: tree_taper Diameter below bark
Height	varchar	False	Initial value: tree_taper
Quality_taper	varchar	False	Initial value: tree_taper
Uuid_inventory_wood	varchar	False	Initial value: tree_taper from inventoyData or woodData featureType
Uuid_taper	varchar	False	Initial value: tree_taper
Tree_taperID	varchar	True	
UUID_inventory_tree	varchar	False	
UUID_inventory_wood	varchar(50)	False	

PRIMARY KEY NAME	COLUMNS	COMMENTS
PK_Tree_taper	Tree_taperID	

FOREIGN KEY NAME	COLUMNS	REFERENCES
FK_tree_taper_UUID_inventory	UUID_inventory_tree	Inventorydata(InventorydataID)
FK_tree_taper_UUID_wood	UUID_inventory_wood	Wooddata(WooddataID)

Wooddata

Database table in package 'DDL'

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
Board	varchar	False	
Comment	varchar	False	Initial value: voidable
Cutting_instruction	varchar	False	Initial value: product_type_weightings cutting instruction name
Dbh_uom	varchar	False	Initial value: voidable
Extimated	varchar	False	Initial value: product_type
Knot_position_uom	varchar	False	
Measured	varchar	False	
Quality	varchar	False	Initial value: log
Species	varchar	False	
Uuid_identification	varchar	False	
Uuid_wood	varchar	False	
WooddataID	varchar	True	
UUID_identification	varchar	False	

PRIMARY KEY NAME	COLUMNS	COMMENTS
PK_Wooddata	WooddataID	

FOREIGN KEY NAME	COLUMNS	REFERENCES
FK_woodData_UUID_identification	UUID_identification	Identification(IdentificationID)

Use Case Model

Package in package 'Sintetic_DB'

The Use Case model is a catalogue of system functionality described using UML Use Cases. Each Use Case represents a single, repeatable interaction that a user or "actor" experiences when using the system.

A Use Case typically includes one or more "scenarios" which describe the interactions that go on between the Actor and the System and documents the results and exceptions that occur from the user's perspective.

Use Cases may include other Use Cases as part of a larger pattern of interaction and may also be extended by other use cases to handle exceptional conditions.

Use Case Model
Version Phase 1.0 Proposed
corongiu created on 14/03/2024. Last modified 14/03/2024

Use Case Model diagram

Use Case diagram in package 'Use Case Model'

According to UML, the Use Case Model is made of two packages: the first one includes Actors involved in use cases, the second one includes the primary use cases the project is going to consider according to WP2 Prototypes, WP4 – Demonstrations, and WP5 - Data Analysis and Modelling.

Use Case Model
Version 1.0
corongiu created on 14/03/2024. Last modified 14/03/2024

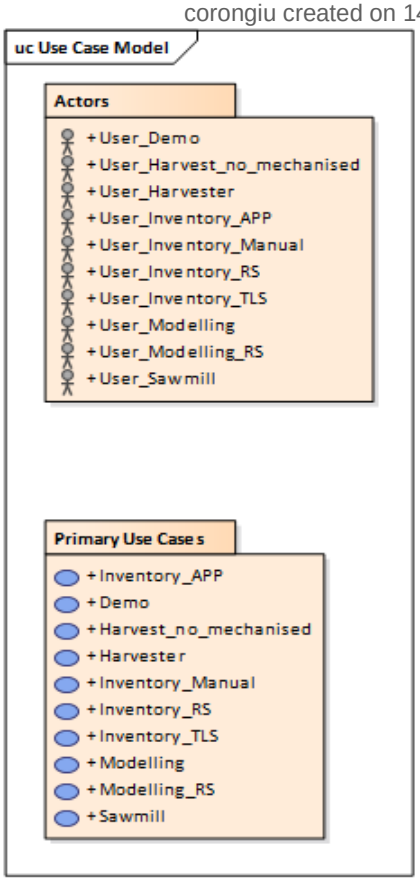


Figure 6: USE Case Model – Actors and Primary Use Cases

Actors

Package in package 'Use Case Model'

Actors are the users of the system being modeled. Each Actor will have a well-defined role, and in the context of that role have useful interactions with the system.

A person may perform the role of more than one Actor, although they will only assume one role during one use case interaction.

An Actor role may be performed by a non-human system, such as another computer program.

Actors
Version 1.0 Phase 1.0 Proposed
corongiu created on 14/03/2024. Last modified 14/03/2024

Actors diagram

Use Case diagram in package 'Actors'

Actors
Version 1.0
corongiu created on 14/03/2024. Last modified 15/03/2024

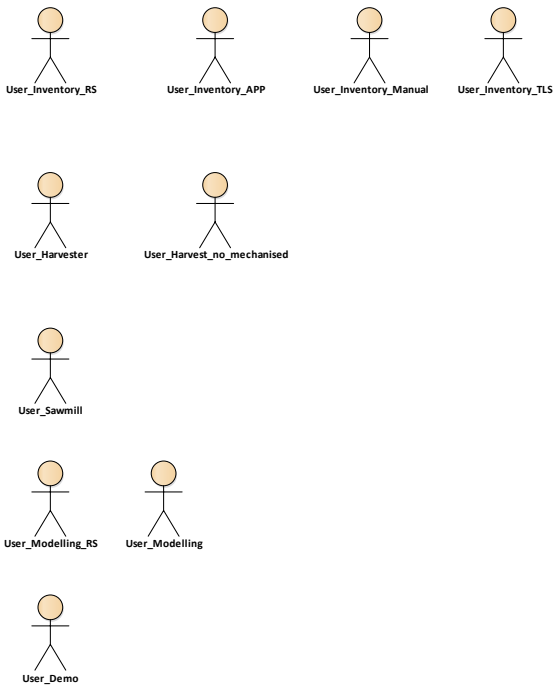


Figure 7: USE Case Model – Actors

Primary Use Cases

Package in package 'Use Case Model'

This package contains use cases which define how an Actor will interact with the proposed system.

Each interaction may be specified using scenarios, sequence diagrams, communication diagrams and other dynamic diagrams or textual descriptions which together describe how the system, when viewed as a "black-box", interacts with a user.

Primary Use Cases
Version 1.0 Phase 1.0 Proposed
corongiu created on 14/03/2024. Last modified 15/03/2024

Primary Use Cases diagram

Use Case diagram in package 'Primary Use Cases'

Primary Use Cases
Version 1.0
corongiu created on 14/03/2024. Last modified 15/03/2024

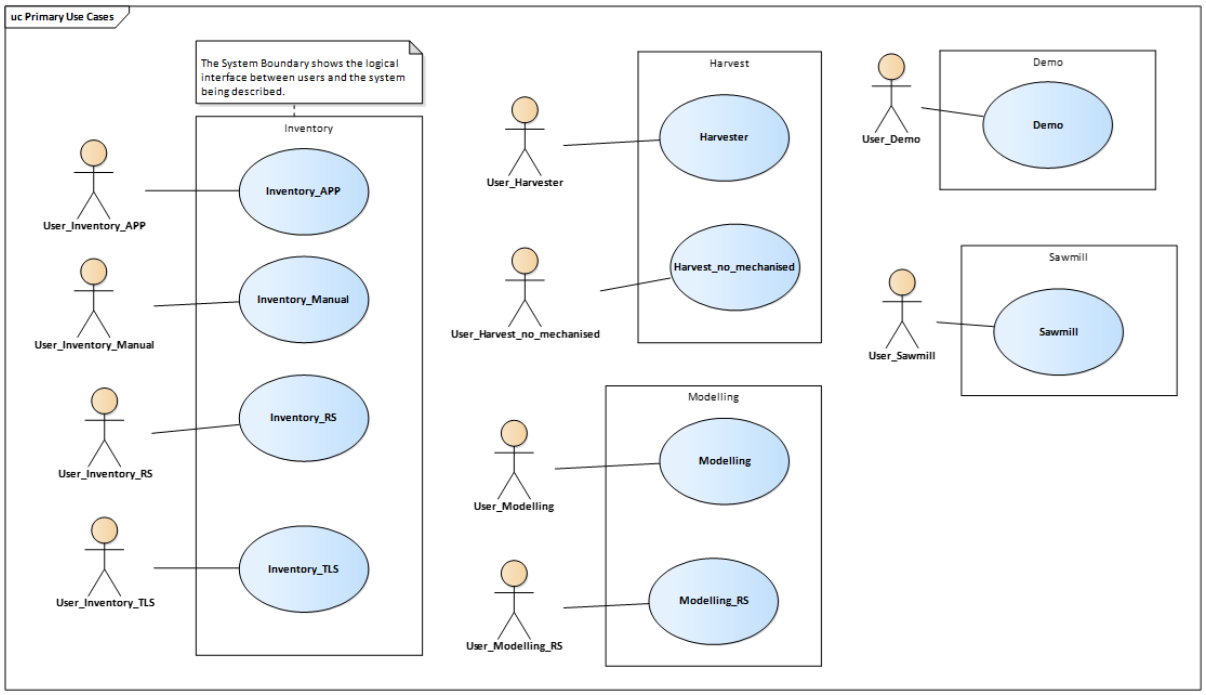


Figure 8: USE Case Model – Primary Use Cases

User_Demo

Actor in package 'Actors'

User_Demo
Version 1.0 Phase 1.0 Proposed
corongiu created on 14/03/2024. Last modified 15/03/2024

ASSOCIATIONS

Association (direction: Unspecified)

Source: Public (Actor) User_Demo

Target: Public (UseCase) Demo

User_Harvest_no_mechanised
Actor in package 'Actors'

User_Harvest_no_mechanised
Version 1.0 Phase 1.0 Proposed
corongiu created on 14/03/2024. Last modified 15/03/2024

ASSOCIATIONS

Association (direction: Unspecified)

Source: Public (Actor) User_Harvest_no_mechanised

Target: Public (UseCase) Harvest_no_mechanised

User_Harvester
Actor in package 'Actors'

User_Harvester
Version 1.0 Phase 1.0 Proposed
corongiu created on 14/03/2024. Last modified 15/03/2024

ASSOCIATIONS

Association (direction: Unspecified)

Source: Public (Actor) User_Harvester

Target: Public (UseCase) Harvester

User_Inventory_APP
Actor in package 'Actors'

User_Inventory_APP
Version 1.0 Phase 1.0 Proposed
corongiu created on 14/03/2024. Last modified 14/03/2024

ASSOCIATIONS

Association (direction: Unspecified)

Source: Public (Actor) User_Inventory_APP

Target: Public (UseCase) Inventory_APP

User_Inventory_Manual
Actor in package 'Actors'

User_Inventory_Manual
Version 1.0 Phase 1.0 Proposed
corongiu created on 14/03/2024. Last modified 14/03/2024

ASSOCIATIONS

Association (direction: Unspecified)

Source: Public (Actor) User_Inventory_Manual — Target: Public (UseCase) Inventory_Manual

User_Inventory_RS
Actor in package 'Actors'

User_Inventory_RS
Version 1.0 Phase 1.0 Proposed
corongiu created on 14/03/2024. Last modified 15/03/2024

ASSOCIATIONS

Association (direction: Unspecified)

Source: Public (Actor) User_Inventory_RS — Target: Public (UseCase) Inventory_RS

User_Inventory_TLS
Actor in package 'Actors'

User_Inventory_TLS
Version 1.0 Phase 1.0 Proposed
corongiu created on 14/03/2024. Last modified 14/03/2024

ASSOCIATIONS

Association (direction: Unspecified)

Source: Public (Actor) User_Inventory_TLS — Target: Public (UseCase) Inventory_TLS

User_Modelling
Actor in package 'Actors'

User_Modelling
Version 1.0 Phase 1.0 Proposed
corongiu created on 15/03/2024. Last modified 15/03/2024

ASSOCIATIONS

Association (direction: Unspecified)

Source: Public (Actor) User_Modelling — Target: Public (UseCase) Modelling

User_Modelling_RS
Actor in package 'Actors'

User_Modelling_RS
Version 1.0 Phase 1.0 Proposed
corongiu created on 14/03/2024. Last modified 15/03/2024

ASSOCIATIONS

Association (direction: Unspecified)

Source: Public (Actor) User_Modelling_RS — Target: Public (UseCase) Modelling_RS

User_Sawmill
Actor in package 'Actors'

User_Sawmill
Version 1.0 Phase 1.0 Proposed
corongiu created on 14/03/2024. Last modified 15/03/2024

ASSOCIATIONS

Association (direction: Unspecified)

Source: Public (Actor) User_Sawmill — Target: Public (UseCase) Sawmill

Demo
Boundary in package 'Primary Use Cases'

Demo
Version 1.0 Phase 1.0 Proposed
corongiu created on 15/03/2024. Last modified 15/03/2024
Extends

Harvest
Boundary in package 'Primary Use Cases'

Harvest
Version 1.0 Phase 1.0 Proposed
corongiu created on 15/03/2024. Last modified 15/03/2024
Extends

Inventory
Boundary in package 'Primary Use Cases'

Inventory
Version 1.0 Phase 1.0 Proposed
corongiu created on 14/03/2024. Last modified 14/03/2024
Extends

Modelling
Boundary in package 'Primary Use Cases'

Modelling
Version 1.0 Phase 1.0 Proposed
corongiu created on 15/03/2024. Last modified 15/03/2024
Extends

Sawmill

Boundary in package 'Primary Use Cases'

Sawmill
Version 1.0 Phase 1.0 Proposed
corongiu created on 15/03/2024. Last modified 15/03/2024
Extends

Demo

UseCase in package 'Primary Use Cases'

WP5

Demo
Version 1.0 Phase 1.0 Proposed
corongiu created on 15/03/2024. Last modified 15/03/2024

ELEMENTS OWNED BY Demo
Sequence : Sequence
Demo : Event
End_Demo : MessageEnd
End_Upload_GeoDB : MessageEnd
GeoDB : Sequence «entity»
GUI : Sequence «entity»

ASSOCIATIONS	
Association (direction: Unspecified)	
Source: Public (Actor) User_Demo	Target: Public (UseCase) Demo

Demo Use Case diagram

Interaction diagram in package 'Primary Use Cases'

Demo Use Case
Version 1.0
corongiu created on 15/03/2024. Last modified 15/03/2024

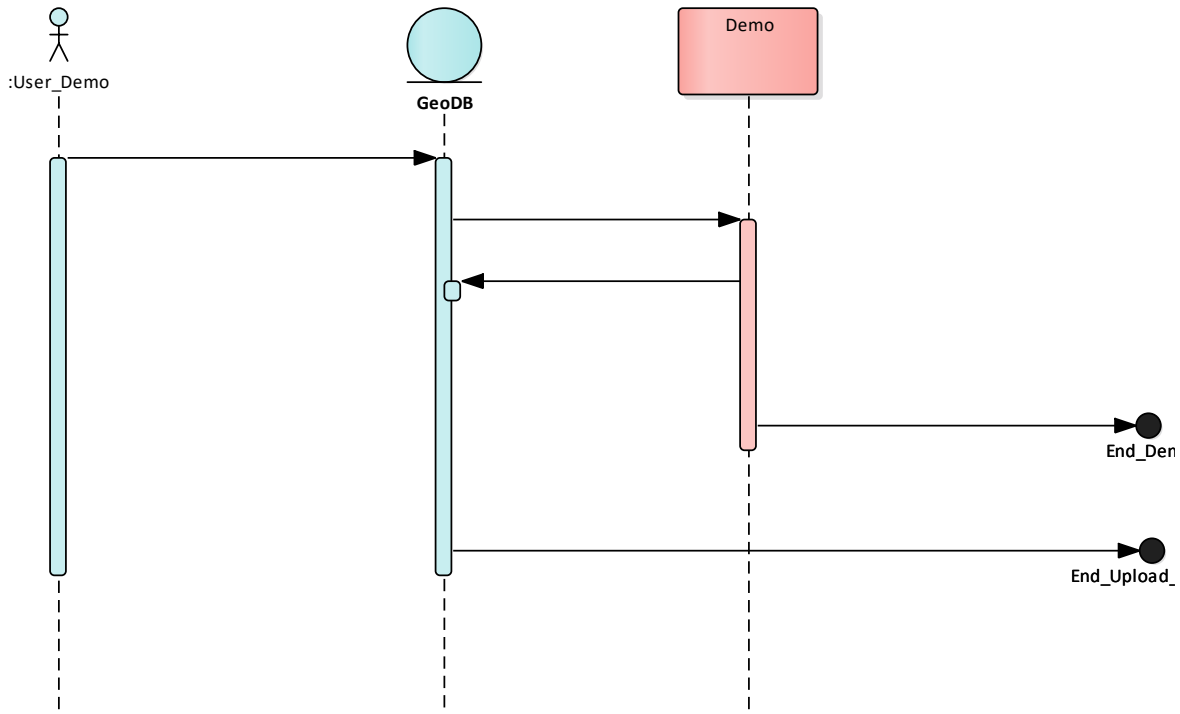


Figure 9: USE Case Model – Demo use case Diagram

INTERACTION MESSAGES
1.0 '' from ':User_Demo' sent to 'GeoDB'. Synchronous Call. Returns void. [Return is False. Iteration is False. New group is False.]
1.1 '' from 'GeoDB' sent to 'Demo'. Synchronous Call. Returns void. [Return is False. Iteration is False. New group is False.]
1.2 '' from 'Demo' sent to 'GeoDB'. Synchronous Call. Returns void. [Return is False. Iteration is False. New group is False.]

INTERACTION MESSAGES

1.3 '' from 'Demo' sent to 'End_Demo'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.4 '' from 'GeoDB' sent to 'End_Upload_GeoDB'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

Sequence

Sequence owned by 'Demo', in package 'Primary Use Cases'

Sequence
Version 1.0 Phase 1.0 Proposed
corongiu created on 15/03/2024. Last modified 15/03/2024

OUTGOING BEHAVIORAL RELATIONSHIPS

Sequence from Sequence to «entity» GeoDB

Demo

Event owned by 'Demo', in package 'Primary Use Cases'

Demo
Version 1.0 Phase 1.0 Proposed
corongiu created on 15/03/2024. Last modified 15/03/2024

End_Demo

MessageEnd owned by 'Demo', in package 'Primary Use Cases'

End_Demo
Version 1.0 Phase 1.0 Proposed
corongiu created on 15/03/2024. Last modified 15/03/2024

INCOMING BEHAVIORAL RELATIONSHIPS

Sequence from Demo to End_Demo

End_Upload_GeoDB

MessageEnd owned by 'Demo', in package 'Primary Use Cases'

End_Upload_GeoDB
Version 1.0 Phase 1.0 Proposed
corongiu created on 15/03/2024. Last modified 15/03/2024

INCOMING BEHAVIORAL RELATIONSHIPS
Sequence from «entity» GeoDB to End_Upload_GeoDB

GeoDB

Sequence «entity» owned by 'Demo', in package 'Primary Use Cases'

GeoDB
Version 1.0 Phase 1.0 Proposed
corongiu created on 15/03/2024. Last modified 15/03/2024

OUTGOING BEHAVIORAL RELATIONSHIPS
Sequence from «entity» GeoDB to End_Upload_GeoDB
Sequence from «entity» GeoDB to Demo

INCOMING BEHAVIORAL RELATIONSHIPS
Sequence from Sequence to «entity» GeoDB
Sequence from Demo to «entity» GeoDB

GUI

Sequence «entity» owned by 'Demo', in package 'Primary Use Cases'

GUI
Version 1.0 Phase 1.0 Proposed
corongiu created on 15/03/2024. Last modified 15/03/2024

Harvest_no_mechanised

UseCase in package 'Primary Use Cases'

Harvest_no_mechanised
Version 1.0 Phase 1.0 Proposed
corongiu created on 15/03/2024. Last modified 15/03/2024

ELEMENTS OWNED BY Harvest_no_mechanised
Sequence : Sequence
DataCollection : Event
End_Harvest_no_mechanised : MessageEnd
GeoDB : Sequence «entity»

ELEMENTS OWNED BY Harvest_no_mechanised

GUI : Sequence «entity»

ASSOCIATIONS

Association (direction: Unspecified)

Source: Public (Actor) User_Harvest_no_mechanised

Target: Public (UseCase) Harvest_no_mechanised

Harvest_no_mechanised Use Case diagram

Interaction diagram in package 'Primary Use Cases'

Harvest_no_mechanised Use Case
Version 1.0
corongiu created on 15/03/2024. Last modified 15/03/2024

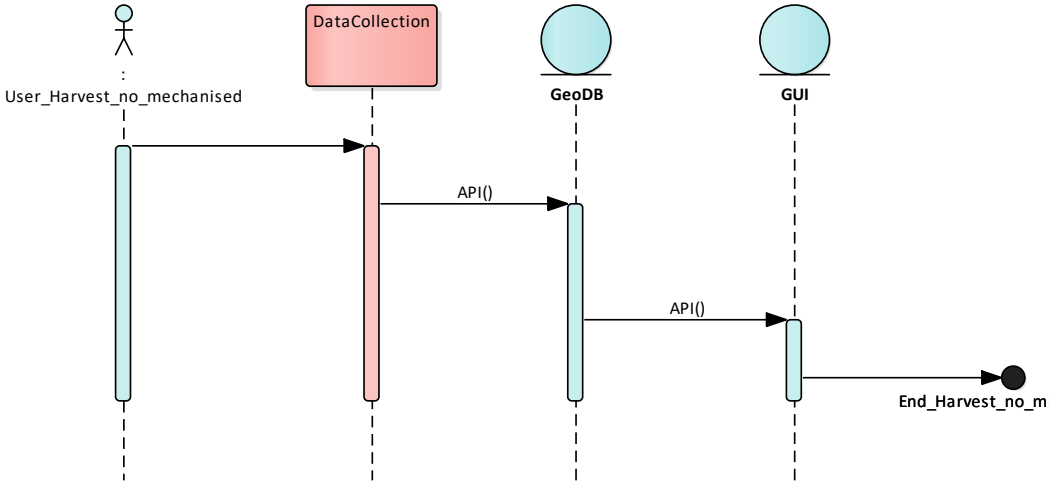


Figure 10: USE Case Model – HArvest_no_mechanised use case Diagram

INTERACTION MESSAGES

1.0 '' from ':User_Harvest_no_mechanised' sent to 'DataCollection'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.1 'API' from 'DataCollection' sent to 'GeoDB'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

INTERACTION MESSAGES

1.2 'API' from 'GeoDB' sent to 'GUI'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.3 '' from 'GUI' sent to 'End_Harvest_no_mechanised'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

Sequence

Sequence owned by 'Harvest_no_mechanised', in package 'Primary Use Cases'

Sequence
Version 1.0 Phase 1.0 Proposed
corongiu created on 15/03/2024. Last modified 15/03/2024

OUTGOING BEHAVIORAL RELATIONSHIPS

Sequence from Sequence to DataCollection

DataCollection

Event owned by 'Harvest_no_mechanised', in package 'Primary Use Cases'

DataCollection
Version 1.0 Phase 1.0 Proposed
corongiu created on 15/03/2024. Last modified 15/03/2024

End_Harvest_no_mechanised

MessageEnd owned by 'Harvest_no_mechanised', in package 'Primary Use Cases'

End_Harvest_no_mechanised
Version 1.0 Phase 1.0 Proposed
corongiu created on 15/03/2024. Last modified 15/03/2024

INCOMING BEHAVIORAL RELATIONSHIPS

Sequence from «entity» GUI to End_Harvest_no_mechanised

GeoDB

Sequence «entity» owned by 'Harvest_no_mechanised', in package 'Primary Use Cases'

GeoDB
Version 1.0 Phase 1.0 Proposed
corongiu created on 15/03/2024. Last modified 15/03/2024

OUTGOING BEHAVIORAL RELATIONSHIPS
Name: API Sequence from «entity» GeoDB to «entity» GUI

INCOMING BEHAVIORAL RELATIONSHIPS
Name: API Sequence from DataCollection to «entity» GeoDB

GUI

Sequence «entity» owned by 'Harvest_no_mechanised', in package 'Primary Use Cases'

GUI
Version 1.0 Phase 1.0 Proposed
corongiu created on 15/03/2024. Last modified 15/03/2024

OUTGOING BEHAVIORAL RELATIONSHIPS
Sequence from «entity» GUI to End_Harvest_no_mechanised

INCOMING BEHAVIORAL RELATIONSHIPS
Name: API Sequence from «entity» GeoDB to «entity» GUI

Harvester

UseCase in package 'Primary Use Cases'

Harvester
Version 1.0 Phase 1.0 Proposed
corongiu created on 15/03/2024. Last modified 15/03/2024

ELEMENTS OWNED BY Harvester
CentralComputer : Object «entity» processor head
Sequence : Sequence
End_Harvester : MessageEnd
GeoDB : Sequence «entity»

ELEMENTS OWNED BY Harvester
GUI : Sequence «entity»

ASSOCIATIONS	
Association (direction: Unspecified)	
Source: Public (Actor) User_Harvester	Target: Public (UseCase) Harvester

Harvester Use Case diagram

Interaction diagram in package 'Primary Use Cases'

Harvester Use Case
Version 1.0
corongiu created on 15/03/2024. Last modified 15/03/2024

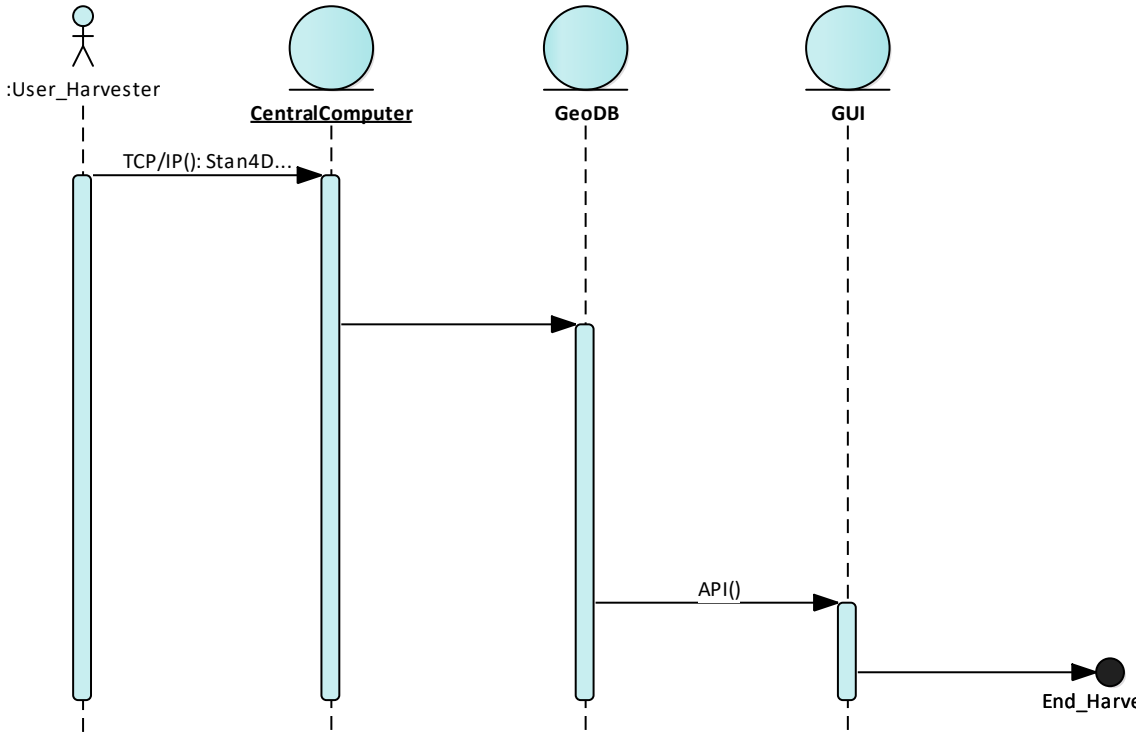


Figure 11: USE Case Model – Harvester use case Diagram

INTERACTION MESSAGES

1.0 'TCP/IP' from ':User_Harvester' sent to 'CentralComputer'.

Synchronous Call. Returns Stan4D....

[Return is False. Iteration is False. New group is False.]

1.1 '' from 'CentralComputer' sent to 'GeoDB'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.2 'API' from 'GeoDB' sent to 'GUI'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.3 '' from 'GUI' sent to 'End_Harvester'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

CentralComputer

Object «entity» owned by 'Harvester', in package 'Primary Use Cases'

processor head

CentralComputer
Version 1.0 Phase 1.0 Proposed
corongiu created on 15/03/2024. Last modified 15/03/2024

Sequence

Sequence owned by 'Harvester', in package 'Primary Use Cases'

Sequence
Version 1.0 Phase 1.0 Proposed
corongiu created on 15/03/2024. Last modified 15/03/2024

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: TCP/IP
Sequence from Sequence to «entity» CentralComputer

End_Harvester

MessageEnd owned by 'Harvester', in package 'Primary Use Cases'

End_Harvester
Version 1.0 Phase 1.0 Proposed
corongiu created on 15/03/2024. Last modified 15/03/2024

INCOMING BEHAVIORAL RELATIONSHIPS

Sequence from «entity» GUI to End_Harvester

GeoDB

Sequence «entity» owned by 'Harvester', in package 'Primary Use Cases'

GeoDB
Version 1.0 Phase 1.0 Proposed
corongiu created on 15/03/2024. Last modified 15/03/2024

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: API
Sequence from «entity» GeoDB to «entity» GUI

INCOMING BEHAVIORAL RELATIONSHIPS

Sequence from «entity» CentralComputer to «entity» GeoDB

GUI

Sequence «entity» owned by 'Harvester', in package 'Primary Use Cases'

GUI
Version 1.0 Phase 1.0 Proposed
corongiu created on 15/03/2024. Last modified 15/03/2024

OUTGOING BEHAVIORAL RELATIONSHIPS

Sequence from «entity» GUI to End_Harvester

INCOMING BEHAVIORAL RELATIONSHIPS

Name: API
Sequence from «entity» GeoDB to «entity» GUI

Inventory_APP

UseCase in package 'Primary Use Cases'

Inventory_APP
Version 1.0 Phase 1.0 Proposed
corongiu created on 14/03/2024. Last modified 14/03/2024

ELEMENTS OWNED BY Inventory_APP
DataCollection : Event
End_Inventory_APP : MessageEnd
Sequence : Sequence
GeoDB : Sequence «entity»
GUI : Sequence «entity»

ASSOCIATIONS	
Association (direction: Unspecified)	
Source: Public (Actor) User_Inventory_APP	Target: Public (UseCase) Inventory_APP

Inventori_APP Use Case diagram

Interaction diagram in package 'Primary Use Cases'

Inventori_APP Use Case
Version 1.0
corongiu created on 14/03/2024. Last modified 15/03/2024

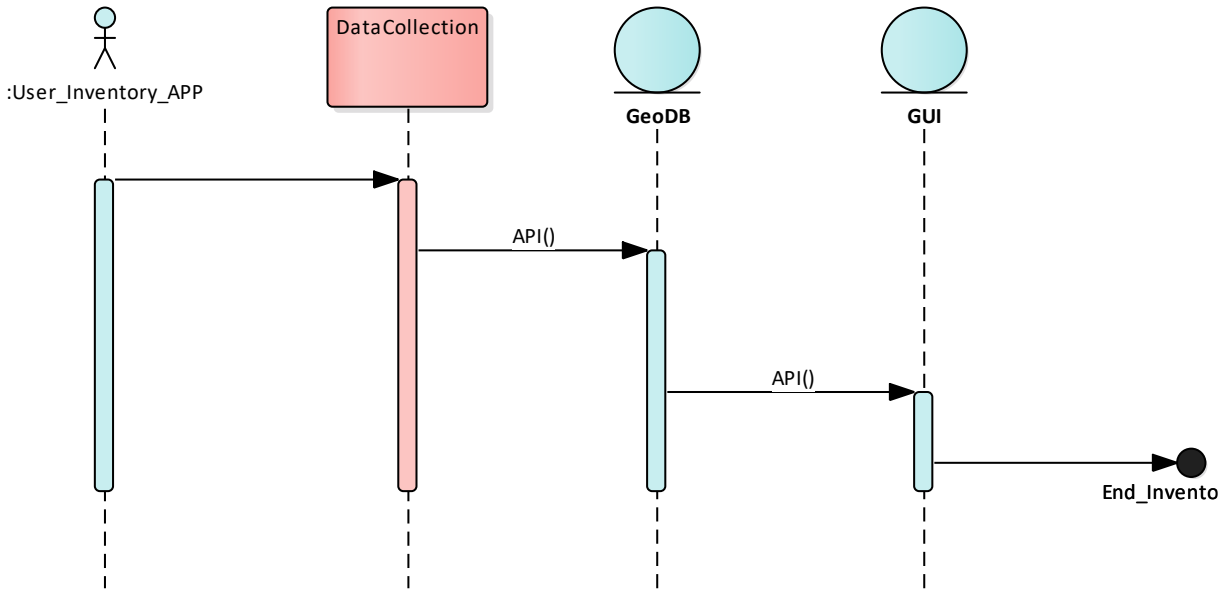


Figure 12: USE Case Model – Inventory use case Diagram

INTERACTION MESSAGES

1.0 '' from ':User_Inventory_APP' sent to 'DataCollection'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.1 'API' from 'DataCollection' sent to 'GeoDB'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.2 'API' from 'GeoDB' sent to 'GUI'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.3 '' from 'GUI' sent to 'End_Inventory_APP'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

DataCollection

Event owned by 'Inventory_APP', in package 'Primary Use Cases'

DataCollection
Version 1.0 Phase 1.0 Proposed
corongiu created on 14/03/2024. Last modified 14/03/2024

End_Inventory_APP

MessageEnd owned by 'Inventory_APP', in package 'Primary Use Cases'

End_Inventory_APP
Version 1.0 Phase 1.0 Proposed
corongiu created on 14/03/2024. Last modified 15/03/2024

INCOMING BEHAVIORAL RELATIONSHIPS

Sequence from «entity» GUI to End_Inventory_APP

Sequence

Sequence owned by 'Inventory_APP', in package 'Primary Use Cases'

Sequence
Version 1.0 Phase 1.0 Proposed
corongiu created on 14/03/2024. Last modified 14/03/2024

OUTGOING BEHAVIORAL RELATIONSHIPS

Sequence from Sequence to DataCollection

OUTGOING BEHAVIORAL RELATIONSHIPS

GeoDB

Sequence «entity» owned by 'Inventory_APP', in package 'Primary Use Cases'

GeoDB
Version 1.0 Phase 1.0 Proposed
corongiu created on 14/03/2024. Last modified 14/03/2024

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: API
Sequence from «entity» GeoDB to «entity» GUI

INCOMING BEHAVIORAL RELATIONSHIPS

Name: API
Sequence from DataCollection to «entity» GeoDB

GUI

Sequence «entity» owned by 'Inventory_APP', in package 'Primary Use Cases'

GUI
Version 1.0 Phase 1.0 Proposed
corongiu created on 14/03/2024. Last modified 14/03/2024

OUTGOING BEHAVIORAL RELATIONSHIPS

Sequence from «entity» GUI to End_Inventory_APP

INCOMING BEHAVIORAL RELATIONSHIPS

Name: API
Sequence from «entity» GeoDB to «entity» GUI

Inventory_Manual

UseCase in package 'Primary Use Cases'

Inventory_Manual
Version 1.0 Phase 1.0 Proposed
corongiu created on 14/03/2024. Last modified 14/03/2024

ELEMENTS OWNED BY Inventory_Manual
Sequence : Sequence
DataCollection : Event
End_Inventory_Manual : MessageEnd
GeoDB : Sequence «entity»
GUI : Sequence «entity»

ASSOCIATIONS	
Association (direction: Unspecified)	
Source: Public (Actor) User_Inventory_Manual	Target: Public (UseCase) Inventory_Manual

Inventory_Manual Use Case diagram

Interaction diagram in package 'Primary Use Cases'

Inventory_Manual Use Case
Version 1.0
corongiu created on 14/03/2024. Last modified 15/03/2024

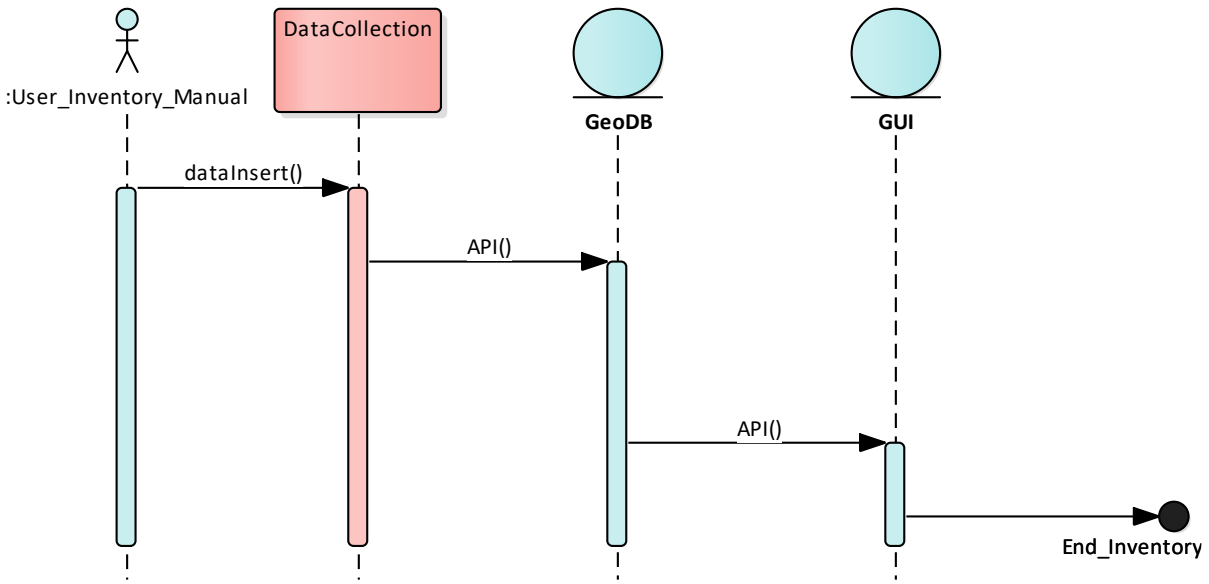


Figure 13: USE Case Model – Inventory_Manual use case Diagram

INTERACTION MESSAGES

1.0 'dataInsert' from ':User_Inventory_Manual' sent to 'DataCollection'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.1 'API' from 'DataCollection' sent to 'GeoDB'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.2 'API' from 'GeoDB' sent to 'GUI'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.3 '' from 'GUI' sent to 'End_Inventory_Manual'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

Sequence

Sequence owned by 'Inventory_Manual', in package 'Primary Use Cases'

Sequence
Version 1.0 Phase 1.0 Proposed
corongiu created on 14/03/2024. Last modified 14/03/2024

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: dataInsert
Sequence from Sequence to DataCollection

DataCollection

Event owned by 'Inventory_Manual', in package 'Primary Use Cases'

DataCollection
Version 1.0 Phase 1.0 Proposed
corongiu created on 14/03/2024. Last modified 14/03/2024

End_Inventory_Manual

MessageEnd owned by 'Inventory_Manual', in package 'Primary Use Cases'

End_Inventory_Manual
Version 1.0 Phase 1.0 Proposed

corongiu created on 14/03/2024. Last modified 15/03/2024

INCOMING BEHAVIORAL RELATIONSHIPS

Sequence from «entity» GUI to End_Inventory_Manual

GeoDB

Sequence «entity» owned by 'Inventory_Manual', in package 'Primary Use Cases'

GeoDB
Version 1.0 Phase 1.0 Proposed
corongiu created on 14/03/2024. Last modified 14/03/2024

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: API
Sequence from «entity» GeoDB to «entity» GUI

INCOMING BEHAVIORAL RELATIONSHIPS

Name: API
Sequence from DataCollection to «entity» GeoDB

GUI

Sequence «entity» owned by 'Inventory_Manual', in package 'Primary Use Cases'

GUI
Version 1.0 Phase 1.0 Proposed
corongiu created on 14/03/2024. Last modified 14/03/2024

OUTGOING BEHAVIORAL RELATIONSHIPS

Sequence from «entity» GUI to End_Inventory_Manual

INCOMING BEHAVIORAL RELATIONSHIPS

Name: API
Sequence from «entity» GeoDB to «entity» GUI

Inventory_RS

UseCase in package 'Primary Use Cases'

Inventory_RS

Version 1.0 Phase 1.0 Proposed
corongiu created on 14/03/2024. Last modified 14/03/2024

ELEMENTS OWNED BY Inventory_RS
Sequence : Sequence
DataCollection : Event
End_Inventory_RS : MessageEnd
GeoDB : Sequence «entity»
GUI : Sequence «entity»

ASSOCIATIONS	
Association (direction: Unspecified)	
Source: Public (Actor) User_Inventory_RS	Target: Public (UseCase) Inventory_RS

Inventory_RS Use Case diagram

Interaction diagram in package 'Primary Use Cases'

Inventory_RS Use Case
Version 1.0
corongiu created on 14/03/2024. Last modified 15/03/2024

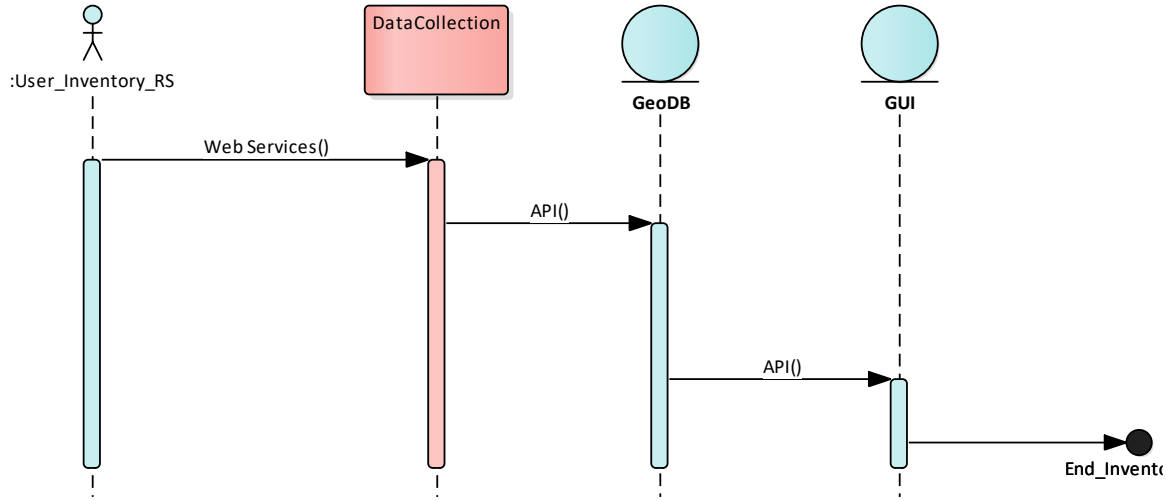


Figure 14: USE Case Model – Inventory_RS use case Diagram

INTERACTION MESSAGES
1.0 'Web Services' from ':User_Inventory_RS' sent to 'DataCollection'. Synchronous Call. Returns void. [Return is False. Iteration is False. New group is False.]
1.1 'API' from 'DataCollection' sent to 'GeoDB'. Synchronous Call. Returns void. [Return is False. Iteration is False. New group is False.]
1.2 'API' from 'GeoDB' sent to 'GUI'. Synchronous Call. Returns void. [Return is False. Iteration is False. New group is False.]
1.3 '' from 'GUI' sent to 'End_Inventory_RS'. Synchronous Call. Returns void. [Return is False. Iteration is False. New group is False.]

Sequence

Sequence owned by 'Inventory_RS', in package 'Primary Use Cases'

Sequence
Version 1.0 Phase 1.0 Proposed
corongiu created on 14/03/2024. Last modified 14/03/2024

OUTGOING BEHAVIORAL RELATIONSHIPS
Name: Web Services Sequence from Sequence to DataCollection

DataCollection

Event owned by 'Inventory_RS', in package 'Primary Use Cases'

DataCollection
Version 1.0 Phase 1.0 Proposed
corongiu created on 14/03/2024. Last modified 14/03/2024

End_Inventory_RS

MessageEnd owned by 'Inventory_RS', in package 'Primary Use Cases'

End_Inventory_RS
Version 1.0 Phase 1.0 Proposed
corongiu created on 14/03/2024. Last modified 15/03/2024

INCOMING BEHAVIORAL RELATIONSHIPS

Sequence from «entity» GUI to End_Inventory_RS

GeoDB

Sequence «entity» owned by 'Inventory_RS', in package 'Primary Use Cases'

GeoDB
Version 1.0 Phase 1.0 Proposed
corongiu created on 14/03/2024. Last modified 14/03/2024

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: API
Sequence from «entity» GeoDB to «entity» GUI

INCOMING BEHAVIORAL RELATIONSHIPS

Name: API
Sequence from DataCollection to «entity» GeoDB

GUI

Sequence «entity» owned by 'Inventory_RS', in package 'Primary Use Cases'

GUI
Version 1.0 Phase 1.0 Proposed
corongiu created on 14/03/2024. Last modified 14/03/2024

OUTGOING BEHAVIORAL RELATIONSHIPS

Sequence from «entity» GUI to End_Inventory_RS

INCOMING BEHAVIORAL RELATIONSHIPS

Name: API
Sequence from «entity» GeoDB to «entity» GUI

Inventory_TLS

UseCase in package 'Primary Use Cases'

Inventory_TLS
Version 1.0 Phase 1.0 Proposed
corongiu created on 14/03/2024. Last modified 14/03/2024

ELEMENTS OWNED BY Inventory_TLS
Sequence : Sequence
DataCollection : Event
End_Inventory_TLS : MessageEnd
GeoDB : Sequence «entity»
GUI : Sequence «entity»

ASSOCIATIONS
Association (direction: Unspecified)
Source: Public (Actor) User_Inventory_TLS — Target: Public (UseCase) Inventory_TLS

Inventory_TLS Use Case diagram

Interaction diagram in package 'Primary Use Cases'

Inventory_TLS Use Case
Version 1.0
corongiu created on 14/03/2024. Last modified 15/03/2024

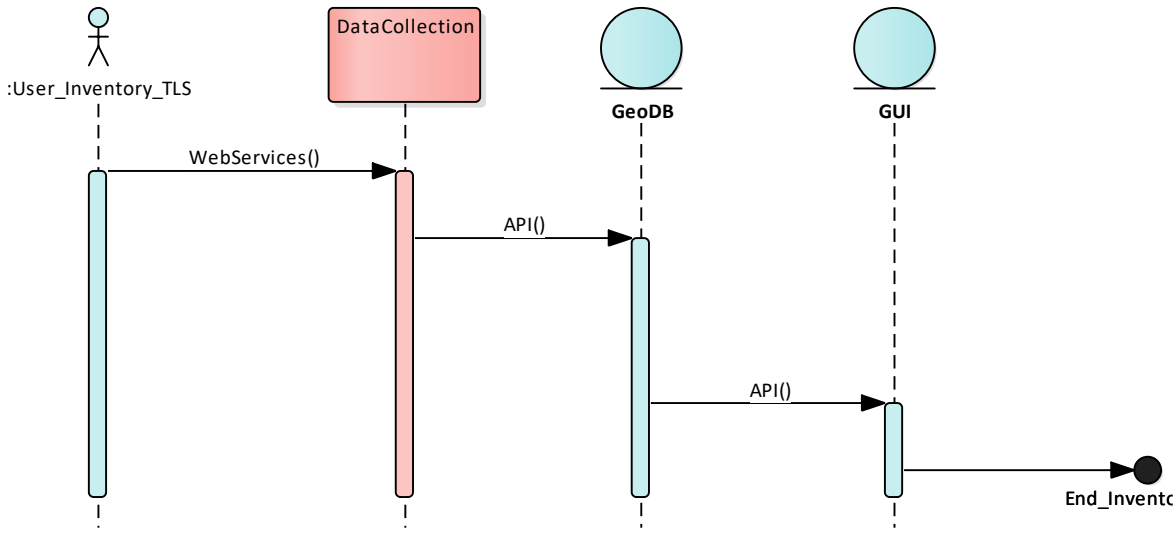


Figure 15: USE Case Model – Inventory_TLS use case Diagram

INTERACTION MESSAGES

1.0 'WebServices' from ':User_Inventory_TLS' sent to 'DataCollection'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.1 'API' from 'DataCollection' sent to 'GeoDB'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.2 'API' from 'GeoDB' sent to 'GUI'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.3 '' from 'GUI' sent to 'End_Inventory_TLS'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

Sequence

Sequence owned by 'Inventory_TLS', in package 'Primary Use Cases'

Sequence
Version 1.0 Phase 1.0 Proposed
corongiu created on 14/03/2024. Last modified 14/03/2024

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: WebServices
Sequence from Sequence to DataCollection

DataCollection

Event owned by 'Inventory_TLS', in package 'Primary Use Cases'

DataCollection
Version 1.0 Phase 1.0 Proposed
corongiu created on 14/03/2024. Last modified 14/03/2024

End_Inventory_TLS

MessageEnd owned by 'Inventory_TLS', in package 'Primary Use Cases'

End_Inventory_TLS
Version 1.0 Phase 1.0 Proposed
corongiu created on 14/03/2024. Last modified 15/03/2024

INCOMING BEHAVIORAL RELATIONSHIPS

Sequence from «entity» GUI to End_Inventory_TLS

GeoDB

Sequence «entity» owned by 'Inventory_TLS', in package 'Primary Use Cases'

GeoDB
Version 1.0 Phase 1.0 Proposed
corongiu created on 14/03/2024. Last modified 14/03/2024

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: API
Sequence from «entity» GeoDB to «entity» GUI

INCOMING BEHAVIORAL RELATIONSHIPS

Name: API
Sequence from DataCollection to «entity» GeoDB

GUI

Sequence «entity» owned by 'Inventory_TLS', in package 'Primary Use Cases'

GUI
Version 1.0 Phase 1.0 Proposed
corongiu created on 14/03/2024. Last modified 14/03/2024

OUTGOING BEHAVIORAL RELATIONSHIPS

Sequence from «entity» GUI to End_Inventory_TLS

INCOMING BEHAVIORAL RELATIONSHIPS

Name: API
Sequence from «entity» GeoDB to «entity» GUI

Modelling

UseCase in package 'Primary Use Cases'

WP5

Modelling
Version 1.0 Phase 1.0 Proposed
corongiu created on 15/03/2024. Last modified 15/03/2024

ELEMENTS OWNED BY Modelling
End_Output_GeoDB : MessageEnd
Sequence : Sequence
Modelling : Event
End_Modelling : MessageEnd
GeoDB : Sequence «entity»
GUI : Sequence «entity»

ASSOCIATIONS	
Association (direction: Unspecified)	
Source: Public (Actor) User_Modelling	Target: Public (UseCase) Modelling

Modelling Use Case diagram

Interaction diagram in package 'Primary Use Cases'

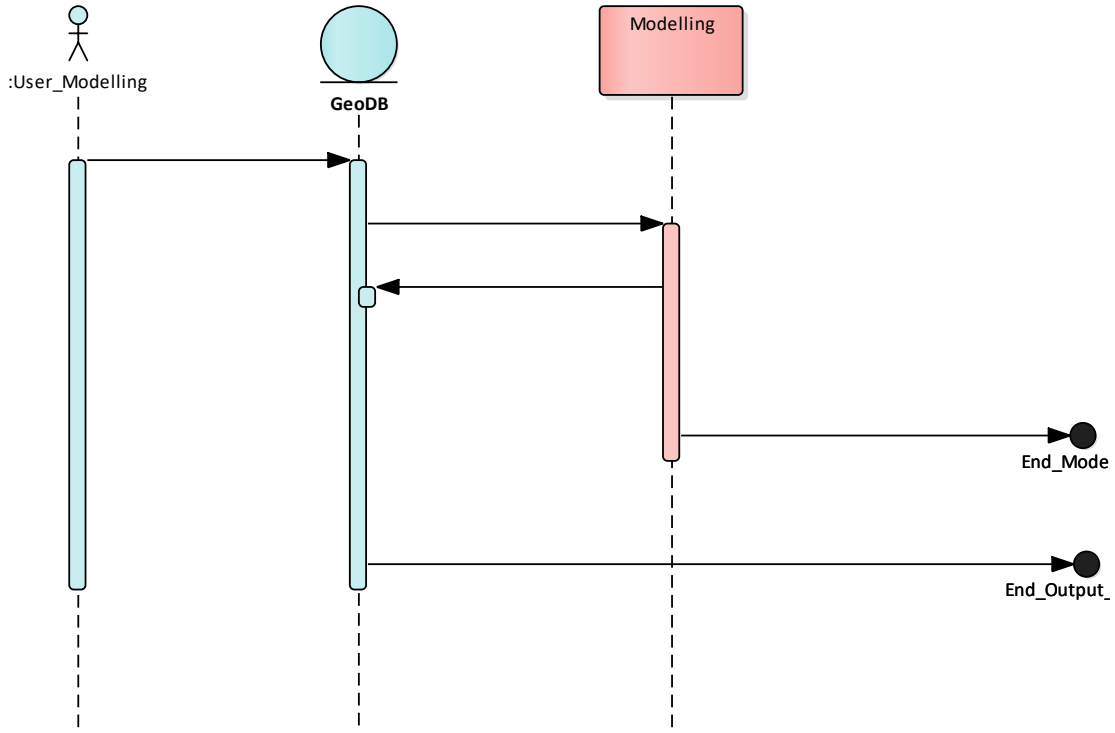


Figure 16: USE Case Model – Modelling use case Diagram

INTERACTION MESSAGES
1.0 '' from ':User_Modelling' sent to 'GeoDB'. Synchronous Call. Returns void. [Return is False. Iteration is False. New group is False.]
1.1 '' from 'GeoDB' sent to 'Modelling'. Synchronous Call. Returns void. [Return is False. Iteration is False. New group is False.]
1.2 '' from 'Modelling' sent to 'GeoDB'. Synchronous Call. Returns void. [Return is False. Iteration is False. New group is False.]
1.3 '' from 'Modelling' sent to 'End_Modelling'. Synchronous Call. Returns void. [Return is False. Iteration is False. New group is False.]
1.4 '' from 'GeoDB' sent to 'End_Output_GeoDB'. Synchronous Call. Returns void. [Return is False. Iteration is False. New group is False.]

End_Output_GeoDB

MessageEnd owned by 'Modelling', in package 'Primary Use Cases'

End_Output_GeoDB
Version 1.0 Phase 1.0 Proposed
corongiu created on 15/03/2024. Last modified 15/03/2024

INCOMING BEHAVIORAL RELATIONSHIPS

Sequence from «entity» GeoDB to End_Output_GeoDB

Sequence

Sequence owned by 'Modelling', in package 'Primary Use Cases'

Sequence
Version 1.0 Phase 1.0 Proposed
corongiu created on 15/03/2024. Last modified 15/03/2024

OUTGOING BEHAVIORAL RELATIONSHIPS

Sequence from Sequence to «entity» GeoDB

Modelling

Event owned by 'Modelling', in package 'Primary Use Cases'

Modelling
Version 1.0 Phase 1.0 Proposed
corongiu created on 15/03/2024. Last modified 15/03/2024

End_Modelling

MessageEnd owned by 'Modelling', in package 'Primary Use Cases'

End_Modelling
Version 1.0 Phase 1.0 Proposed
corongiu created on 15/03/2024. Last modified 15/03/2024

INCOMING BEHAVIORAL RELATIONSHIPS

Sequence from Modelling to End_Modelling

GeoDB

Sequence «entity» owned by 'Modelling', in package 'Primary Use Cases'

GeoDB
Version 1.0 Phase 1.0 Proposed
corongiu created on 15/03/2024. Last modified 15/03/2024

OUTGOING BEHAVIORAL RELATIONSHIPS
Sequence from «entity» GeoDB to Modelling
Sequence from «entity» GeoDB to End_Output_GeoDB

INCOMING BEHAVIORAL RELATIONSHIPS
Sequence from Modelling to «entity» GeoDB
Sequence from Sequence to «entity» GeoDB

GUI

Sequence «entity» owned by 'Modelling', in package 'Primary Use Cases'

GUI
Version 1.0 Phase 1.0 Proposed
corongiu created on 15/03/2024. Last modified 15/03/2024

Modelling_RS

UseCase in package 'Primary Use Cases'

WP5

Modelling_RS
Version 1.0 Phase 1.0 Proposed
corongiu created on 15/03/2024. Last modified 15/03/2024

ELEMENTS OWNED BY Modelling_RS
RS_analysis : Event
Sequence : Sequence
End_Modelling_RS : MessageEnd
GeoDB : Sequence «entity»
GUI : Sequence «entity»

ASSOCIATIONS

Association (direction: Unspecified)

Source: Public (Actor) User_Modelling_RS

Target: Public (UseCase) Modelling_RS

Modelling_RS Use Case diagram

Interaction diagram in package 'Primary Use Cases'

Modelling_RS Use Case
Version 1.0
corongiu created on 15/03/2024. Last modified 15/03/2024

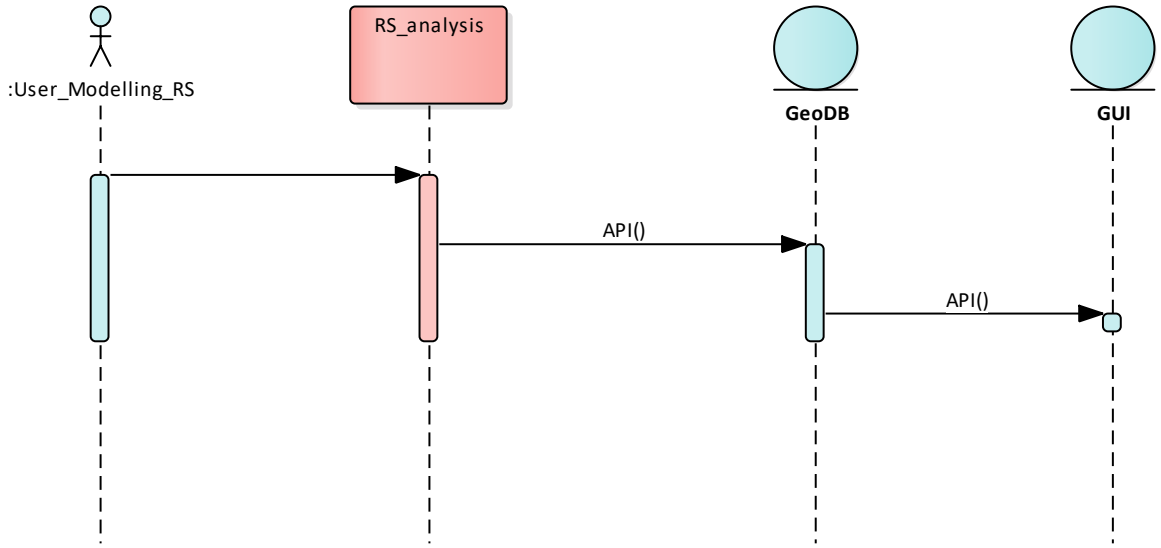


Figure 17: USE Case Model – Modelling_RS use case Diagram

INTERACTION MESSAGES

1.0 '' from ':User_Modelling_RS' sent to 'RS_analysis'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.1 'API' from 'RS_analysis' sent to 'GeoDB'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.2 'API' from 'GeoDB' sent to 'GUI'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

RS_analysis
Event owned by 'Modelling_RS', in package 'Primary Use Cases'

RS_analysis
Version 1.0 Phase 1.0 Proposed
corongiu created on 15/03/2024. Last modified 15/03/2024

Sequence
Sequence owned by 'Modelling_RS', in package 'Primary Use Cases'

Sequence
Version 1.0 Phase 1.0 Proposed
corongiu created on 15/03/2024. Last modified 15/03/2024

OUTGOING BEHAVIORAL RELATIONSHIPS

Sequence from Sequence to RS_analysis

End_Modelling_RS
MessageEnd owned by 'Modelling_RS', in package 'Primary Use Cases'

End_Modelling_RS
Version 1.0 Phase 1.0 Proposed
corongiu created on 15/03/2024. Last modified 15/03/2024

GeoDB
Sequence «entity» owned by 'Modelling_RS', in package 'Primary Use Cases'

GeoDB
Version 1.0 Phase 1.0 Proposed
corongiu created on 15/03/2024. Last modified 15/03/2024

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: API
Sequence from «entity» GeoDB to «entity» GUI

INCOMING BEHAVIORAL RELATIONSHIPS

Name: API
Sequence from RS_analysis to «entity» GeoDB

GUI
Sequence «entity» owned by 'Modelling_RS', in package 'Primary Use Cases'

GUI
Version 1.0 Phase 1.0 Proposed
corongiu created on 15/03/2024. Last modified 15/03/2024

INCOMING BEHAVIORAL RELATIONSHIPS
Name: API Sequence from «entity» GeoDB to «entity» GUI

Sawmill

UseCase in package 'Primary Use Cases'

Sawmill
Version 1.0 Phase 1.0 Proposed
corongiu created on 15/03/2024. Last modified 15/03/2024

ELEMENTS OWNED BY Sawmill
Scanner : Event
Sequence : Sequence
End_Sawmill : MessageEnd
InternalDB : Object «entity» Sawmill DataStore
GeoDB : Sequence «entity»
GUI : Sequence «entity»

ASSOCIATIONS	
Association (direction: Unspecified)	
Source: Public (Actor) User_Sawmill	Target: Public (UseCase) Sawmill

Sawmill Use Case diagram

Interaction diagram in package 'Primary Use Cases'

Sawmill Use Case
Version 1.0
corongiu created on 15/03/2024. Last modified 15/03/2024

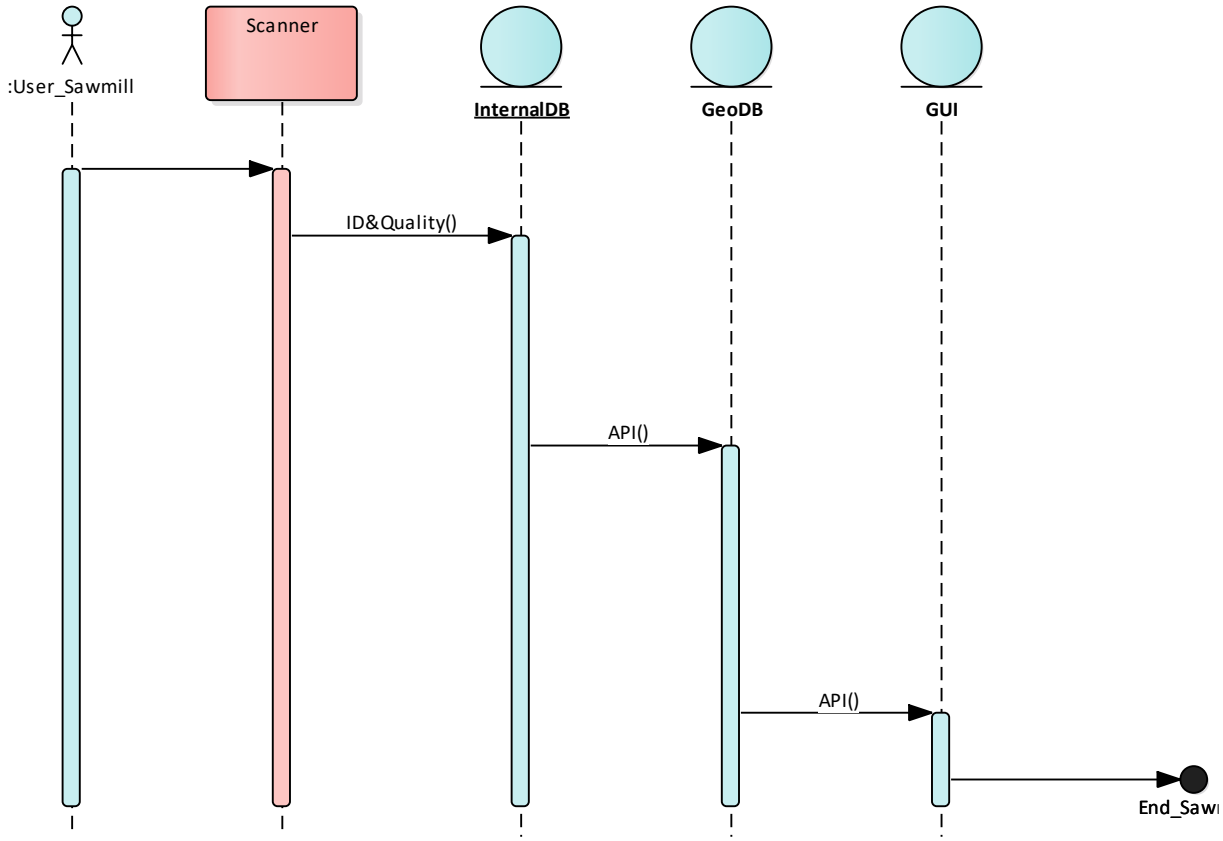


Figure 18: USE Case Model – Sawmill use case Diagram

INTERACTION MESSAGES
1.0 '' from ':User_Sawmill' sent to 'Scanner'. Synchronous Call. Returns void. [Return is False. Iteration is False. New group is False.]
1.1 'ID&Quality' from 'Scanner' sent to 'InternalDB'. Synchronous Call. Returns void. [Return is False. Iteration is False. New group is False.]
1.2 'API' from 'InternalDB' sent to 'GeoDB'. Synchronous Call. Returns void. [Return is False. Iteration is False. New group is False.]
1.3 'API' from 'GeoDB' sent to 'GUI'. Synchronous Call. Returns void. [Return is False. Iteration is False. New group is False.]

INTERACTION MESSAGES

1.4 '' from 'GUI' sent to 'End_Sawmill'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

Scanner

Event owned by 'Sawmill', in package 'Primary Use Cases'

Scanner+Tomography

Scanner
Version 1.0 Phase 1.0 Proposed
corongiu created on 15/03/2024. Last modified 15/03/2024

Sequence

Sequence owned by 'Sawmill', in package 'Primary Use Cases'

Sequence
Version 1.0 Phase 1.0 Proposed
corongiu created on 15/03/2024. Last modified 15/03/2024

OUTGOING BEHAVIORAL RELATIONSHIPS

Sequence from Sequence to Scanner

End_Sawmill

MessageEnd owned by 'Sawmill', in package 'Primary Use Cases'

End_Sawmill
Version 1.0 Phase 1.0 Proposed
corongiu created on 15/03/2024. Last modified 15/03/2024

INCOMING BEHAVIORAL RELATIONSHIPS

Sequence from «entity» GUI to End_Sawmill

InternalDB

Object «entity» owned by 'Sawmill', in package 'Primary Use Cases'

Sawmill DataStore

InternalDB
Version 1.0 Phase 1.0 Proposed
corongiu created on 15/03/2024. Last modified 15/03/2024

GeoDB

Sequence «entity» owned by 'Sawmill', in package 'Primary Use Cases'

GeoDB
Version 1.0 Phase 1.0 Proposed
corongiu created on 15/03/2024. Last modified 15/03/2024

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: API
Sequence from «entity» GeoDB to «entity» GUI

INCOMING BEHAVIORAL RELATIONSHIPS

Name: API
Sequence from «entity» InternalDB to «entity» GeoDB

GUI

Sequence «entity» owned by 'Sawmill', in package 'Primary Use Cases'

GUI
Version 1.0 Phase 1.0 Proposed
corongiu created on 15/03/2024. Last modified 15/03/2024

OUTGOING BEHAVIORAL RELATIONSHIPS

Sequence from «entity» GUI to End_Sawmill

INCOMING BEHAVIORAL RELATIONSHIPS

Name: API
Sequence from «entity» GeoDB to «entity» GUI

D1.5 Geospatial and platform data model, conceptual schema - Annexes

Annex 1: Meetings between partners involved to get info for the GeoDB design

Below are the minutes of the meetings with the partners to get information about input and output data and to arrive at the definition of the Sintetic GeoDB design (task 1.3).

APP data input to geodatabase - 07th of February 20244

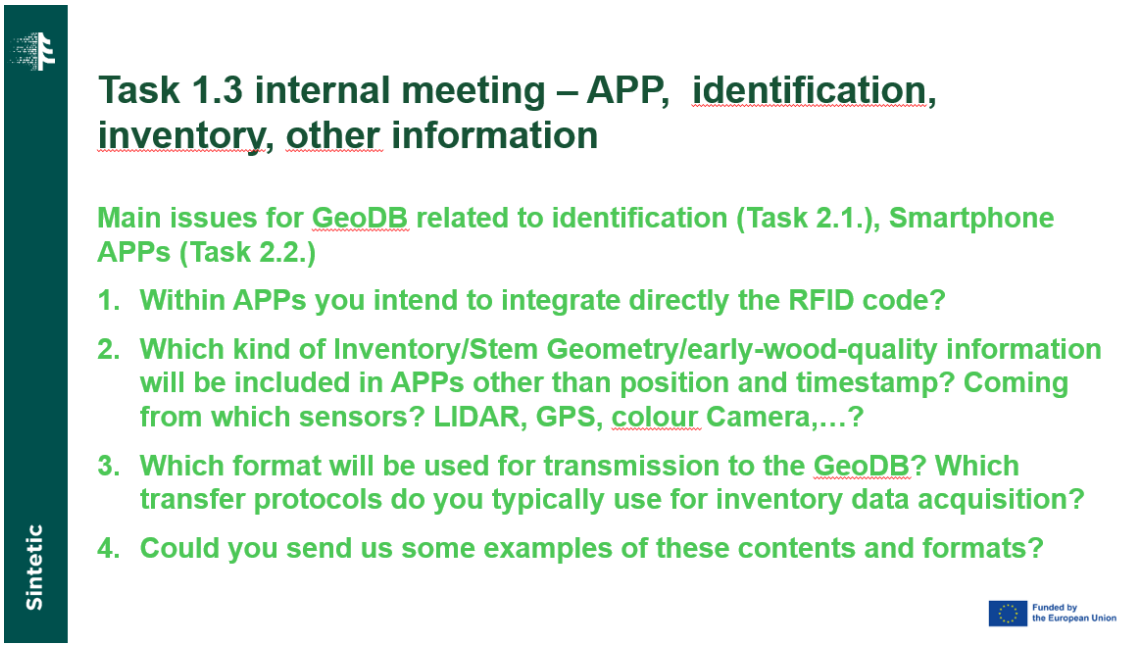
Scope, location and general information

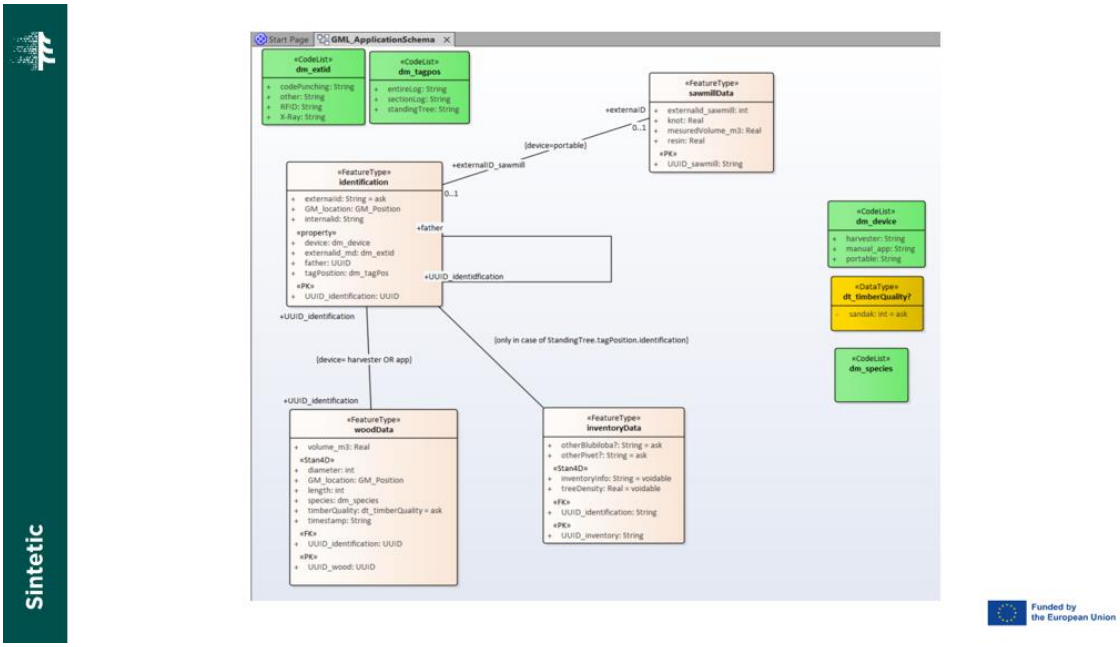
Zoom meeting planned 07th of February 2024 to get some information about data from the APPs that would be used as input for the SINTETIC GeoDB design (task 1.3). Presentation and discussion of the first draft of GML application schema with particular attention to “FeaturesTypes” definition for traceability system.

Meeting minutes

Manuela Corongiu directed the meeting asking questions to partners.

Presentation 1 – Manuela Corongiu (LAMMA):

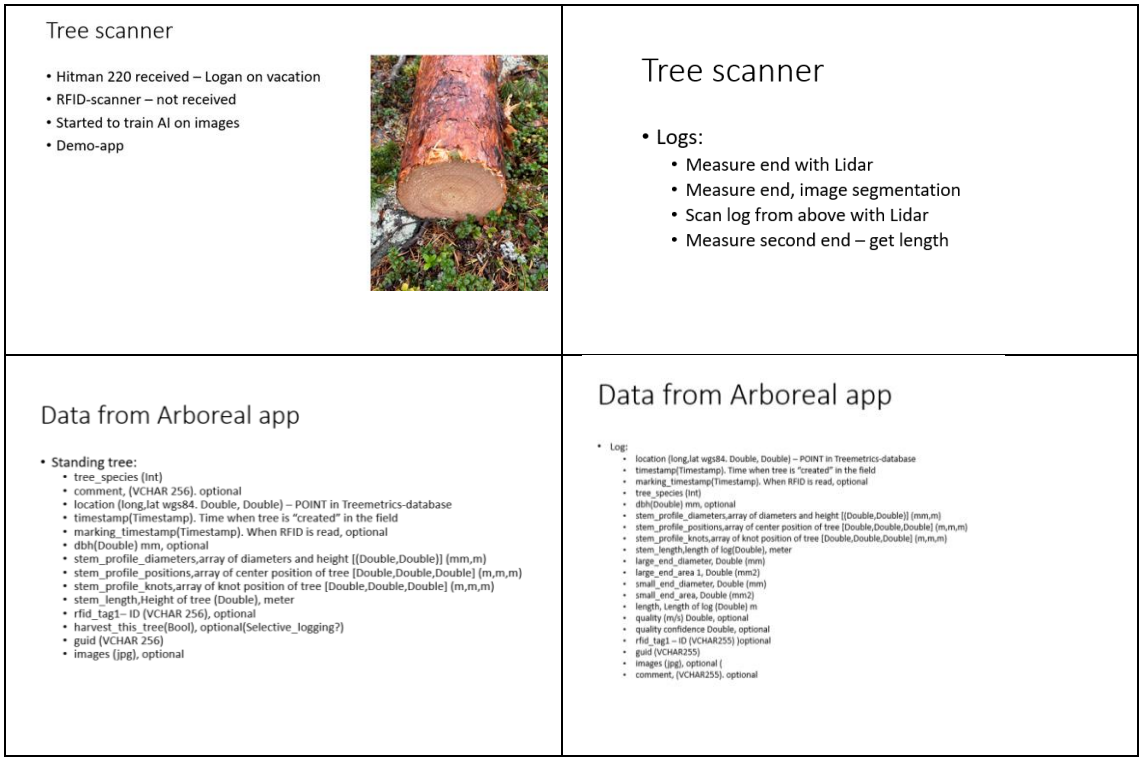




Discussion Points:

The main issues for GeoDB related to identification (Task2.1.), and Smartphone APPs (Task 2.2.) were discussed. These points were answered by Johan Ekenstedt (Arboreal), Alex Poveda and Garret Mullooly (Treemetrics) and Guido Milazzo and Cristiano Guadagnino (Bluebiloba)

Presentation 2 - Johan Ekenstedt (Arboreal) explained the number of variables and expected data format to be sent to the Geodatabase



Johan Ekenstedt (Arboreal) showed us a demo of the steps that will be used to gather information from the logs.

Although he said that the data won't be in STANForD format, from Treemetrics they said that it would be easy to transform the data into that format if needed. Although they pointed out that the files are heavier than if we use another format.

For the APP, used terminologies and units should be the ones of StanForD 2010 ontology. Library codes and information are sent by Garret Mullooly (Treemetrics) to Johan Ekenstedt (Arboreal) and all participants.

Alex Poveda (Treemetrics) explained and showed us an excel file with all the types of files and relations that can be obtained from the actual StanForD from the harvester. Also, some information regarding the metadata. More details on StanForD format and contents are available on https://www.skogforsk.se/english/projects/stanford/stanford-2010/

Alex Poveda (Treemetrics) presented the working in progress on standardization of attribute names and type between apps data and StanForD Name.

Example extracted from Tree Data Definition.xlxs:

Table	tree					**StanForD Name**
Name	**Type**	**Data**	**Required**	**Units**	**Description**	
id	integer	ID	YES		Automated database ID	
stand_id	integer	stand_id	YES		"stand" object	
tree_id	string	VCHAR(255)	YES		Initial tree ID (from UAV /TLS analysis)	
field_id	string	VCHAR(255)	NO		Field marks ID (E.g. number painted in the tree)	
rfid_tag1	string	VCHAR(255)	NO		RFID tag ID	
rfid_tag2	string	VCHAR(255)	NO		Second RFID tag ID (when required?)	
tag_type	string	VCHAR(255)	NO		To be defined?	

Some parameters, methodologies and technical solutions that will be standardized in SINTETIC have been discussed.

- The volume under bark could be estimated using specific allometry equations for each species.
- Treemetrics suggested that it would be better to take the info regarding the Diameter Reading in mm and the lengths in dm.
- Terminology and units used should be the one in Stan4D 2010 (protocol) ontology for the APP.
- The position of the bucked log in the tree should be stored together with other stem measurements and crosscut information.
- Both the Punching code and the RFID need to be stored and related to a Universal Unique Identifier (UUID).

- When placing RFID, it will be necessary to take into account that it will be placed on standing trees, logs (trunks) and logSections (logs), connected together.

- LiDAR data should be processed to obtain parameters but not transferred immediately as the files would weigh a lot. Some prefiltering should be done before storing it in the Geodatabase (makes sense to store the filtered point cloud?)

Guido Milazzo and Cristiano Guadagnino (Bluebiloba) also shared information that could be included in the SINTETIC geodatabase, but further discussion is necessary with the FMMF partner about GDPR 2016/679 regulation because some collected data regard forest owners.

Geometry of the property, terrain features, type of silviculture, basal area or volume that should be removed in the forest operation, risk of fire (type of fuel/amount of fuel): this info will be sent in GeoJson format. An example of data structure from Bluebiloba App will be sent to T.13 to LAMMA.

Final Remarks

The LAMMA and CNR will discuss in the next steps of Task 1.3 which kind of APIs will be developed to upload StanForD data into the Geodb from Harvester and Apps, and which API to upload GeoJson data from App inventory. Who will be in charge of these developments will be discussed inside WP1 and WP3.

Based on the meeting outcomes, the following actions are noted/planned (urgent actions in **green bold).**

Partner in charge	When	Related WP/Task	Action Required
Acronym	Date	Description	Description
ARBOREAL	08/02/2024	T 1.3	**Data structure from App:** example
TREE	08/02/2024	T 1.3	**Data structure from App:** example
BLUBILOBA	08/02/2024	T 1.3	**Data structure from app:** example

Harvester prototype data, identification, other information - 08th of February 2024

Scope, location and general information

Meeting to get some information about data from the Harvester prototype, identification methods and devices, and Lidar data that would be used as input for the Sintetic GeoDB design (task 1.3)

Meeting minutes

Manuela Corongiu directed the meeting asking questions to the different partners.

Presentation 1 – Manuela Corongiu (LAMMA):

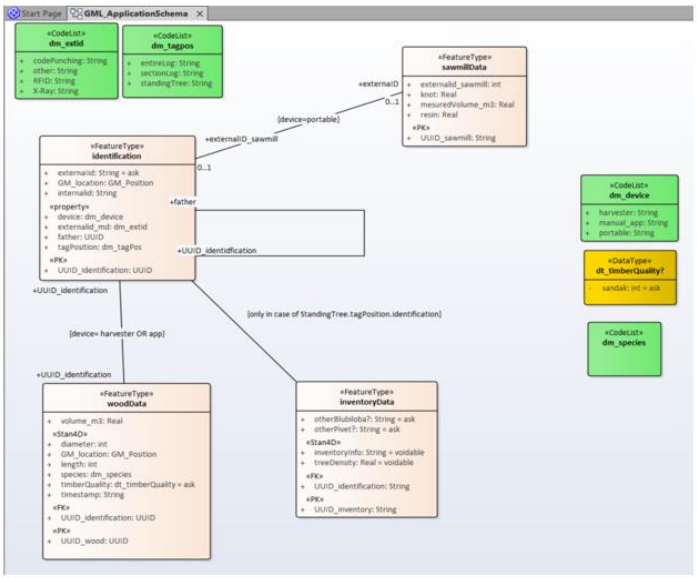
Task 1.3 Internal meeting – harvester prototype, identification, other information

Main issues for GeoDB related to identification (Task 2.1.), harvester (Task 2.3)

1. **Which kind of ID detection system (Task 2.1) for mechanized forest harvesters (Task 2.3) is available?**
2. **The output communication is always in Stan4D? Which S4D file name extension (categories)? Which "ready-to-use" solutions to transfer data to the GeoDB?**
3. **Lidar data need to be included as-is in the geoDB? which format?**
4. **Which quality information, and parameters have to be included in (e.g. coming from sensors providing bucking suggestions?**
5. **Could you send us some examples of these contents and formats?**

Sintetic

Funded by the European Union



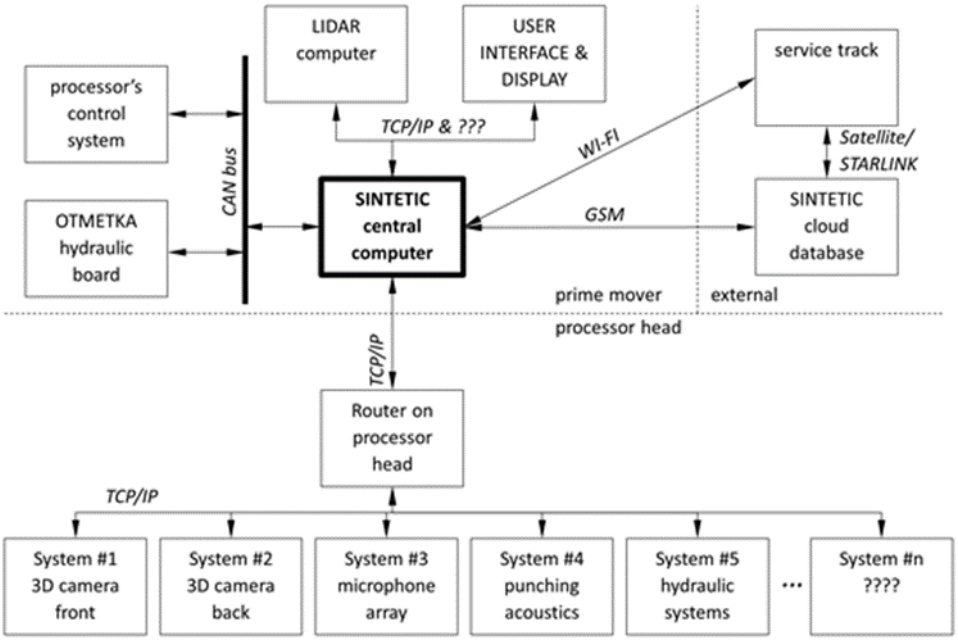
Sintetic

Funded by the European Union

Discussion Points:

The main issues for GeoDB are related to identification (Task2.1.), and harvester (Task 2.3.) were discussed. These points were answered by Jakub Sandak (Innorenew), Alex Poveda and Garret Mullooly (Treemetrics).

Jakub (INNORENEW) explain the process about harvesting machine and sent the related confidential documentation. A syntesis of the process is represented below.



Jakub (INNO) describes data types collected in harvester phase:

3D camera will generate cloud of point that we will process and convert into something more “useful”.

Microphones will record sound, but we will process it into knot positions and/or log stiffness.

Pressure sensors will generate voltage (proportional to the pressure of hydraulic oil)

To be more specifically:

1. The core data from processor is customized StandForD file with additional parameters (maybe not part of the standard structure):

2. OTMETKA_ID,

3. SIMTRONA_ID,

4. standard_quality_class (A B C D),

5. suitability index (INNORENEW proposal) – can be ten number from 0 to1 (or 0 to 255) defining the suitability for different downstream conversion.

6. File with LIDAR 3D point cloud (saved in its native format)

7. File with all raw data generated by sensors of the processor (we will define the format and presentation, only referenced in your DB (and not included as such)

From Treemetrics a suggestion to relate lidar data before cut to stan4D new class e.g. suitability class as quality information.

To recap discussion, main suggestions are:

- New variable integration on the S4D standard, as extensions (to evaluate)

- Timestamp not necessary aligned on time

- Lidar Data to be uploaded in Sintetic GeoDB (to evaluate if raw or derived)
- Relate each information coming from the device sensors
- Structure quality data in terms of properties of wood (about pulp wood, logs, branches, etc.)
- Take into account lidar metadata info, if available

All partners are engaged to study in deep Stand4D structure and XSD schemas for each category and evaluate Extension or external files relationship (https://www.skogforsk.se/english/projects/stanford/).

Final Remarks

Based on the meeting outcomes, the following actions are noted/planned (urgent actions in **green bold**).

Partner in charge	When	Related WP/Task	Action Required
Acronym	DD/MM/YYYY	Description	Description
INNORENEW	08/02/2024	Task 2.3	Confidential document

Sawmill property quality information and link to identification - 09th of February 2024

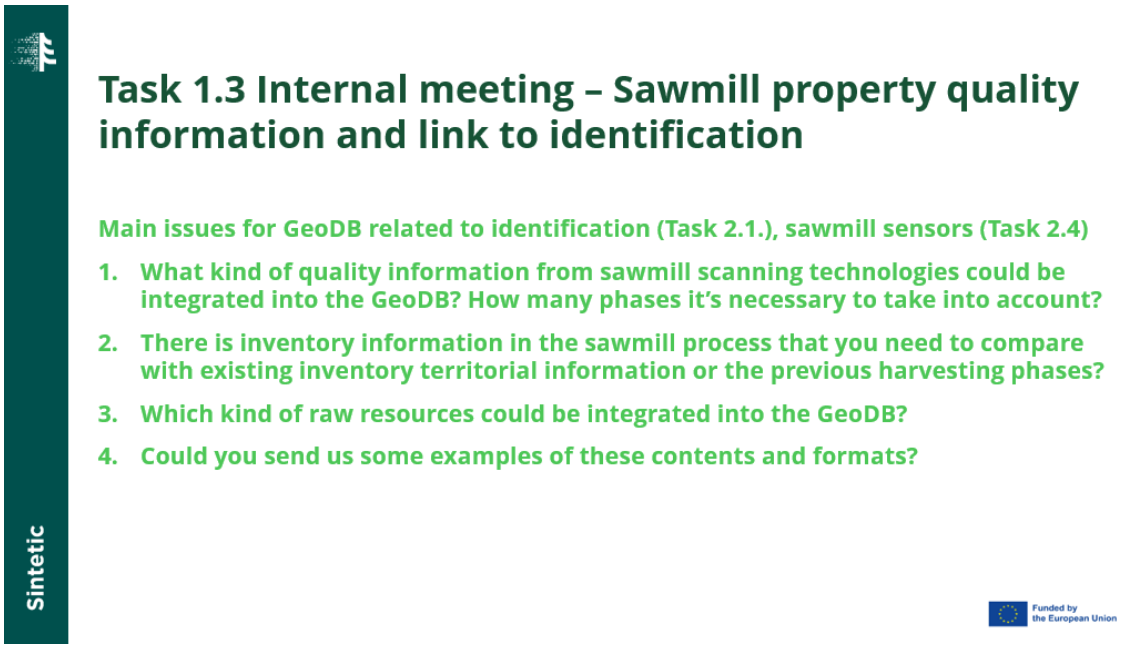
Scope, location and general information

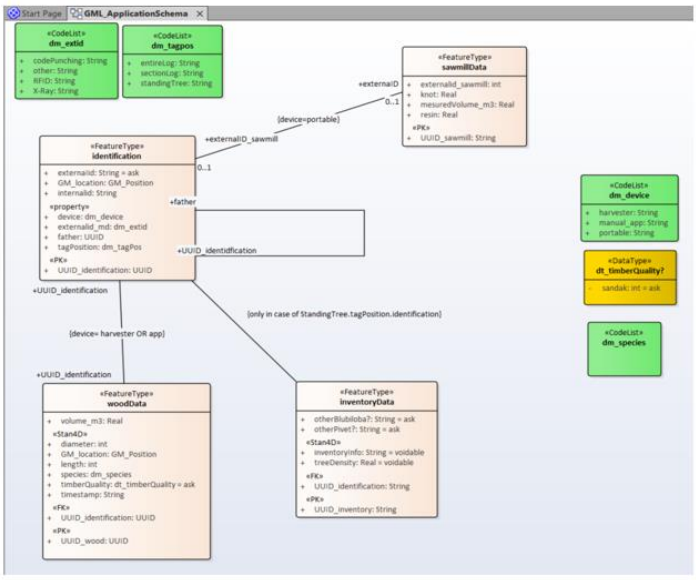
Meeting to get some information about property quality data from sawmills that would be used as input for the Sintetic GeoDB design (task 1.3)

Meeting minutes

Manuela Corongiu directed the meeting asking questions to the different partners.

Presentation 1 – Manuela Corongiu (LAMMA):





Funded by the European Union

Discussion Points:

Enrico Ursella (MICROTEC) presented the list of the features we could export per each product on each scanner.

ID of the stem/log.
Year, month, day, hour, minute, second of the scan
species
log large diameter (mm), log small diameter (mm), log length(mm), log volume (dm3)
sum of the volume of knots (dm3), number of knots, average diameter of the knots (mm), max knot diameter(mm), average length(mm).
for the sound knots only: sum, number, average diameter, max diameter, average length. A knot is classified as sound if the dead part is less than 5% of its length.
for the dead knots only: sum, number, average diameter, max diameter, average length. A knot is classified as dead if the dead part is more than 5% of its length.
Percentage of heartwood volume respect to the total volume (%), average green density of the heartwood (g/dm3), average green density of the sapwood (g/dm3).
Max pith deviation (mm). Deviation of the pith respect to a regular trajectory. This is a useful feature for detection of top breaks.
Maxima eccentricity (mm). This is the distance between the pith and the centroid of the log. It is useful to indicate the presence of compression wood.

For the boards a list of possible features is:

ID of the board.
Year, month, day, hour, minute, second of the scan
species
board average width(mm), thickness(mm) length(mm)
number of knots, average diameter of the knots (mm), max knot diameter(mm)
for the sound knots only: number, average diameter, max diameter
for the dead knots only: number, average diameter, max diameter
Percentage of brown stain and blue stain
Number of resin pockets
Sum of the length of all cracks

Ugo Gerard (Piveteau) reports the sawmill receives long stems that are bucked in shorter logs so that they can produce data for both stem and logs, as reported above.

Enrico Ursella (MICROTEC) reports increasing demand for biological properties and knot information in sawmill products.

Antonio Ruano (CTFC) suggests introducing an ontological approach to data exchange in the wood value chain. References on this approach which connects actors along the wood supply chain using semantic interoperability are available on the EU OntoCommons project https://ontocommons.eu/.

Final Remarks

Based on the meeting outcomes, the following actions are noted/planned (urgent actions in **green bold**).

Partner in charge	When	Related WP/Task	Action Required
Acronym	DD/MM/YYYY	Description	Description
PIVETEAU	12/02/2024	WP2, T2.3	Data Structure and variables: Example
MICROTEC	09/02/2024	T2.3 Leader	Data Structure and variables: Example

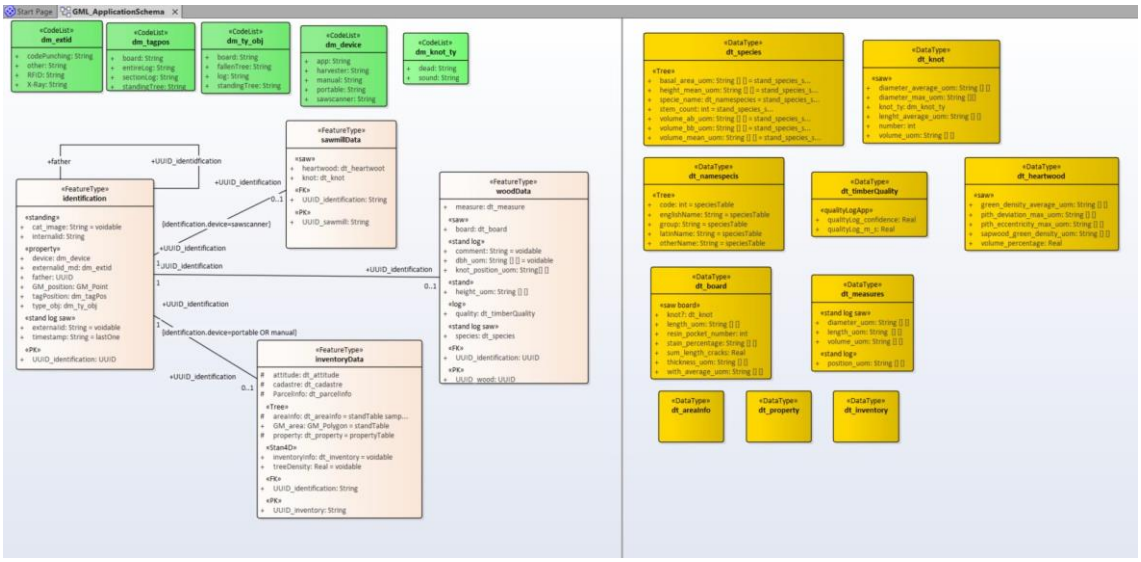
DataDefinition_matchingGeoDB - 20th of February 2024

Scope, location and general information

Zoom meeting planned 20th of February 2024 to verify the matching between data attributes defined in the previous meeting (ref: minutes 07 and 09 February) with TREE and ARBOREAL partners and the updated application schema (task 1.3). Presentation and discussion of the of GML application schema with particular attention to “FeaturesTypes” definition for traceability system.

Meeting minutes

Manuela Corongiu (LAMMA) directed the meeting to update the TREE attribute tables (in TeamSintetic Tree Data definition_machingGeoDB.xlsx) and GML schema in real time, as reported below.



In particular, Manuela explained that traceability is made possible through recursive relationship in the Identification table.

Stefano Romanelli (LaMMA) showed a test using some fake records inserted into the geodatabase to show how it is possible to trace the original tree that gave life to logs and boards.

Lamma also confirmed the whole information Johan Ekenstedt (ARBOREAL) sent have been included into the GML Application Schema.

Concerning the identification and interpretation of OTMETKA's code from the image, Johanproposes RaspAPI or a similar solution for sending the image to their server, allowing them to decode it and send it back (paying attention to an available internet connectivity) for code assignment. This seems to be the simplest option, because it doesn't involve storing the image in the DB but only its decoding.

Alex Poveda (TREE) commented on the fields of the table sent previously comparing them with those inserted in the DB schema.

There was a misunderstanding of the table name "stand" which caused the related fields to be interpreted as attributing them to the standing tree instead of the forest.

Some considerations were made on tree species and their nomenclature. Probably, even if it will be complex, a list of the species that could appear in SINTETIC together with their respective names in the project languages, will have to be made. It will be necessary to define which species we want to be included in the project. Then different formulas for different trees, like volume formulas without bark etc., can be used. At least the species we are going to harvest are the ones needed.

Alex Poveda (TREE) will verify some attributes not yet well defined (Tree Data definition_machingGeoDB.xlsx) and will send to LAMMA as soon as possible to update the Application schema in order to generate and test the corresponding physical GeoDB in the PostgreSQL environment.

Discussion Points:

- The description of Attribute Table (Tree Data definition_machingGeoDB.xlsx) and selection of records matching with GeoDB Application Schema have been reviewed during the meeting: TREE will revise some attributes and will send a final version of the attribute table to update the schema before the reversing in physical table in PostgreSQL.

- Solutions to manage raster files and images (Otmetka and Apps images). Alex Poveda (TREE) asked which DBMS should be used for managing raster images. Leandro Rocchi (CNR), who came later, said that in his opinion it's convenient to use PostgreSQL+Postgis by integrating the images into, because it facilitates geoprocessing operations. But he said other solutions could also be considered, such as to store images in a file system.

- How to manage wood waste: logs in the forest discarded and logs in sawmills for waste or different products.

Some questions came up about wood waste and logs discharged; in particular, if information must be stored in the database (and how), or if traceability with lack of information could be foreseen.

Final Remarks

Based on the meeting outcomes, the following actions are noted/planned (urgent actions in **green bold**).

Partner in charge	When	Related WP/Task	Action Required
Acronym	DD/MM/YYYY	Description	Description
TREE	23/02/2024	WP3, T3.2	**Variables updated**: File xls on Teams
LAMMA	27/02/2024	T1.3 Leader	**Application schema updated**

Annex 2: Tree Data definition_machingGeoDB

Below the matching table between Treemetrics Data Definition and the SinteticGeoDB contents (task 1.3).

Table	property						SINTETIC GeoDB
Name	Type	Object type	Required	Can be acluated?	Units	Description	
id	integer	ID	YES			Automated database ID	
timestamp	dateTime	TIMESTAMP	YES			Time and date when the property is created	identification.timestamp
name	string	VCHAR(255)	YES			Name of the property	
ownership	string	VCHAR(100)	NO			Type of ownership	
owner_name	string	VCHAR(255)	NO			Name of the owner ?	inventoryData.property.dt_property
address	string	VCHAR(100)	NO			Full address of the forest ?	inventoryData.property.dt_property
zip_code	string	VCHAR(10)	NO			Zip code of the forest ?	inventoryData.property.dt_property
country	string	VCHAR(2)	NO			Country where the forest is located	
location	string	POINT	YES			Forest location. POINT object (Lat, Long)	

Table	forest_stand - Forest area unit or stratum- (areaInfo)						forest stand inventory
Name	Type	Data	Required	Can be caclu ated?		Description	
id	integer	ID	YES			Automated database ID	
property	object linked	property_id	YES			"property" object	
timestamp	dateTime	TIMESTAMP	YES			Time and date when the "stand" is created	identification.timestamp
stand_id	string	VCHAR(10)	YES			Stand name. Identifier	
location	string	POINT	YES			Stand location. POINT object (Lat, Long)	identification.GM_position
boundaries	string	POLYGON	YES			Stand boundaries	inventoryData_GM_area
elevation	double	DOUBLE	NO		m	Stand average/relevant elevation above see level	inventoryData.areaInfo.dt_areaInfo
slope	string	DOUBLE	NO		%	Stand average/relevant slope	inventoryData.areaInfo.dt_areaInfo
ruggedness	string	VCHAR(255)	NO			Stand ruggdness decripiton	inventoryData.areaInfo.dt_areaInfo
chm	string	VCHAR(255)	NO			Reference to TIFF file (TIFF can be sotred in file-based DB, Bucets, etc)	inventoryData.areaInfo.dt_areaInfo
accessibility	integer	SMALLINT	NO			Accessibility code	inventoryData.areaInfo.dt_areaInfo
soil_type	integer	SMALLINT	NO			Soil type list??	inventoryData.areaInfo.dt_areaInfo
bearing _capacity	double	DOUBLE	NO		kN/m3	average pressure between the harvesting machines and the soil	inventoryData.areaInfo.dt_areaInfo

quality_classfication	integer	SMALLINT	NO			quality classificaiton (to be defined)	inventoryData.areaInfo.dt_areaInfo
snow_cover_frequency	double	DOUBLE	NO		days	minimum height of snow that hampers harvesting activities	inventoryData.areaInfo.dt_areaInfo
area	double	DOUBLE	NO		hectareas	Stand area	inventoryData.areaInfo.dt_areaInfo
productive_area	double	DOUBLE	NO		hectareas	Planted area (usually adjustment from maps or % of the total area)	inventoryData.areaInfo.dt_areaInfo
basal_area	double	DOUBLE	NO	YES	m2/ha	Stand basal area	inventoryData.areaInfo.dt_areaInfo
stem_count	integer	SMALLINT	NO	YES		Number of stems within the stand	inventoryData.areaInfo.dt_areaInfo
biomass	double	DOUBLE	NO	YES	tons	Total biomass	inventoryData.areaInfo.dt_areaInfo
volume_ab	double	DOUBLE	NO	YES	m3	Total volume above bark within the stand	inventoryData.areaInfo.dt_areaInfo
volume_bb	double	DOUBLE	NO	YES	m3	Total volume below bark within the stand	inventoryData.areaInfo.dt_areaInfo
mean_height	double	DOUBLE	NO	YES	m	Mean tree height in m (e.g. 25.8)	inventoryData.areaInfo.dt_areaInfo
mean_tree_volume_ab	double	DOUBLE	NO	YES	m3	Mean tree volume above bark in m3 (e.g. 0.82)	inventoryData.areaInfo.dt_areaInfo
mean_tree_volume_bb	double	DOUBLE	NO	YES	m3	Mean tree volume below the bark in m3 (e.g. 0.82)	inventoryData.areaInfo.dt_areaInfo
species	string	VCHAR(255)	NO	YES	list	List of species and Number of stems (e.g. AA, SS, LP)	inventoryData.areaInfo.dt_areaInfo
age	double	DOUBLE	NO	Calculated or Provided	years	User provided or the Average age from lower levels -in years- (e.g. 85)	inventoryData.areaInfo.dt_areaInfo

Table	sample_plots (inventoryData)						
Name	**Type**	**Data**	**Required**	**Can be acluated?**		**Description**	**area di saggio (radius of a small area)**
id	integer	ID	YES			Automated database ID	
forest_stand	object linked	forest_stand	YES			forest stand object	
timestamp	dateTime	TIMESTAMP	YES			Time and date when the "stand" is created	identification.timestamp
plot_id	string	VCHAR(10)	YES			Sample plot id (external id)	
location	string	POINT	YES			Plot location. POINT object (Lat, Long)	identification.GM_position
sample_type	string	VCHAR(30)	YES			Choose from: Circular, Rectangular, Transect, Other	
length1 (or radius)	double	DOUBLE	YES			Radius or length of the plot	
length2	double	DOUBLE	YES			Second side length for rectangular plot	
sample_area	double	DOUBLE	YES	Calculated or Provided	m2	Sample area can be calculated from plot radius/length or directly provided	inventoryData.areaInfo.dt_areaInfo
basal_area	double	DOUBLE	NO	YES	m2/ha	Stand basal area	inventoryData.areaInfo.dt_areaInfo
stem_count	integer	SMALLINT	NO	YES		Number of stems within the stand	inventoryData.areaInfo.dt_areaInfo
volume_ab	double	DOUBLE	NO	YES	m3	Total volume above bark within the stand	inventoryData.areaInfo.dt_areaInfo
volume_bb	double	DOUBLE	NO	YES	m3	Total volume below bark within the stand	inventoryData.areaInfo.dt_areaInfo
biomass	double	DOUBLE	NO	YES	tons	Total biomass	inventoryData.areaInfo.dt_areaInfo
mean_height	double	DOUBLE	NO	YES	m	Mean tree height in m (e.g. 25.8)	inventoryData.areaInfo.dt_areaInfo
mean_tree_volume_ab	double	DOUBLE	NO	YES	m3	Mean tree volume above bark in m3 (e.g. 0.82)	inventoryData.areaInfo.dt_areaInfo
mean_tree_volume_bb	double	DOUBLE	NO	YES	m3	Mean tree volume below the bark in m3 (e.g. 0.82)	inventoryData.areaInfo.dt_areaInfo

species	string	VCHAR(255)	NO	YES	list	List of species and Number of stems (e.g. AA, SS, LP)	inventoryData.areaInfo.dt_areaInfo
age	double	DOUBLE	NO	Calculated or Provided	years	User provided or the Average age from lower levels -in years- (e.g. 85)	inventoryData.areaInfo.dt_areaInfo

Table	tree (inventoryData)						
Name	**Type**	**Data**	**Required**		**Units**	**Description**	
id	integer	ID	YES			Automated database ID	
stand_id	object linked	stand_id	YES			forest stand object id	
plot_id	object linked	plot_id	NO			sample plot object id	Note that trees may belong to the plot or directly to the stand for inventory without smapling
tree_id	string	VCHAR(255)	YES			Initial tree ID (from UAV /TLS analysis)	
external_id	string	VCHAR(255)	NO			external tree id	identification.externalid
tag_type	string	VCHAR(255)	NO			e.g. RFID, field paint, etc	
timestamp	dateTime	TIMESTAMP	YES			Time and date when the "tree" is created	identification.timestamp
tree_species	object linked		YES			"specie" object	woodData
location	string	POINT	YES			Tree location. POINT object (Lat, Long)	identification.GM_position.GM_point
biomass	double	DOUBLE	NO		m3	Tree biomass	
age	double	DOUBLE	NO			Tree age	
stem_volme_ab	double	DOUBLE	NO		m3	Stem volume above bark	woodData.measue.dt_measure.volume_uom
stem_volme_bb	double	DOUBLE	NO		m3	Stem volume below bark	woodData.measue.dt_measure.volume_uom
total_height	double	DOUBLE	NO		m	Stem length (tree height)	
quality_type	object linked		NO			quality object type	Allow for multiple quality parameter per tree
quality_value	string	VCHAR(255)	NO			quality value	Allow for multiple quality parameter per tree
intermediate_support	boolean	BOOLEAN	NO			Tree used as intermediate support for the cable	?
anchor	boolean	BOOLEAN	NO			Tree used as anchor for the cable	?

Table	tree_taper (DBH-Height description)						
Name	**Type**	**Data**	**Required**		**Units**	**Description**	
tree_id	object linked	stand_id	YES			tree object id	
diameter_ab			NO			Diameter above bark	woodData.measue.dt_measure.diameter_uom
diameter_bb			NO			Diameter below bark	
Height			YES				woodData.height_uom
bark_thickness	double	DOUBLE	NO	Calculated or Provided	mm	Bark thickness for given diameter	Needed to estimate without bark values

quality_type	object linked		NO			quality object type	Allow for multiple quality parameter per tree section
quality_value	string	VCHAR(255)	NO			quality value	Allow for multiple quality parameter per tree section
Table	quality_indicators						
Name	**Type**	**Data**	**Required**		**Units**	**Description**	
quality_id	integer	ID	YES			Automated database ID	woodData.quality.dt_timberquality
quality_indicator	string	VCHAR(255)	YES			name of quality indicator (leaning, branchiness, lowest,branch, defect, etc)	woodData.quality.dt_timberquality

Table	log_product (log product definition)						
Name	**Type**	**Data**	**Required**		**Units**	**Description**	
id	integer	SMALLINT	YES			Automated database ID	
name	string	VCHAR(255)	YES			Product name reference	
tree_species	object linked	tree_specie	YES			"tree_specie" object	
max_sweep	double	DOUBLE	NO		mm/m	Maximum sweep in the log product	
quality_allowed	integer	SMALLINT	NO		list	Type of defect allowed for his type of product (1,2,3)	
max_length	double	DOUBLE	NO		m	Maximum log length allowed	
min_length	double	DOUBLE	YES		m	Minimum log length allowed	
min_small_end_diamter	double	DOUBLE	YES		mm	Minimum SED allowed	
max_small_end_diamter	double	DOUBLE	NO		mm	Maximum SED allowed	
min_large_end_diamter	double	DOUBLE	NO		mm	Minimum LED allowed	
max_large_end_diameter	double	DOUBLE	NO		mm	MaximumLED allowed	
quality_indicator	object linked	tree_specie	NO			"quality" object	
min_quality_indicator_value	string	VCHAR(255)	NO			minimum quality indicator value	
max_quality_indicator_value	string	VCHAR(255)	NO			Maximum quality indicator value	

Table	product_type_weightings						
Name	**Type**	**Data**	**Required**			**Description**	**related to product**
id	integer	ID	YES			Automated database ID	?
name	string	VCHAR(255)	YES			cutting instruction name	?

Table	cutting_instruction						
Name	**Type**	**Data**	**Required**			**Description**	
id	integer	ID	YES			Automated database ID	?
name	integer	ID	YES			cutting_instruction name	A cutting instrouction contain multiple products weighting
product_type_id	object linked	product_type	YES			"product_type" object	
weighting	double	DOUBLE	NO			product type weting values	?

Table	log						
Name	**Type**	**Data**	**Required**		**Units**	**Description**	
id	integer	SMALLINT	YES			Automated DB ID	
tree_id	integer	SMALLINT	YES			Related to other database objects: tree	
external_id	string	VCHAR(255)	NO			external tree id	identification.externalid
product_type_id	integer	SMALLINT	NO			"product_type" link	?
tag_type	string	VCHAR(255)	NO			e.g. RFID, field paint, etc	
timestamp	dateTime	TIMESTAMP	NO			Timestamp when log is created?	identification.timestamp
cutting_instruction_id	object linked		YES			"cutting_instruction" link	
length	double	DOUBLE	NO		m	Log length	woodData.measure.dt_measure.length_uom
small_end_diameter	double	DOUBLE	NO		mm	Actual SED	
large_end_diameter	double	DOUBLE	NO		mm	Actual LED	
biomass	double	DOUBLE	NO		m3	Log biomass	
volume_ab	double	DOUBLE	NO		m3	Total log volume above bark (Smaliana or similar)	woodData.measure.dt_measure.volume_uom
volume_bb	double	DOUBLE	NO		m3	Total log volume below bark (Smaliana or similar)	woodData.measure.dt_measure.volume_uom
quality_type	object linked		NO			quality object type	Allow for multiple quality parameter per log
quality_value	string	VCHAR(255)	NO			quality value	Allow for multiple quality parameter per log
log_bottom_height	double	DOUBLE	YES		m	Log bottom height in the stem	? Only need if we want to build the tree back from logs
log_top_height	double	DOUBLE	YES		m	Log top height in the stem	? Only need if we want to build the tree back from logs

---

Table	species						
Name	**Type**	**Data**	**Required**	**Can be acluated?**		**Description**	
id	integer	ID	YES			Automated database ID	
species_code	integer	VCHAR(10)	YES			Short name or code of species	

species_group	integer	VCHAR(100)	NO			Spcies grouped by region or user needs (e.g. hardwood, softwood)	
species_english	integer	VCHAR(100)	NO			Species name in English	

Table	species_parameters						
Name	**Type**	**Data**	**Required**	**Can be acluated?**		**Description**	
id	integer		YES			Automated database ID	
species_code	object linked	VCHAR(10)	YES			Automated database ID	
density	double	DOUBLE	NO			Wood density used for quality and biomass calculations	
root_shoot	double	DOUBLE	NO			Root to shoot ratio (used for biomass calculation)	
bm_expansion_factor	double	DOUBLE	NO			Biomass expasion factor (used for biomass calculation)	
volume_equation	string	VCHAR(255)	NO			Taper equation	