SPECIFICATION OF THE AURORAL ONTOLOGY FOR THE RURAL ENVIRONMENT

Date: 20 December 2023
Version : 0.16

D4.1 Specification of the AURORAL ontology for the rural environment

Document Information

Project title	AURORAL
Grant Agreement	101016854 — AURORAL
Project Coordinator	Rui Barroso
Deliverable number and title	D.4.1. Specification of the AURORAL ontology for the rural environment
Deliverable type	Report
Dissemination level	Public
Lead beneficiary	UPM
Due submission date	M36 – 31 Dec 2023
Submission date	31/12/2023
Doc. Version	1.0
Doc. Reference	AURORAL_D4.1

Keywords

ontology specification, ontology engineering, semantic model, ontology governance

Disclaimer

Co-funded by the Horizon 2020 programme
of the European Union

This report is part of the AURORAL project that has received funding by the European Union's Horizon 2020 Programme under Grant Agreement number 101016854.

Document Information

Document Author(s) and Reviewer(s):

All Approvers are required. Reviewers are required unless explicitly listed as Optional.

Name	Entity	Role *(Author / Contributor / Reviewer)*
Raúl García-Castro	UPM	Author
Juan Cano	UPM	Author
María Poveda-Villalón	UPM	Author
Ahlem Rhayem	UPM	Author
Andrea Cimmino	UPM	Author
Jorge Almela	BVR	Contributor
Carmina Bocanegra-Yanez Jose Diaz de Greñu Alberto Diez Frias	BOSONIT/ELLIOT CLOUD	Reviewer
Ana valido Raquel Cetra Patricia Silva	CCDR-A	Reviewer

Document history

The Document Author is authorized to make the following types of changes to the document without requiring that the document be re-approved:

- Editorial, formatting, and spelling
- Clarification

To request a change to this document, contact the Document Author or Project Coordinator.

Changes to this document are summarized in the following table in reverse chronological order.

Revision N.	Date	Made by	Short Description of Changes
0.1	17/09/21	Raúl García-Castro	First draft of the document
0.2	27/09/21	Juan Cano	Added first content
0.3	07/10/21	Raúl García-Castro	Updates on the content
0.4	13/10/21	Raúl García-Castro	Draft for technical check
0.5	20/06/22	Raúl García-Castro, Juan Cano, Ahlem Rhayem	Draft for review
0.6	29/06/22	Ahlem Rhayem, Raúl García Castro	Updated version
0.7	10/12/22	Ahlem Rhayem	Update content
0.8	15/12/22	María Poveda-Villalón	Reviewing the deliverable
0.9	21/12/22	Ahlem Rhayem	Update the deliverable
0.10	23/12/22	Maria Poveda	Second reviewing
0.11	25/12/22	Ahlem Rhayem	Updated deliverable

0.12	01/12/2023	Ahlem Rhayem	Update content
0.13	30/11/2013	Raúl García-Castro, Juan Cano, Ahlem Rhayem, Andrea Cimmino, María Poveda-Villalón	Reviewing the deliverable
0.14	11/12/2023	Ahlem Rhayem	Updated version
0.15	13/12/2023	María Poveda-Villalón	Review and updated content
0.16	19/12/2023	Ahlem Rhayem	Review and update content
1.0	20/12/2023	María Poveda-Villalón	First version

Document Location

The latest version of this controlled document is stored in https://cloud.auroral.eu/index.php/s/J5fiB93HpTktzkD

Table of Contents

List of Tables

List of Figures

Abbreviations

DAMA	Data Management
DCAT	Data Catalog Vocabulary
FOAF	Friend Of A Friend Ontology
GOMO	Governance Operational Model for Ontologies
Geo	Geospatial Ontology
HCTL	Hypermedia Controls Ontology
IOF	Industrial Ontologies Foundry
JSON-LD	JavaScript Object Notation for Linked Data
LOT	Linked Open Terms
ODRL	Open Digital Rights Language
OM	Ontology of units of Measure
OOPS!	OntOlogy Pitfall Scanner!
Org	Organisation Ontology
ORSD	Ontology Requirements Specification Document
OWL	Ontology Web Language
OBO	Open Biological and Biomedical Ontology
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
SAREF	Smart Appliance Reference Ontology
SHACL	Shapes Constraint Language
TD	Thing Description
tmjourney	Transmodel Journey
TTL	Turtle format
URI	Uniform Resource Identifier
WGS84	World Geodetic System 1984
WIDOCO	Wizard for documenting ontologies
WoT	Web of Things
W3C	World Wide Web Consortium

Executive Summary

This document presents the AURORAL ontology network for the rural environment status at the end of the related task T4.1 entitled *"Adaptation of existing semantic data models for the rural environment*". This task aims at supporting the semantic interoperability within AURORAL platform through common information models defined by ontologies implemented in standard languages like Ontology Web Language (OWL). In addition, this document includes a summary of the methodology for building ontologies followed and the ontology governance model.

Since there is a large number of domains and use cases to be considered in the conceptualization of AURORAL, the developed ontology is organized as a modular ontology network. In total, thirteen modules were developed for various use cases.

The structure of this deliverable is as follows:

- A description of the used ontology development methodology entitled Linked Open Terms (LOT) methodology.
- A presentation of the ontology development infrastructure deployed in AURORAL.
- A description of an ontology governance process to determine the main principles and best practices followed during the development of the ontology.
- Finally, each ontology module is described in detail, indicating its requirements and the description of the current model.

The latest version of the AURORAL ontology development network and all development artifacts can be found on the AURORAL ontology portal (https://auroral.iot.linkeddata.es).

1. Introduction

The Semantic interoperability within the AURORAL project is achieved through common ontologies to facilitate the exploitation and the transparency use of heterogeneous data collected from diverse domains. The goal of this document is to present the AURORAL ontology network that was used throughout the different components of AURORAL and by the different actors that will consume the information. This task is considered essential for the other work packages, WP5 and WP6, where the AURORAL ontology network serves as a shared vocabulary and model covering the data value chain required by the project.

As there is a broad number of use cases to be considered in the AURORAL conceptualization, thirteen modules were developed for diverse use cases categorized into six different domains. These domains are tourism, health, energy, mobility, dairy farming, and circular economy.

During the development of the AURORAL ontology network, existing and standard ontologies were considered, such as the ETSI Smart Applications REFerence (SAREF) ontology, The World Wide Web Consortium (W3C) Web of Things ontology, the time ontology, the unit of measure ontology, etc.

AURORAL ontologies are formalized following Description Logics and implemented in the W3C Web Ontology Language standard OWL.

This deliverable starts by describing the following methodology for the development of the AURORAL ontologies models, which is the Linked Open Terms (LOT) methodology. After that, an ontology governance process is suggested and detailed. The main goal of this section is to ensure a well-established ontology following a set of principles and best practices. After that, we will describe the ontology development infrastructure used. Finally, the different ontology modules are described before the document is wrapped with the conclusions section.

The main novelties of this document regarding previous versions are:

- A new section on the governance of the AURORAL ontology has been added.
- Certain sections on the ontology's modules have been updated, namely: the core ontology, tourism, biomass, logistics and energy.
- New sections on new modules have been added such as: CarBooking, farming, mobility, cell-tower and electrical vehicle charger.

The structure of this deliverable is as follows:

- Section 2 explains the methodology used to develop the AURORAL ontologies, namely the LOT methodology.
- Section 3 describes the AURORAL ontology governance model.
- Section 4 describes the ontology development infrastructure.
- Section 5 presents an overview of the AURORAL ontology network.
- Sections 6 to 18 present the thirteen ontology modules.
- Section 19 concludes the document.

2. Ontology development methodology

This section presents the ontology development methodology used for the development of the AURORAL ontology network. This methodology includes four activities: 1) ontology requirements specification, 2) ontology implementation, 3) ontology publication, and 4) ontology maintenance. This development methodology is named Linked Open Terms (LOT) [1] and is based on the NeOn methodology [2]. It has been used and refined in previous ontology development processes of the European projects VICINITY [3], BIMMER [4], DELTA [5], COGITO [6], easyTV [7] among others, and has been adapted to the particularities of AURORAL. For this reason, it is important to mention that some parts of this section might be similar to other EU projects deliverables as they describe and adapt a common methodology.

Figure 1 shows an overview of the phases that are performed and the artefacts that result from them: the ontology requirements specification document (ORSD), the ontology implementation, the ontology available online, issues, bugs, etc. It should be mentioned that each phase is divided in a set of detailed activities that have been described in previous version of this deliverable.

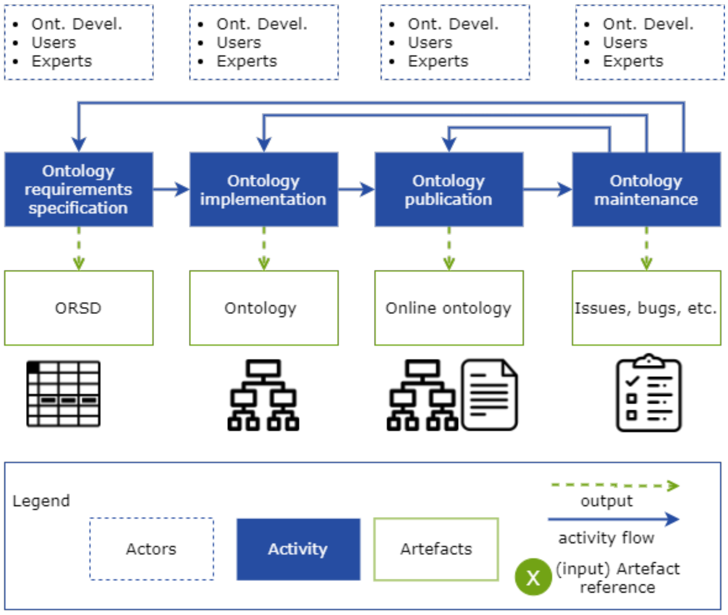


Figure 1 - LOT: Ontology development methodology followed in AURORAL [1]

The aim of the **requirements specification phase** is to identify and define the requirements that the ontology to be created needs to fulfil. During this first activity, it is necessary to involve experts in the domain to generate the appropriate industry perspective and knowledge. The main following steps are: the definition of the main purpose and scope of the ontology, the data collection and use case specification, and the ontology requirements identification and validation.

During the **ontology implementation phase**, the ontology is built using a formal language (OWL) based on the requirements identified in the previous activity.

Taking as input the set of requirements collected in the previous activity, the ontology implementation activity is carried out through several sprints. To this end, the ontology developers schedule the ontology development according to the requirements that were identified, and the ontology development team builds the ontology iteratively by implementing only a certain number of requirements in each iteration. The output of each iteration is a new version of the ontology implementation.

The main steps of the ontology implementation are as follow:

- The conceptualization step that aims to create a semantic model from the identified requirements.
- The encoding step interests in the implementation of the ontology in an implementation language such as OWL.
- The evaluation step aims to evaluate the ontology before its online publication. The development of the ontology must guarantee that the developed ontology is consistent, does not have syntactic, modelling, or semantic errors, fulfils the requirements scheduled for the ontology.

During the **ontology publication** phase, the ontology development team provides an online ontology that is accessible both as a human-readable document and as a machine-readable file from its Uniform Resource Identifier (URI).

During the **ontology maintenance** phase, the ontology is updated, and new requirements can be proposed to be added to the ontology. Furthermore, during this activity, the ontology development team, together with domain experts and users, can identify and correct errors in the ontology. The main steps of this phase are the correction of a detected bug or update of the ontology when a new requirement or a modification were proposed.

3. Ontology governance

While in it important to have a well stablished methodology for building ontologies, in many organizations or projects involving many domains and diverse use cases, it is also of paramount importance to define a governance model to ensure common practices, principles and rules among ontology stakeholders. This is the case of the AURORAL project in which many domains are involved in the different use cases and the ontologies should be use used and may be extended by open call participants. The term governance was firstly introduced for data-oriented projects as: *the exercise of authority and control (planning, monitoring and execution) over the management of data assets* [8]. The purpose of data governance is to ensure that the data is managed correctly, in accordance with policies and best practices. Another well-known framework for data governance is DAMA [9]. According to this framework data governance covers a set of principles and best practices related to diverse activities such as: data architecture, data modelling, data storage, data security, data integration and interoperability, data quality, metadata, data warehousing, reference and master data, and documents and content. Based on these definitions we could say that ontology governance aims to manage the various components involved in the ontology development process through a set of principles and best practices.

Regarding governance models for ontologies there exist different models such as GOMO, OBO, and IOF. A description of these models is given below:

GOMO: BASF company has proposed a Governance Operational Model for Ontologies called “GOMO” [10]. This model is not publicly available. The main components of the GOMO model are: the principles, standards, best practices, training and outreach. Details about this model were not provided publicly.

- *Principles*: Set of high-level fundamental rules on how the organization should develop, publish, maintain, and consume ontologies. They provide a general framework, a basis for the other elements (in GOMO the standards, best practices, etc.).
- *Standards*: Set of specifications that are to be followed when developing, publishing and maintaining ontologies. Each one is supplied with a Quality and Assurance criteria to allow human or software-based evaluation of whether that standard has been implemented correctly. Examples: Naming conventions, what are the formats that must be used in the ontology, how MUST/SHOULD the IRIs be structured, all the ontologies must include certain human-readable documentation components, etc.
- *Best practices*: Set of recommendations and guidelines that explain and illustrate how to follow the principles and implement the standards during the ontology development process.

OBO: A list of 14 principles has been defined [15] for evaluating Open Biological and Biomedical Ontologies (OBO). For each principle, an explanation of how it has been verified has been provided. Here are some examples of these principles: The ontology must be openly available, the ontology is available in a common formal language, the ontology has textual definitions for its classes and properties, among others.

IOF: To build the Industry Ontology Foundry (IOF) [16], a set of 15 principles was defined during the development of this ontology. Among these principles, the ontology's architecture must follow a predefined structure that defines the general/common ontology, the domain ontology, the application ontology, etc. The ontology must be well documented, the ontology must follow predefined naming conventions, etc.

After reviewing and analysing existing governance models for ontologies available (note that for GOMO only main concepts are public) and considering AURORAL needs, the governance model described in the following sections has been defined.

3.1. Proposed Governance model

In this section, we propose a model for the AURORAL ontology governance. Our model is composed of a set of **roles**, **principles** and **best practices** to ensure good management of the ontology development process. It should be noted that the proposed governance model is aligned to the previously defined methodology for building the ontologies. In the next sections we first introduce the defined principles, then the best practices defined for each principle and finally we introduce the roles considered during the governance model.

3.1.1. Principles

By analysing existing governance models (GOMO, OBO, IOF) for building ontologies the following 15 principles have been defined for the AURORAL governance model:

1. **Scope and context**: The ontology scope and context should be clearly defined before the ontology development process, that is, each ontology should be based on a well-stablished use case.
2. **Textual definitions**: The ontology must provide textual definition of the requirements and all ontology elements (concepts, properties, individuals, etc.).
3. **Ontology architecture**: The ontology architecture should be specified, including the level of generalisation of the ontology to be developed: top-level ontologies, general or cross-domain ontologies and domain ontologies.
4. **Common format**: The ontology should be encoded using standard languages such as OWL or RDF(S).
5. **Modularity**: The ontologies should be developed following a modular approach in accordance with the domains and scopes addressed.
6. **Naming conventions**: The ontology should follow the defined guidelines for ontology elements (concepts, properties, individuals, etc).
7. **Knowledge reuse**: The ontology should reuse existing ontologies, standards or knowledge sources whenever available.
8. **Ontology quality**: The ontology should meet quality criteria like coverage based on requirements, use of correct modelling decisions and lack of errors (semantic, syntactic, coherence, consistency).
9. **URI and namespace**: The ontology modules should have a unique and persistent URI.
10. **Availability**: The ontology should be openly available on the web in human and machine-readable formats.
11. **Maintenance**: The ontology development team should stablish and follow ontology maintenance processes and activities.
12. **Versioning**: A version strategy should be defined for the ontology versions and releases.
13. **Roles**: The different roles involved in the ontology lifecycle should be identified and assigned to corresponding persons or stakeholders.
14. **Communication**: Communication channels and strategies should be defined within the ontology stakeholders as well as external communities.

15. **Documentation**: Human oriented documentation should be provided through the ontology lifecycle, including requirements, scope and context, modelling description, examples, etc.

As mentioned, even though governance model principles are independent of the ontology development methodology, for the AURORAL case we have mapped the selected principles to the LOT methodology phases and activities. Such mappings are shown in Figure 2.

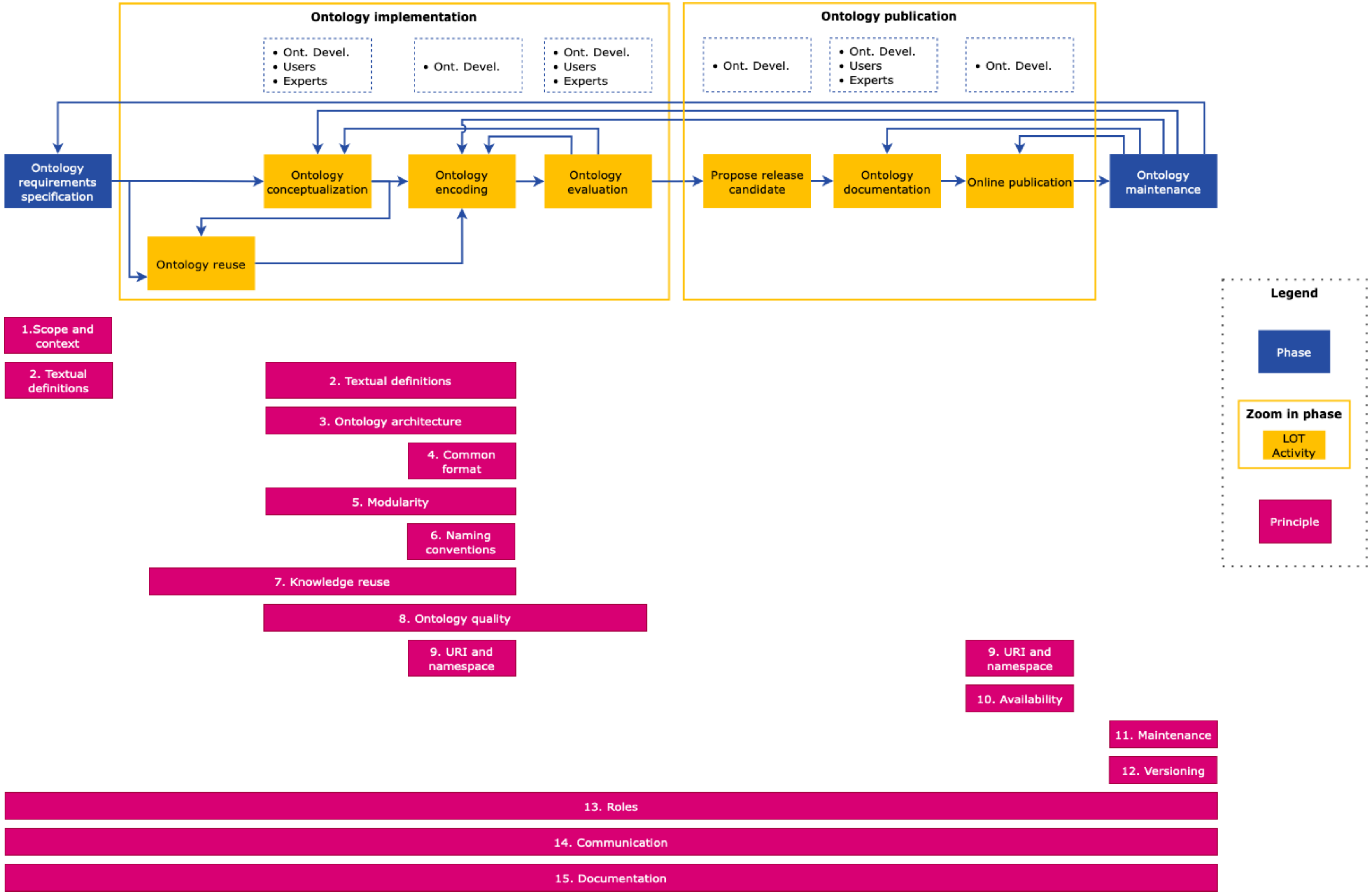


Figure 2 - Auroral governance model mappings to LOT methodology

3.1.2. Best Practices

For each principle, a set of best practices were defined. These best practices are detailed below:

1. **Scope and context**:
    - Stablish meetings between the ontology development team and use case owners to discuss and clearly define the scope and context.
2. **Textual definitions**:
    - Use a competency question or a natural language sentence to define the requirements.
    - Add human oriented identifiers (rdfs:label) and definitions (rdfs:comment) for all ontology elements.
3. **Ontology architecture**:
    - Analyse the ontology scope and requirements to identify the architecture layer in which the ontology to be developed fit.
    - Analyse whether existing ontologies should be extended or generalized according to the needs.
4. **Common format:**
    - Use OWL or RDF(S) languages encoded in several formats like turtle (ttl), n-triples, JavaScript Object Notation for Linked Data (json-ld) or RDF/XML.
5. **Modularity**:
    - Develop ontology modules according to the layer of each ontology in the architecture.
    - Define ontology modules according with the topic of concern maximizing cohesion and minimizing coupling.
6. **Naming conventions**:
    - Use CamelCase notation starting classes with capital letter and properties and individuals with lower case for the first word.
7. **Knowledge reuse**:
    - Look for existing ontologies or knowledge sources.
    - Reuse existing ontologies by owl:imports (hard reuse) or by referencing the ontology elements URIs (soft reuse).
    - When the reuse is not explicit by URIs, acknowledge the source by means of annotations.
8. **Ontology quality**:
    - Use ontology reasoners and validators.
    - Review modelling decisions with domain experts and ontology users.
    - Ensure the requirements coverage.
9. **URI and identifier space**:
    - Design ontology name and prefix for each ontology.
    - Choose between the slash and the hash in the prefix.
    - Use of permanent URIs.
10. **Availability**:
    - Publish the ontology under its URI using content negotiation.
    - Associate an open license to the ontology.
    - Publish an ontology portal with links to all ontology's resources.
11. **Maintenance**:
    - Set up an issue tracker for each ontology.

- Monitor the issues and suggestions done for the ontologies and stablish a procedure to answer and address the requests.

12. **Versioning**:
    - Stablish a versioning strategy defining which type of changes trigger a new version or minor revisions.
    - Keep updated and compile all resources related to a given version.
13. **Roles**:
    - Identify the roles involved in the ontology lifecycle and assign them to specific persons or teams.
14. **Communication**:
    - Define the communication channels to be use for each situation, such as email, instant messaging and teleconference systems (Zoom, google meet), or issue trackers, to facilitate communication with relevant stakeholders.
    - The ontology development team should manage meetings relating to the ontology engineering process.
    - The ontology users, usually software developers, should manage meetings relating to the use of the ontology.
15. **Documentation**:
    - Add detailed and updated metadata annotations to the ontology files.
    - Add detailed and updated metadata annotations to the ontology elements.
    - Generate human readable documentation in HTML format.
    - Generate payloads examples.

3.1.3. Roles

During the existing governance models analysis 8 roles were identified for the AURORAL governance model. Table 1 describes these roles, along with the roles and tasks associated with them and the corresponding AURORAL project partners associated to each role.

Table 1 - Ontology development roles for AURORAL model

Roles	Tasks	Actors
Ontology Project leader	• Identify the actors responsible for the ontology development lifecycle. • Define the specific tasks assigned to each actor involved in the ontology development process. • Decide on the appropriate standardization and methodology to guide ontology development. • Propose potential updates and corrections to improve ontology quality and accuracy.	Raúl García Castro (UPM)
Ontology curator	• Collaborate with ontology developers to review and evaluate the ontology. • Propose recommendations for updates and corrections to refine the ontology model during the validation.	María Poveda Villalón (UPM)
Ontology developer	• Follow the methodology to develop the ontology. • Verify and validate the model in collaboration with the ontology curator. • Update the model based on feedback from the decision-maker, ontology curator and domain experts, • Generate documentation for the ontology developed.	Ahlem Rhayem (UPM)

	• Publish the ontology online.	
Use case expert	• Gather data about use cases and send it to the decision maker, the manager and the ontology developers. • Formatting the data in an understandable format and language	Pilots
Domain experts	• Participate in meetings with the ontology developers to discuss the data and explain the domain terminology and the use case. • Validate the model if it covers the domain knowledge or not. • Suggest updates for the model.	Experts from the pilots
Software developers	• Participate in meetings with the ontology developers to discuss the use of the ontology in the software and explain the needs from the software regarding the ontologies.	Andrea Cimmino (UPM), Juan Cano (UPM), Jorge Almela (BVR)
Contributor	• Suggest updates and contributions for the ontology.	David Buján Carballal Others
Users	• Use ontology and possible suggest updates, new needs and improvements.	Pilots, Open Calls

4. Ontology development infrastructure

This section describes the ontology development infrastructure used to support the activities in the ontology development process described above.

4.1. Infrastructure to Support the Requirements Specification Activity

The ontology development team used spreadsheets to store the requirements per use case. These requirements were reorganized per domain and converted into an HTML file and uploaded to the AURORAL ontology portal with the most relevant information for users to facilitate the visualization. Figure 3 shows an excerpt of the requirements collected for the AURORAL ontology network and Figure 4 shows an excerpt of the HTML documentation of the AURORAL requirements.

Identifier (domain+id)	Responsible partner	Competency Question / Natural language sentence (fact)
tour1	UPM	What are the attributes of an activity?
tour2	UPM	An activity have difficulty
tour3	UPM	There are five types of activity difficulty: Very easy, easy, medium, hard, v
tour4	UPM	An activity have a Channel
tour5	UPM	What are the attributes of a channel?
tour6	UPM	An activity have a price
tour7	UPM	What are the attributes of a price?
tour8	UPM	An activity can be an event
tour9	UPM	What are the attributes of an event?
tour10	UPM	An activity can be a tour
tour11	UPM	An activity have a language
tour12	UPM	An activity have a location
tour13	UPM	What are the attributes of a location (point)?

Figure 3 - Excerpt of AURORAL ontology requirements

AURORAL

Here you can find the list of the requirements identified for the AURORAL ontology and their main features.

Identifier	Partner	Competency Question / Natural language sentence (fact)	Answer	Status	Superseded by	Priority
tour-1	UPM	What are the attributes of an activity?	Identifier, hidden description, long description, short description, isBookable, externalBookingURL			
tour-2	UPM	An activity have difficulty				
tour-3	UPM	There are five types of activity difficulty: Very easy, easy, medium, hard, very hard				
tour-4	UPM	An activity have a Channel				
tour-5	UPM	What are the attributes of a channel?	channelName			

Figure 4 – Excerpt of Auroral ontology requirements in HTML format

4.2. Infrastructure to Support the Ontology Implementation Activity

To support the implementation activity, the ontology development team uses several tools to edit, store, and evaluate the ontology.

- For the ontology edition, the ontology development team uses Chowlk [26] and Protégé [16] which allows the creation, visualisation, and manipulation of ontologies.
- For ontology storage, the ontology development team uses GitHub [30]. A GitHub repository is created for each ontology in the AURORAL ontology network. Each repository includes:
  - A folder with the implementation of the ontology.
  - A folder with the ontology modelling diagrams.
  - A folder with the documentation of the ontology.
  - A folder with the requirements and tests of the ontology.

The development team uses the OnToology [11, 38] tool to trigger continuous integration activities like ontology documentation generation using Widoco [12] and evaluation using OOPS! [13, 40] and THEMIS [41].

4.3. Infrastructure to Support the Ontology Publication Activity

To support the publication activity, the ontology development team creates an ontology portal online to make the ontology and all the associated information (repository, requirements, tests, releases, etc.) available to users. This ontology portal has different sections:

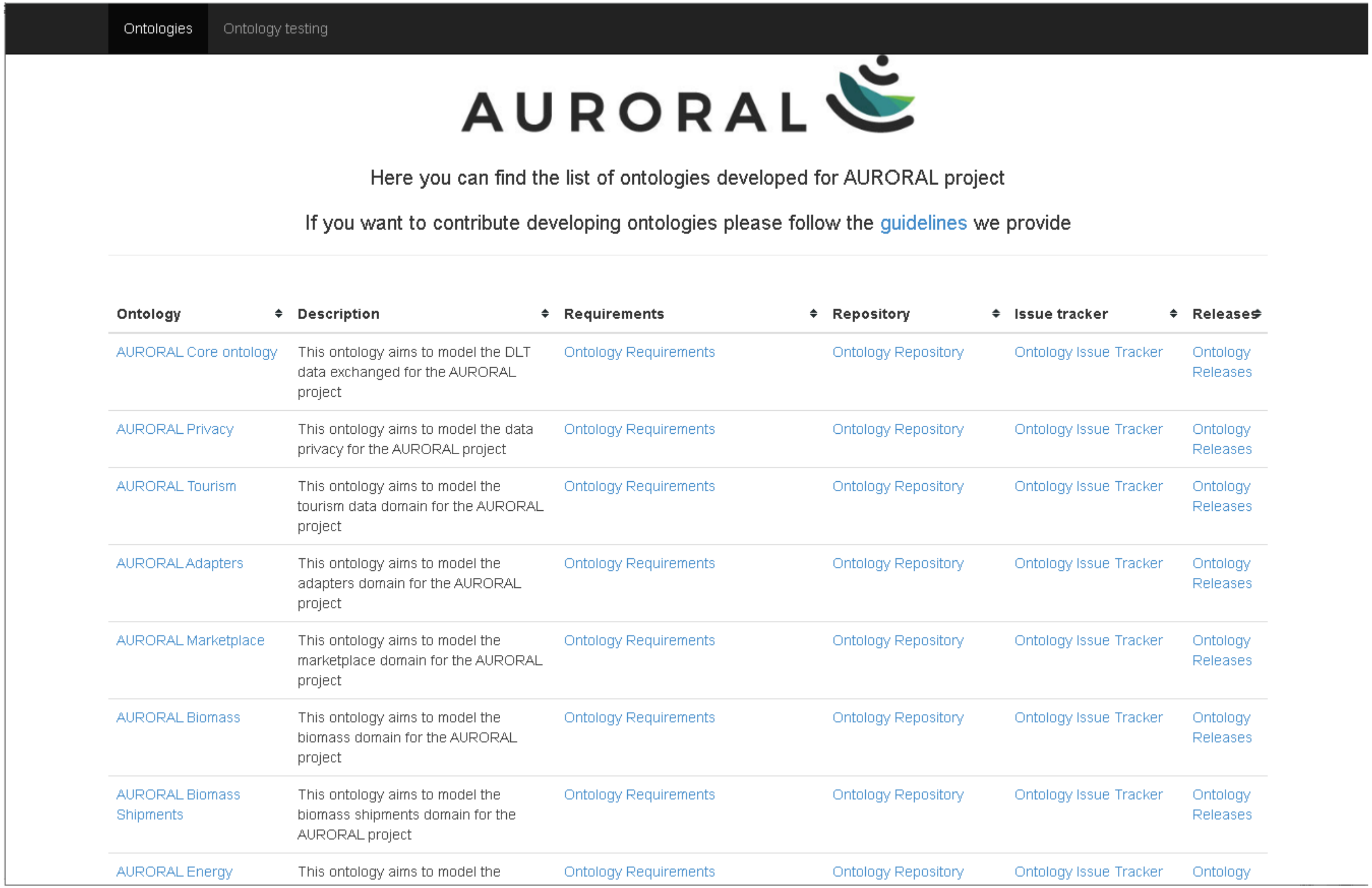
- Ontologies
- Ontologies testing

4.3.1. Ontologies

The Ontologies section, which is the main section of the portal, shows the main information about the ontologies created in the AURORAL ontology network. The section follows a tabular approach that includes the following:

- Link to the ontology documentation published on the Web.
- Ontology Description
- HTML description of the requirements identified by the domain experts.
- Link to each GitHub repository
- Links to each GitHub issue tracker
- Link to ontology releases page
- Link to a list of payloads involved in the data exchange for AURORAL project.

Figure 5 shows an overview of the information exposed in the Auroral ontology portal. Figure 6 lists the payloads defined for each ontology. It contains examples defined in different formats (Turtle, Json-LD 1.1) for each ontology. It also contains the ontology context, which corresponds to the ontology representation in json-LD format, and the Shapes Constraint Language (SHACL) [43] for ontology validation.



Ontologies | Ontology testing

AURORAL

Here you can find the list of ontologies developed for AURORAL project

If you want to contribute developing ontologies please follow the guidelines we provide

Ontology	Description	Requirements	Repository	Issue tracker	Releases
AURORAL Core ontology	This ontology aims to model the DLT data exchanged for the AURORAL project	Ontology Requirements	Ontology Repository	Ontology Issue Tracker	Ontology Releases
AURORAL Privacy	This ontology aims to model the data privacy for the AURORAL project	Ontology Requirements	Ontology Repository	Ontology Issue Tracker	Ontology Releases
AURORAL Tourism	This ontology aims to model the tourism data domain for the AURORAL project	Ontology Requirements	Ontology Repository	Ontology Issue Tracker	Ontology Releases
AURORAL Adapters	This ontology aims to model the adapters domain for the AURORAL project	Ontology Requirements	Ontology Repository	Ontology Issue Tracker	Ontology Releases
AURORAL Marketplace	This ontology aims to model the marketplace domain for the AURORAL project	Ontology Requirements	Ontology Repository	Ontology Issue Tracker	Ontology Releases
AURORAL Biomass	This ontology aims to model the biomass domain for the AURORAL project	Ontology Requirements	Ontology Repository	Ontology Issue Tracker	Ontology Releases
AURORAL Biomass Shipments	This ontology aims to model the biomass shipments domain for the AURORAL project	Ontology Requirements	Ontology Repository	Ontology Issue Tracker	Ontology Releases
AURORAL Energy	This ontology aims to model the	Ontology Requirements	Ontology Repository	Ontology Issue Tracker	Ontology

Figure 5 - AURORAL ontology network portal

AURORAL

Here you can find the list of interfaces involved in the data exchange for AURORAL project

Ontology	Exchanged Data	SHACL shapes	RDF examples (Turtle)	JSON-LD 1.1 examples	Context
AURORAL Core ontology	Data related to provide information about the auroral service.	Core Shape	Service in RDF	Service in Json-LD	Core context
AURORAL Privacy ontology	Data related to provide information about the privacy.	Privacy Shape	Privacy in RDF	Privacy in Json-LD	Privacy context
AURORAL Adapters	Data related to provide information about the Device and its measurement.	Adapters Shape	Adapters device in RDF	Adapters device in Json-LD	adapters context
AURORAL Tourism	Data related to provide information about the tourism activity.	Activity Shape	Tourism activity in RDF	Tourism Activity in Json-LD	Tourism context
AURORAL Marketplace ontology	Data related to provide information about the biomass marketplace.	marketplace Shape	Marketplace in RDF	Marketplace in Json-LD	Marketplace context
AURORAL Biomass	Data related to provide information about Biomass characteristics.	Biomass Shape	Biomass in RDF	Biomass in Json-LD	Biomass context
AURORAL Logistic ontology	Data related to provide information about logistic data.	Logistic Shape	RDF Example	Logistic in Json-LD	Logistic context
AURORAL Energy ontology	Data related to provide information about building energy consumption.	Energy Shape	Energy in RDF	Energy in Json-LD	Energy context
AURORAL car-booking	Data related to provide information about the car booking.	Car-booking Shape	Car-booking in RDF	Car-booking in Json-LD	Car-booking context

Figure 6 - AURORAL Ontology Payloads Portal

4.3.2. Ontologies Testing

The section on Ontologies testing was based on different proposed tools such as the Ontology Pitfall Scanner! (OOPS!) [13, 40] and THEMIS [41].

OOPS! allows the detection of potential pitfalls that could lead to modelling errors. These pitfalls are classified into three distinct categories: minor pitfalls, important pitfalls, and critical pitfalls. Minor pitfalls are issues that do not pose serious problems. Important pitfalls indicate that, even if they are not critical, it is crucial to deal with them. On the other hand, critical pitfalls are likely to affect the consistency of the ontology and therefore it is imperative to address them. Figure 7 presents the given results of OOPS! for the tourism ontology.

Results for P04: Creating unconnected ontology elements.	3 cases	Minor
Results for P10: Missing disjointness.	Ontology*	Important
Results for P11: Missing domain or range in properties.	12 cases	Important
Results for P13: Inverse relationships not explicitly declared.	14 cases	Minor

Figure 7 - Example of OOPS! results

THEMIS allows the verification of the ontology regarding their functional requirements. Figure 8 presents a screenshot of the tourism ontology validation using Themis.

Tests results

Test	Result	Problem
Activity mapPoint Image	Passed	None
Hard type Difficulty	Passed	None
Neutral type Emotion	Passed	None
VeryHard type Difficulty	Passed	None
Disagree type Emotion	Passed	None
Agree type Emotion	Passed	None
Emotion isAbout Activity	Passed	None
Activity postedToChannel Channel	Passed	None
User hasEmotion Emotion	Passed	None
VeryEasy type Difficulty	Passed	None

Figure 8 – Example of Themis results

4.4. Infrastructure to Support the Ontology Maintenance Activity

To provide support for maintenance activity, ontology developers use the GitHub issue tracker, which manages and maintains the list of issues identified by domain experts and ontology developers. The GitHub issue tracker provides the status of the issue, assignee, and description and allows one to add comments to the issue to discuss it. Each issue tracker is associated with a GitHub repository and, consequently, with an ontology in the AURORAL ontology network.

To manage changes in the ontology, all new proposals and improvements must be agreed upon by all members of the ontology development team. If domain experts, users, or ontology developers wish to add, delete, or modify concepts in the ontology, they must create a new issue in the GitHub issue tracker associated with the ontology to be modified, which will be used to discuss the approval or rejection of the proposal.

As a result, 57 issues were created in the issue tracker, with the core ontology and adapters having the highest number of issues, with 19 questions for the core ontology and 29 for the adapters ontology.

Figure 9 shows the GitHub issue tracker for the tourism ontology.

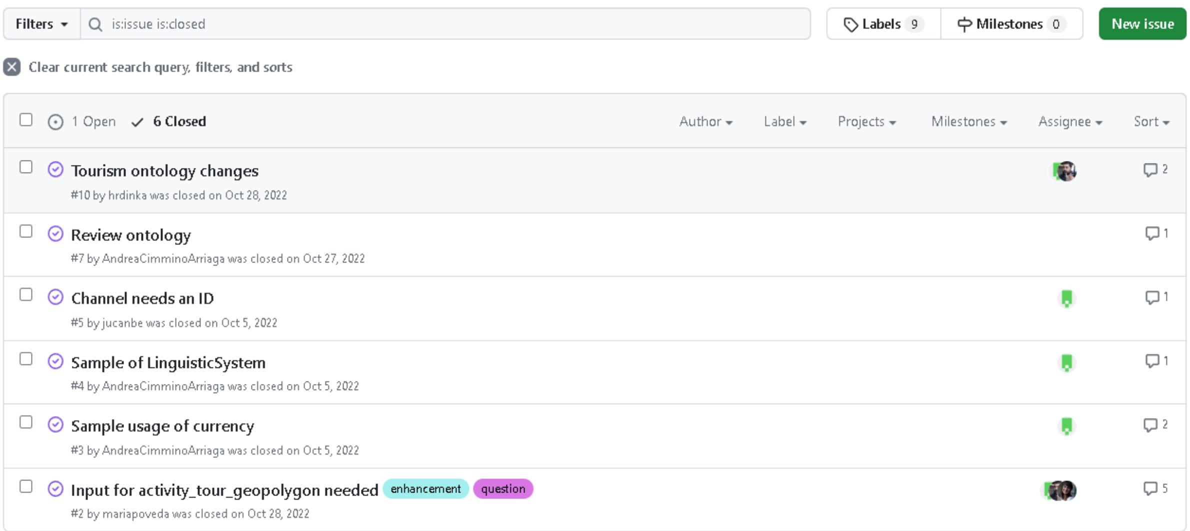


Figure 9 - Example of the GitHub issue tracker

5. Overview of the AURORAL ontology network

The goal of Task T4.1 is to develop the AURORAL ontology network by adapting existing semantic data models and ontologies for the rural environment. This ontology will be used throughout the different components of the AURORAL environment and the different actors that will consume and produce data.

The AURORAL ontology network will be based on existing data models and ontologies. Therefore, before the AURORAL ontology network is developed, a review of existing data models and ontologies to be reused has been performed. The following list enumerates the data models and ontologies considered relevant for the initial version of the AURORAL ontology network:

- **W3C Thing Description (TD) Ontology** [21]. It represents the TD information model, which is one of the building blocks of the Web of Things (WoT). The TD ontology can also be used to process contextual information on Things and for alignments with other WoT-related ontologies.
- **W3C The Organization Ontology** [29]. It is designed to allow the publication of information on organizations and organizational structures, including government organizations. The ontology includes: (1) organizational structure, (2) reporting structure, (3) location information, and (4) organizational history.
- **W3C ODRL Information Model 2.2** [30]. The Open Digital Rights Language (ODRL) is a policy expression language that provides a flexible and interoperable information model, vocabulary, and encoding mechanisms to represent statements about the usage of content and services. The ODRL Information Model describes the underlying concepts, entities, and relationships that form the foundational basis for the semantics of the ODRL policies.
- **Ontology of units of Measure (OM)** [22]. This ontology defines concepts, objects, properties and instances that represent the different measures and units. It includes common units such as the meter and kilogram, as well as other units such as the mile or nautical mile, etc.
- **Smart Appliances REFerence Ontology (SAREF)** [19]. It is a shared consensus model that facilitates the mapping of existing assets in the smart appliance domain. The main concepts of this ontology are: saref:Device, saref:Measurement, saref:Property, saref:service, among others.
- **Geospatial Ontology (Geo)** [20]. It is a lightweight vocabulary that represents geospatial concepts and their properties.
- **Friend Of A Friend Ontology (FOAF)** [31]. It is designed to present people and the relationships between them.
- **Lexvo** [32]. It is designed to provide information about languages, words, characters, and other human language-related entities.
- **Hypermedia Controls Ontology (Hctl)** [33]: It is designed for links and forms. It offers, among others, a means to reify RDF statements interpreted as links between Web resources. It also provides a versatile exchange format for links and forms in RESTful Web applications.
- **Time ontology** [18]: provides a vocabulary for expressing time related concepts such as instants, date descriptions and intervals and their relations.
- **Geonames ontology** [34]: It aims to present the geospatial information.
- **Tmjourney ontology** [35]: describes information about mobility of the mean of transport.

- **dcat ontology** [36]: it is a vocabulary for publishing data catalogues on the Web, which was originally developed in the context of government data.

Thirteen ontologies have been developed and published in the AURORAL project. These include the AURORAL Core ontology, AURORAL Privacy ontology, AURORAL Tourism ontology, AURORAL Adapters ontology, AURORAL Biomass ontology, AURORAL Logistic ontology, AURORAL Marketplace ontology, AURORAL Energy ontology, AURORAL Car-booking ontology, AURORAL Mobility ontology, AURORAL Farming ontology, AURORAL Cell-tower ontology, and AURORAL Ev-charger ontology. Figure 10 presents the relation of these modules with the six domains. Each domain is linked to its related modules through different coloured lines. For example, The blue colour determines the modules that contain information about the mobility domain.

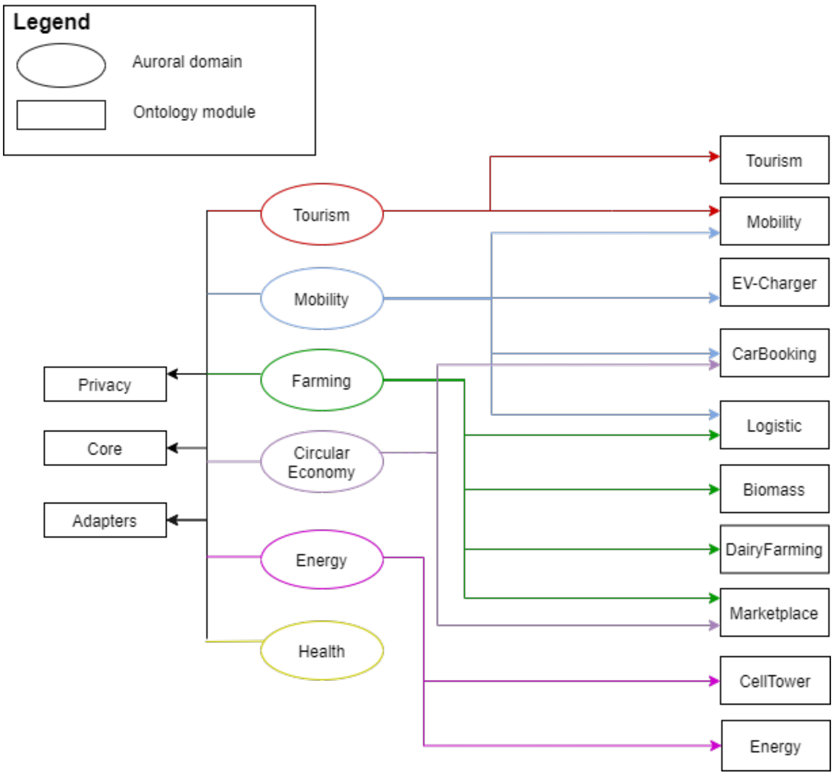


Figure 10 - Auroral ontology network and domains

Figure 11 shows an overview of these ontologies. Coloured boxes are used to represent the developed modules, while white boxes represent reused terms from existing ontologies. The main hierarchies between concepts are also included. These hierarchies are represented by arrows with white endings (triangles) and can be read as follows: the class in the origin of the arrow is a subclass of the class at the end of the arrow. Ad hoc relations between different modules and within modules are also present. Arrows are used to represent these properties between classes and to represent some RDF, RDFS, and OWL constructs. More precisely:

- Plain arrows with white triangles represent the subclass relationship between two classes. The origin of the arrows is the class to be declared as a subclass of the class at the destination of the arrow.
- Plain arrows between two classes indicate that the object property has declared as domain the class in the origin and as range the class in the destination of the arrow. The identifier of the object property is indicated within the arrow.

- Dashed labelled arrows between two classes indicate that the object property can be instantiated between the classes in the origin and the destination of the arrow. The identifier of the object property is indicated within the arrow.
- Dashed arrows with the identifiers between stereotype signs (i.e., “<< >>”) refer to OWL constructs that are applied to some ontology elements, that is, they can be applied to classes or properties depending on the OWL construct being used.
- Dashed arrows with no identifier are used to represent the *rdf:type* relation, indicating that the element in the origin is an instance of the class in the destination of the arrow.

Datatype properties are denoted by rectangles attached to the classes, in a UML-oriented way. Dashed boxes represent data type properties that can be applied to the class to which it is attached, while plain boxes represent that the domain of the datatype property is declared to be the attached class.

For more details on the graphical notation used in diagrams in this and next sections, readers are invited to refer to Chowlk's visual notation [26].

The prefixes, and corresponding ontologies, created and reused in the AURORAL ontology are listed in Table 2 and Table 3, respectively.

Table 2 - List of the ontologies created in the AURORAL Project

Prefix	Namespace
core	https://auroral.iot.linkeddata.es/def/core#
priv	https://auroral.iot.linkeddata.es/def/privacy#
tour	https://auroral.iot.linkeddata.es/def/tourism#
adapt	https://auroral.iot.linkeddata.es/def/adapters#
market	https://auroral.iot.linkeddata.es/def/market#
biomass	https://auroral.iot.linkeddata.es/def/biomass#
logistic	https://auroral.iot.linkeddata.es/def/shipment#
energy	https://auroral.iot.linkeddata.es/def/energy#
car-booking	https://auroral.iot.linkeddata.es/def/car-booking#
mobility	https://auroral.iot.linkeddata.es/def/mobility#
farming	https://auroral.iot.linkeddata.es/def/cow#
cell	https://auroral.iot.linkeddata.es/def/cell#
ev-charger	https://auroral.iot.linkeddata.es/def/electrical-vehicle#

Table 3 - List of the ontologies reused in the AURORAL Project

Prefix	Namespace
hctl	https://www.w3.org/2019/wot/hypermedia#
org	https://www.w3.org/ns/org#
saref	https://saref.etsi.org/core#
geo	http://www.w3.org/2003/01/geo/wgs84_pos#
terms	http://purl.org/dc/terms/
wot	https://www.w3.org/2019/wot/td#

foaf	http://xmlns.com/foaf/0.1/
om	http://www.wurvoc.org/vocabularies/om-2/
lexvo	http://lexvo.org/ontology#
time	https://www.w3.org/2006/time#
geonames	http://www.geonames.org/ontology#
tmjourney	https://w3id.org/transmodel/journeys#
dcat	http://www.w3.org/ns/dcat#

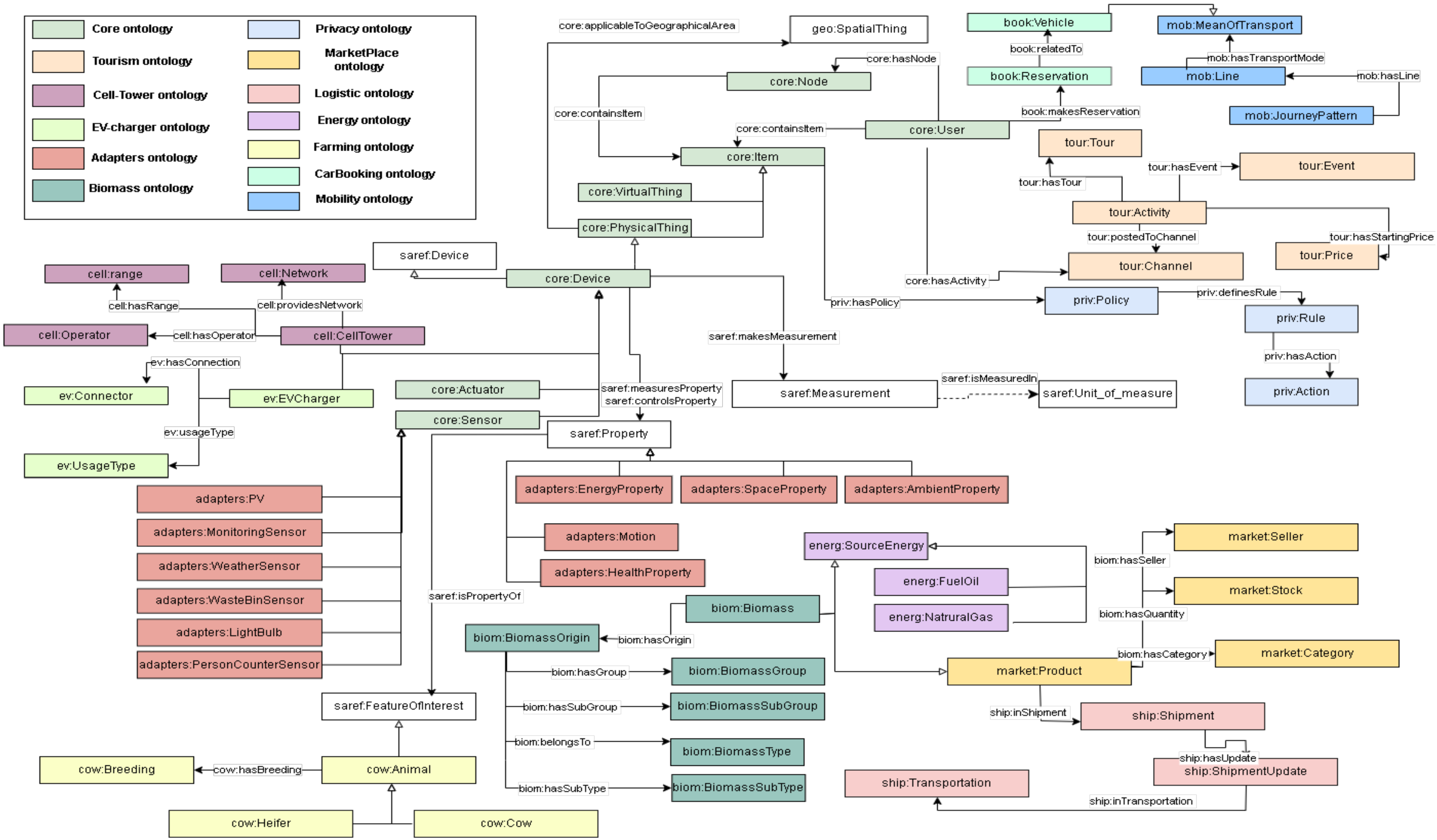


Figure 11 - Overview of AURORAL ontology modules

6. Core Ontology

This section presents an overview about the core ontology. The final conceptual model defined for this ontology is shown in Figure 12. A list of 46 requirements [42] were defined to develop the ontology and a list of 96 tests [43] were defined to evaluate it.

The main classes and properties of this module are the following:

- core:Organisation: is defined as an entity comprising one or more users; core:Organisation is a subclass of org:Organisation in order to reuse its properties. In AURORAL, an organization also has its own properties according to the requirements and its relation to core:Node, core:Notification, core:Audit, and core:User. The core:Organisation class is related also to other external classes, which are geo:SpatialThing and foaf:Image.
- core:Notification is defined as the act of reporting a message to a set of users. This class has its own properties according to the requirements and is related to core:Organisation and core:User.
- core:Node: is defined as the way to organise the users into distinct groups. It has the properties needed to meet the defined requirements and is related to core:Item.
- core:User is defined as a person who belongs to an organisation. This class has its own properties, in order to accomplish the defined requirements, and is a subclass of foaf:Person. The core:User class is related to core:Role, core:Organisation, core:Audit, core:Node, core:Item, and to the external class foaf:Image.
- core:Role is defined as the function assumed by a user inside the organisation. This class has as subclasses the different role types, which can be core:Dev_Ops, core:Admin, core:Dev_owner, core:User, core:Super_user, core:Service_provider, core:Infrastructure_operator, and core:System_integrator.
- core:Item is defined as the physical object that is part of a list of things; core:Item is a subclass of wot:Thing and has two subclasses, core:Device and core:Service.
- core:Audit is defined as the examination and evaluation of the organisations or users.
- core:service: is defined to represent the services of the Auroral project. It has several data properties such as provider, version, number of downloads, etc... It is linked to the lexvo:Language concept with the lexvo:language relationship to specify the language of the service provided.
- core:Domain: is defined to identify the different domains of the service. Six domains are considered in the Auroral project, namely tourism, health, energy, mobility, dairy farming and bicycle economy.
- core:Sensor: is defined to specify the sensor deployed in the Auroral project.
- core:Actuator: is defined to represent the actuator deployed in the Auroral project. Figure 11 shows the main classes and properties of the core ontology. For reasons of visibility, the figure does not contain all the details. Readers are invited to consult the portal site [39].
- core:Dataset: is defined to collect datasets that are used in AURORAL.
- core:TypeService: is defined to specify the type of service: a service could be a commodity service or not commodity.
- core:Readability: presents the readability of a measurement.

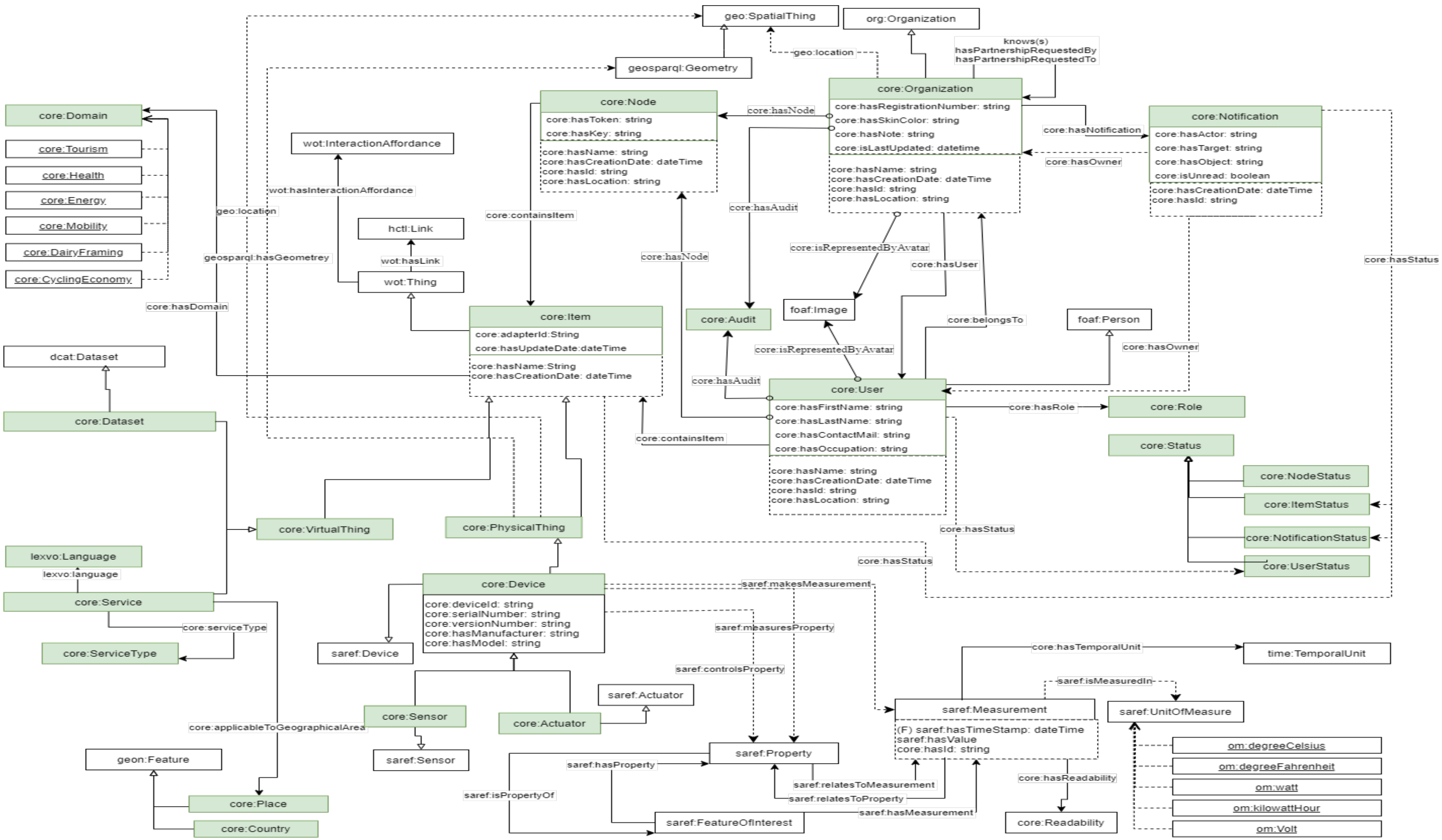


Figure 12 - General overview of the main classes and properties of the AURORAL Core ontology

7. Privacy Ontology

The AURORAL privacy ontology aims to model the privacy for the AURORAL project. A list of 7 requirements [45] were defined to develop the ontology and a list of 11 tests [46] were proposed to evaluate it.

The final conceptual model defined for this ontology module is depicted in Figure 13. The main classes and properties of this module are the following:

- priv:Policy: is defined as a principle of action adopted by a core:Item.
- priv:Rule: is defined as the specification of the state of the policy. It has two subclasses, priv:Permission and priv:Prohibition. The rule has a target (priv:RuleTarget) and an action (priv:Action).
- priv:Action: is defined as the action to be taken by the policy. The priv:Action class has two members: priv:Accessibility and priv:Visibility.
- priv:RuleTarget: is defined as the specification of which users are the target of the policy (in this case, priv:Friends or priv:All).

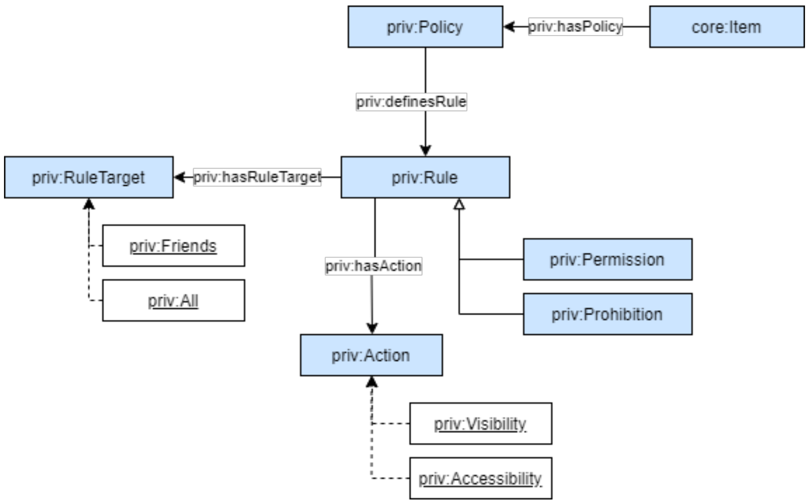


Figure 13 -General overview of the main classes and properties of the AURORAL Privacy ontology

8. Tourism Ontology

The AURORAL tourism ontology aims to model the tourism-related data for the AURORAL project. A list of 19 requirements [47] were defined to develop the ontology and a list of 22 tests [48] were defined to evaluate it.

The conceptual model defined for the Tourism ontology module is depicted in Figure 14. The main classes and properties of this module are the following:

- tour:Activity: is defined as an activity. The activity has a location, defined by geo:Point, a language, defined by dc:LinguisticSystem, has an image, a difficulty, defined by tour:Difficulty and is posted to a channel, defined by the class tour:Channel.
- tour:Channel: Each activity is posted to a channel. Channels are usually used to group activities by theme (restaurants, tours, things to do, etc).
- tour:Price: is the cheapest price for the activity. It's related to tour:Currency, indicating the currency (euros, dollars, etc).
- tour:Difficulty: is defined as the difficulty of an activity or a tour. It can be very easy, easy, medium, hard, or very hard.
- tour:Tour: is defined to determine that an activity can be a tour through the relation tour:hasTour with the class tour:Activity.
- tour:Event: is defined to determine that an activity can be an event through the relation tour:hasEvent.
- tour:Currency: is defined as the currency of the activity price (euros, dollars, etc...).
- tour:Emotion: is defined to present the users emotions about an activity: These emotions are: agree, disagree, and neutral.

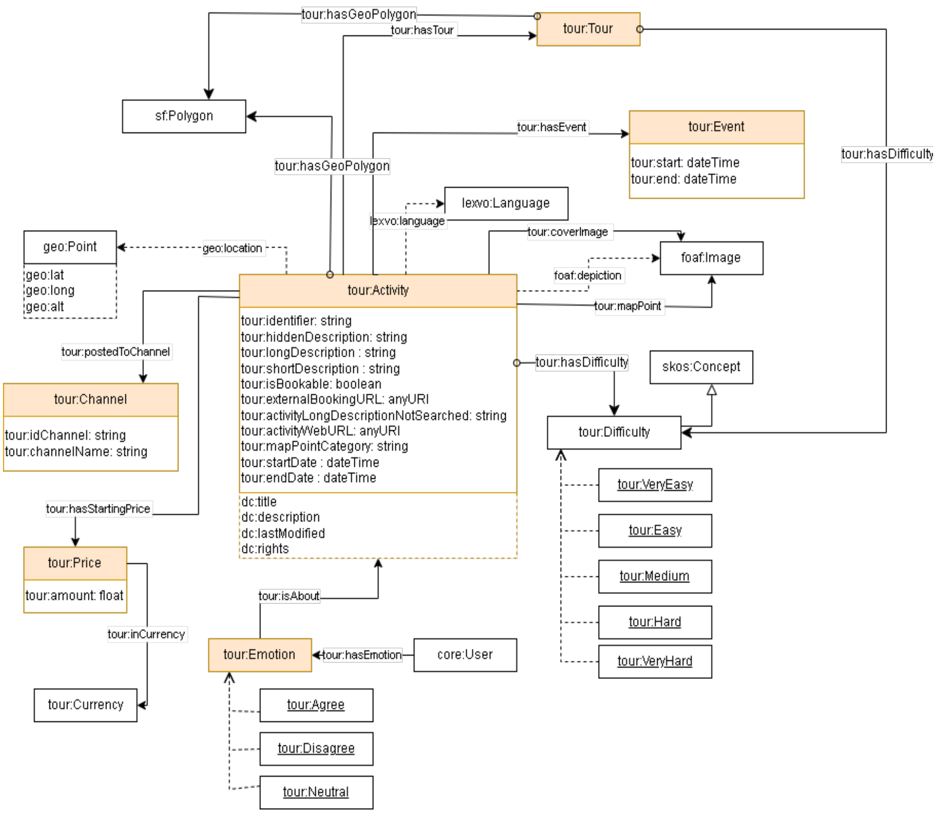


Figure 14 -General overview of the main classes and properties of the AURORAL Tourism ontology

9. Adapters Ontology

The adapters ontology developed for the AURORAL project aims to model the data related to adapters for the AURORAL project. To develop this ontology, a set of 51 requirements [49] was captured. These requirements are classified according to the type of the device. These requirements were identified from different use cases suggested by different pilots, mainly the pilot from Portugal and the pilots from Norway (Halogaland Health and Tourism). In addition, a set of 126 tests were proposed to evaluate it [50].

The main classes and properties of this module are the following:

- adapters:Device: is designed to realize a specific task. This task can be a detection of an occurred event or the initiation of an action. It has two sub-classes (sensor and actuator).
- adapters:Sensor: responds to a stimulus, e.g., a change in the environment, or input data composed from the results of prior observations and generates a result. Sensors detect diverse properties defined by the object properties “saref:measuresProperty” with the range domain the saref:Property concept. It has several sub-classes (adapters:FloodSensor, adapters:EnergyMonitor, adapters:Motionsensor, etc.).
    - adapters:FloodSensor: is designed for detection of water leaks or detecting excessively high-water levels. It has attributes like the identifier and the battery level.
    - adapters:EnergyMonitor: is used to track and control real time energy use. It has different properties such as the returned energy, the total returned energy, etc.
    - adapters:MotionSensor: a motion sensor (or motion detector) is a device that is designed to detect and measure movement.
    - adapters:HumiditySensor: is a device that detects and measures water vapor in its environment and converts its findings into a corresponding electrical signal.
    - adapters:Thermometer: a thermometer is a device used to measure temperature.
    - adapters:GPSEmergencyButton: is a device used to detect the exact location of a specific thing or object. The detected data is the longitude, latitude, altitude, and the acceleration.
    - adapters:WaterAMR sensor: It sends data about the accumulated value of water in the current timestamp.
    - adapters:WeatherSensor: Is a component of an automatic weather station that detects the presence of hydrometers and determines their type (rain, snow, drizzle, etc.) and intensity. It detects data about the temperature, the humidity, the solar irradiation, the pressure, and the rain.
    - adapters:WasteBinSensor: It ensures a real time monitoring of the waste bins' status. It generates data about the distance and the deviated position.
    - adapters:SoilMoistureSensor: it measures the volumetric water content in soil. They provide data about the height and volume of water in soil.
    - adapters:PersonCounterSensor: It is used to count the number of people visiting a city.
    - adapters:QRGeneratorSensor: It calculates the number of people who visit a city and scan a QR code of that location.
    - adapters:LightBulb: Is a glass bulb or tube that produces light when it is supplied with electricity.
    - adapters:AirQualitySensor: This sensor is used to detect metrics to check the air quality.

  - adapters:PresenceSensor: This concept detects the presence of a thing (Person or object) in a given place.
  - adapters:SmartPlug: Is a device that plugs into a standard wall socket and allows further devices to be plugged into it.
  - adapters:BatterySensor: is used to monitor the current, voltage, and temperature of rechargeable batteries.
  - adapters:PV: is an electronic device that converts the energy of light directly into electricity by the photovoltaic effect.
- adapters:Actuator: is a device that is used by, or implements, an (actuation) procedure that changes the state of the world. It is related to the ssn:Property class through the object property sosa:forProperty, indicating the set of properties on which the actuator is based to trigger the action. It has the adapters:WaterValveActuator as sub-class.
  - adapters:WaterValveControl: it is used to control the water valve. It generates an acknowledgement message indicating that the command has been accepted. The generated command can be: ON, OFF, active, close.
  - adapters:Switch: is a device for making and breaking the connection in an electric circuit.
  - adapters:Relay: is the device that opens or closes the contacts to cause the operation of the other electric control.
  - adapters:SmartValve: is a device for remotely switching the supply of water, gas, electricity, etc.
- saref:Property: refers to the observable properties concerning a particular domain or use case. It has diverse sub-classes as presented below.
  - adapters:AmbientProperty: it groups the ambient properties that are detected by the deployed devices. These properties are the ambient temperature, the ambient humidity, the ambient flood, the ambient wind, and the ambient solar radiation properties, the precipitation, the weather, the noise, the visibility among others…
  - adapters:SpaceProperty: describes properties related to space such as the GPS latitude, the GPS altitude, the GPS longitude, the X-axis acceleration, the Y-axis acceleration, the Z-axis acceleration, the distance, etc.
  - adapters:EnergyProperty: this concept groups the properties related to energy such as: the battery level, the total returned energy, the total energy, the power factor, the grid voltage, and so one.
  - adapters:MotionProperty: it is about the motion and the vibration properties.
  - adapters:LightProperty: it describes the properties about luminance.
  - adapters:Sound: it defines the noise level properties.
  - adapters:Pressure: provides information about the pressure properties (for example the atmospheric pressure).
  - adapters:rain: refers to the accumulated precipitation.
  - adapters:SoilMoisture: presents the soil moisture: height and volume.
  - adapters:CarbonDioxide: regroups measurement about the CO_2 concentration.
  - adapters:HealthProperty: presents properties related to the health of the person such as systolic blood pressure, diastolic blood pressure, protein, pulse rate and so on...

Figure 15 shows only a fragment of the adapter ontology to ensure the visibility of the figure. For more details, readers are invited to consult the online version.

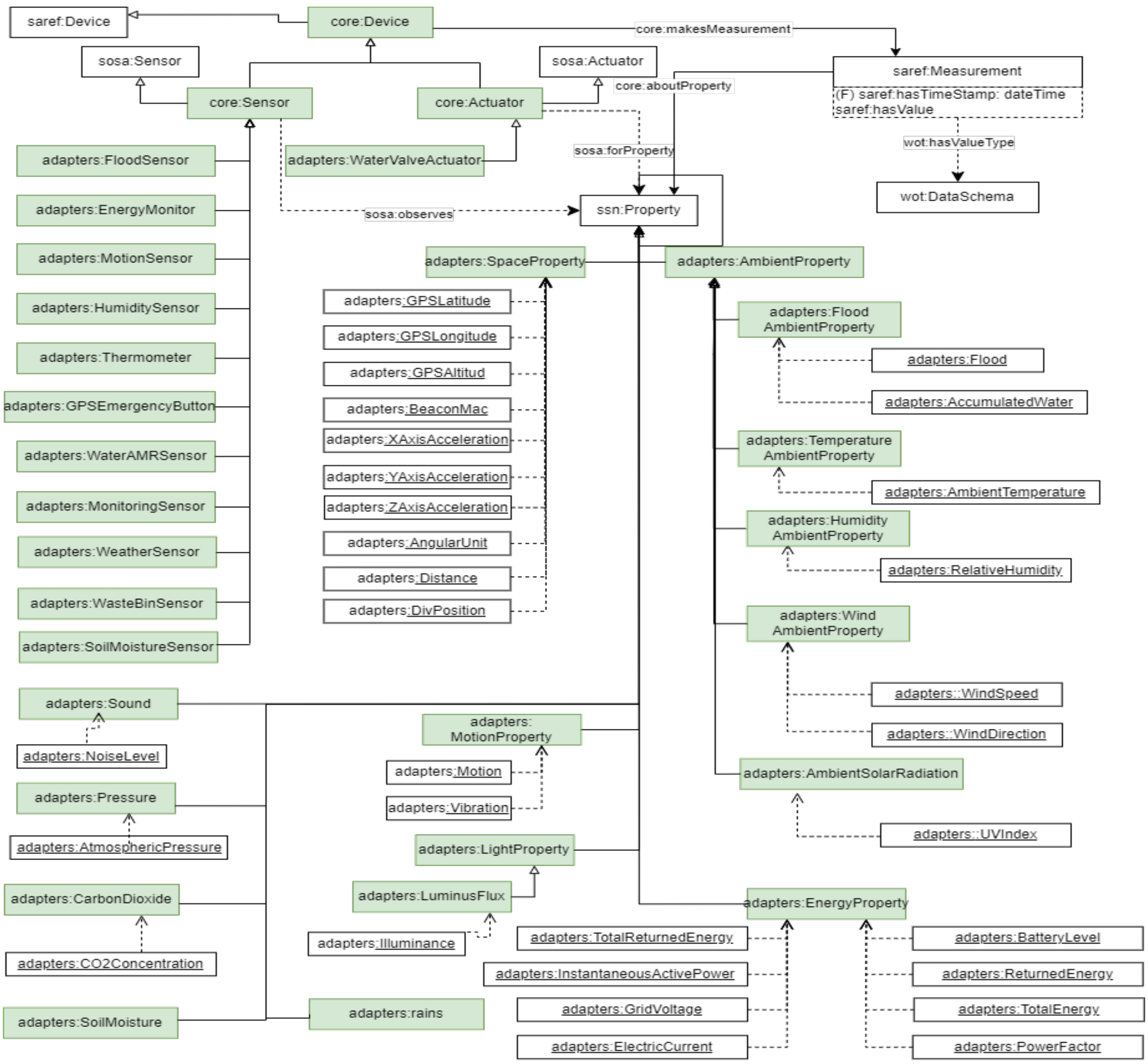


Figure 15 - General overview of the main classes and properties of the AURORAL Adapters ontology

10. Biomass Ontology

The biomass ontology presents the characteristics of biomass. A set of 19 requirements [51] was captured. The requirements for biomass characteristics are taken from the Spain Pilot use case. In addition, a set of 23 tests [52] were defined to evaluate it.

The main concepts of the Biomass ontology are the following.

- biom:Biomass: defines the biomass (for example the Briquette) and their properties such as the unique identifier, the name, the trading Form, the dimension, etc.
- biom:BiomassOrigin: is proposed to identify the source of the biomass. It has as data property the identifier and a description. The BiomassOrigin is linked to the BiomassGroup, BIomassSubGroup, BiomassType, and BiomassSubType, in the object properties hasGroup, hasSubGroup, belongsToType, hasSubType respectively.
- biom:BiomassGroup: identifies the group to which a biomass belongs. According to ISO/FDIS 17225-1, there are five groups: woody biomass, herbaceous biomass, fruit biomass, aquatic biomass, and the group of mixtures and blends.
- biom:BiomassSubGroup: specifies the main subgroups to which a biomass can belong. Each group for the four groups mentioned above is composed of subgroups. Taking the WoodyBiomass group as an example, it is composed of the following subgroups (forest plantation and other virgin wood, by products and residues from wood processing industry, used wood, and blends and mixtures). This concept is related to the Biomass concept with the relation hasSubGroup.
- biom:BiomassType: determines the type of biomass. Each subgroup of each group is composed of various types. For example, the used wood sub-group of the woody biomass group has three types: the chemically untreated used wood, the chemically treated used wood, and the blends and mixtures. It is linked to the BiomassOrigin class with the relation belongsTo.
- biom:BiomassSubType: it specifies the subtypes of each type of biomass.For example, the type chemically, untreated used wood has three sub-types: Without bark, with bark, and bark. This class is related to the BiomassOrigin through the relation hasSubType.
- biom:Humidity: determines the humidity of the biomass.
- biom:Ashes: presents the biomass ashes. Ash is the residue after combustion.
- biom:Quantity: it specifies the quantity of a biomass. There are four types of quantities namely: total quantity, current quantity, bought quantity, sold quantity.
- biom:BoughtPerGroup: Bought quantity is classified according to the subGroup of biomass. This class is defined to present the group of each sub-quantity.
- Biom:SoldPerGroup: Sold quantity is classified according to the subGroup of biomass. This class is defined to present the group of each sub-quantity.
- biom:SoldQuantityPerCaliber: Sold quantity is classified according to the caliber and the trading form: This class is related to the Trading form and the Caliber to present this information.
- biom:TradingForm: defines the trading form of a biomass or a sold quantity.
- biom:Caliber: defines the caliber for each sold quantity of biomass.

Figure 16 describes the main classes and properties of the AURORAL Biomass ontology.

For reasons of clarity, Figure 16 presents the biomass group and the biomass sub-group. for more details about Biomass group, sub-group, type and sub-type readers are invited to consult the full version of the Biomass ontology.

To describe the hierarchy between biomass group, subgroup type and subtype, the SKOS ontology was reused by defining the skos:broader and skos:narrower between these classes.

To specify that one concept has a more generic meaning than another, the relation skos:broader is used. For example, to represent that biom:WoodyBiomass is more generic than biom:PlantationandForest, the relationship skos:broader is defined as follows: biom:PlantationandForest **skos:broader** biom:WoodyBiomass.

To specify that one concept has a more specific meaning than another, the skos:narrower relation is used.

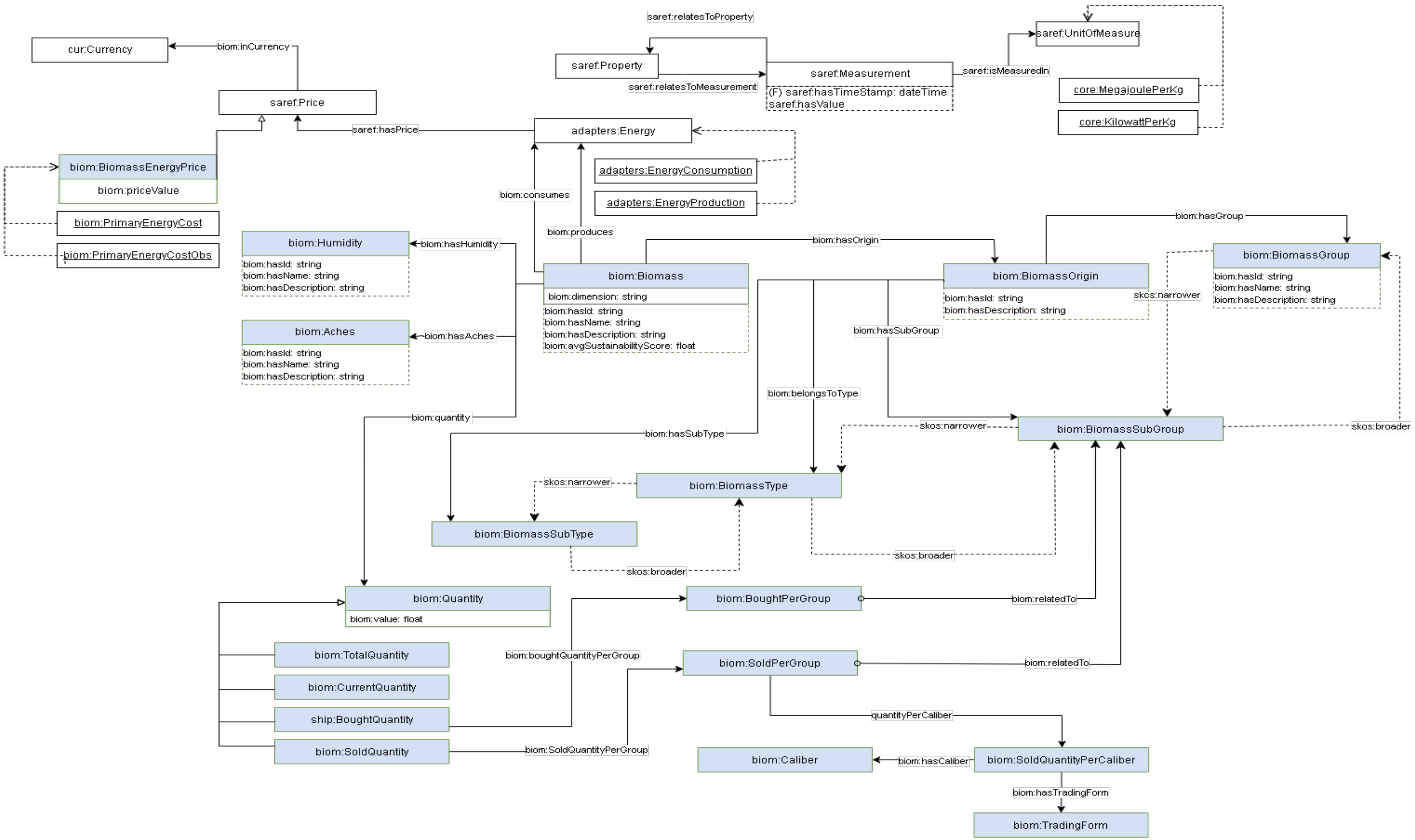


Figure 16- General Overview of the AURORAL Biomass ontology

11. Logistic Ontology

The logistic module presents knowledge relating to the shipment of a given product. It is mainly composed of a set of 33 requirements [53] provided by the Spain Pilot. To evaluate it, a set of 23 tests [54] has been defined.

The main concepts and relations between the concepts of the Biomass shipment ontology.

- ship:Product: This concept specifies the product that will be delivered like for example the Biomass.
- ship:Shipment: This concept defines the main information of the shipment process namely the identifier and the created time.
- ship:Company: This concept describes the company of the product that is delivered.
- ship:ShipmentUpdate: This concept tracks the updates in the product status during the shipment process. It has data properties like the identifier, the time update and the type.
- ship:Transportation: This concept determines the means of transport of the product is shipment.
- ship:Storage: determines the storage location in delivery during shipment update.
- ship:StoragePile: means any outdoor storage on a source's property of material. Every storage has a storage pile.
- ship:Document: This concept specifies the document where the shipment update is reported.
- ship:DocumentType: This concept specifies the type of the used document to report the shipment update of the product.
- Ship:address: specifies the pickup and the delivery address of a shipment
- Ship:time: specifies the pickup and the delivery time of a shipment
- core:User: reused from the core module to defines the engaged users in the logistic process: There are two types of users: the sender and the receiver.
- ship:Price: determines the buyer and the seller price of a shipment.
- ship:Currency: determines the currency of a price.

Figure 17 describes the main classes and properties of the AURORAL Logistic ontology.

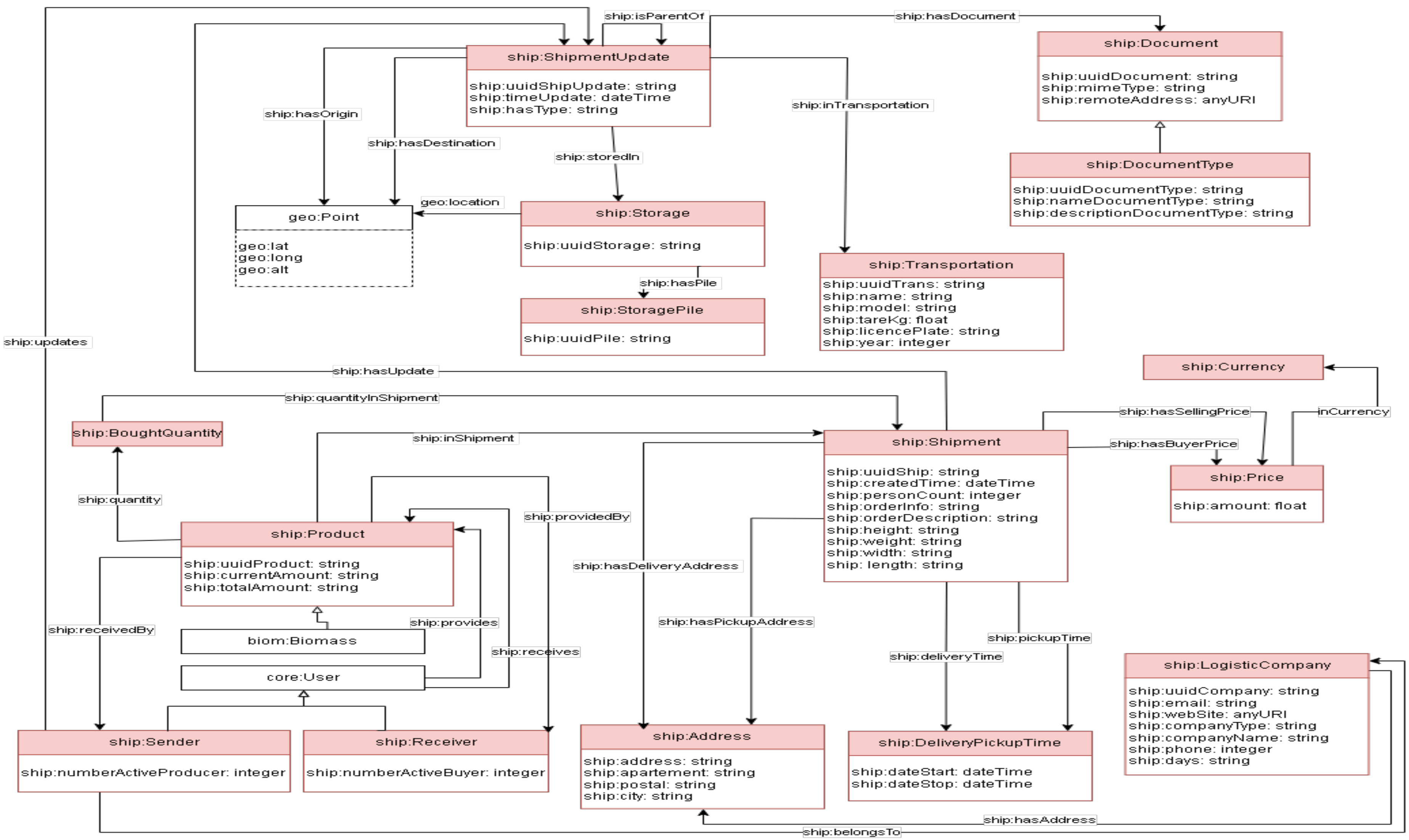


Figure 17 - General overview of the main classes and properties of the AURORAL Logistic ontology

12. Marketplace Ontology

This section presents an overview of the AURORAL marketplace ontology, explaining and defining its main classes. To develop the Marketplace ontology, a set of 8 requirements [55] was captured. These requirements were taken from a use case provided by the Spain Pilot. In addition, a set of 6 tests [56] were defined to evaluate it.

The main concepts and relations between the concepts of the Marketplace ontology are:

- market:Seller: determines the seller of a given product.
- market:Product: determines the product on a marketplace.
- market:Customer : determines who is the customer of a given product.
- market:Stock : determines the quantity of a product in a stock.
- market:Category: determines the category of a product on a marketplace.

Figure 18 describes the main classes and properties of the AURORAL Biomass ontology.

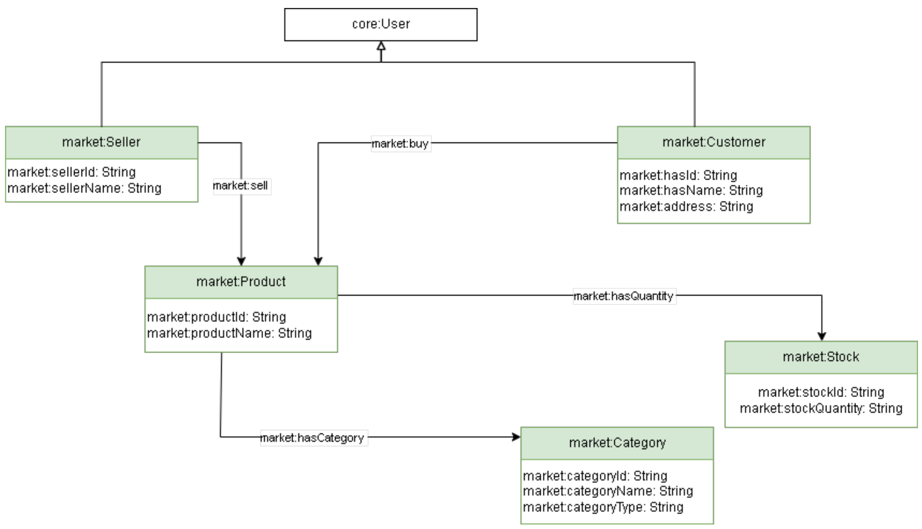


Figure 18 - General overview of the main classes and properties of the AURORAL MarketPlace ontology

13. Energy Ontology

This section presents an Overview about the Energy module. To develop the energy ontology, a set of 16 requirements [57] was captured. Requirements about energy ontology were identified based on a use case provided by the Sweden Pilot. In addition, a set of 40 tests [58] were created to evaluate it.

The main concepts and relations of the Energy ontology are described below:

- saref:FeatureofInterest: This concept is reused to determine the energy collected for each feature of interest. In this ontology, the building and its municipality are the features of interest.
- saref:Property: This concept determines the total energy consumption and intensity of energy use in a given building. For this need, the adapters:Energy concept was reused from the adapters ontology.
- om:UnitOfMeasure: This concept determines the unit of the data collected, which can be watt, kilowatt, etc.
- energ:SourceEnergy: It specifies the source of the collected energy, of which there are five.
- energ:Price: It determines the price of energy consumed per Kilowatt.
- energ:Consumer: It defines the different categories of energy consumers.
- energ:PriceAgreement: This class specifies the type of agreement for energy consumption. The agreement types are: fixed agreement, variable agreement and standard agreement.
- energ:CostPriceSeason: This concept specifies the cost of energy consumption, which changes according to the season.
- energ:Season: This concept contains the four seasons (winter, autumn, spring, summer).

Figure 19 describes the main classes and properties of the AURORAL Energy ontology.

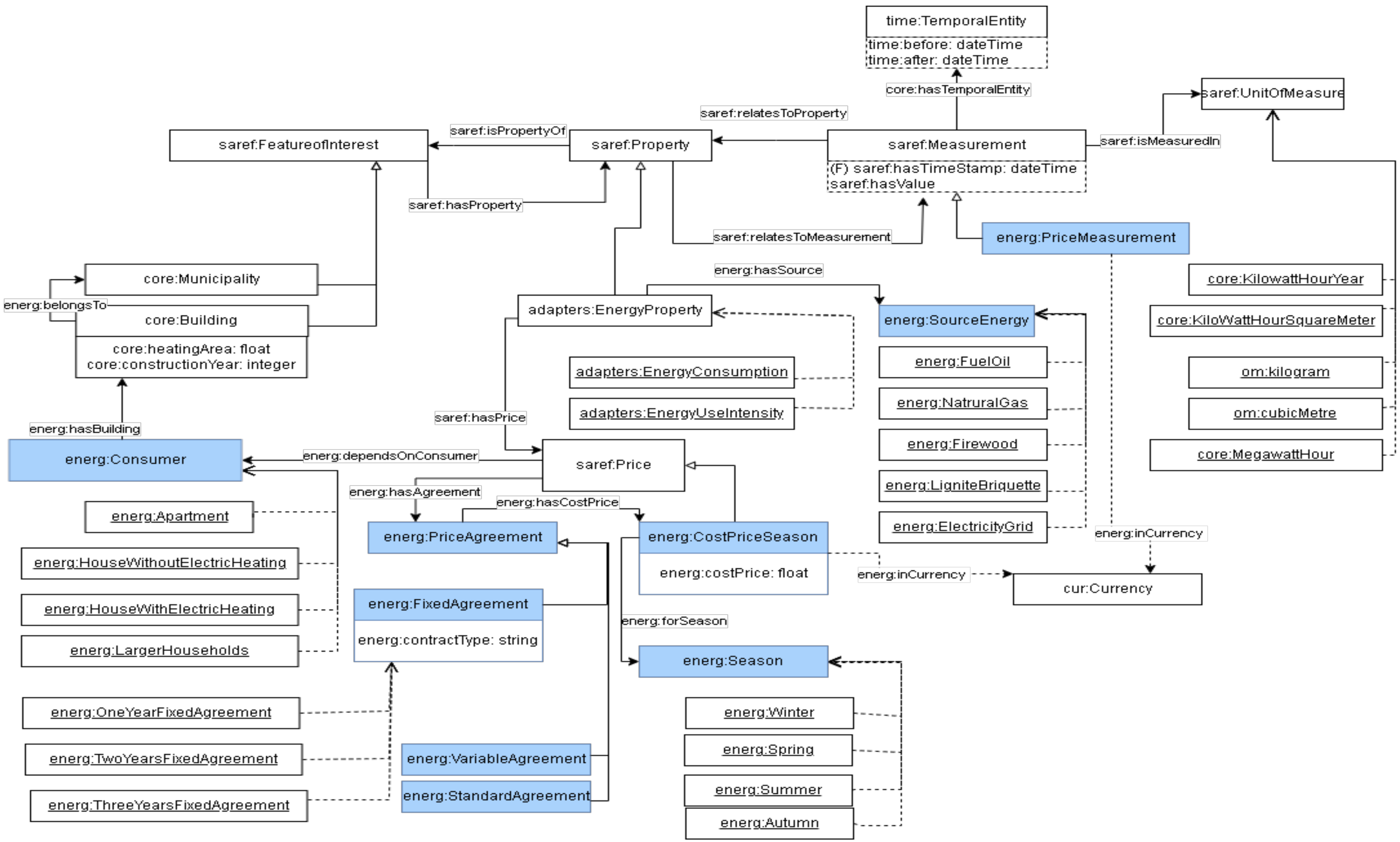


Figure 19 - General overview of the main classes and properties of the AURORAL Energy ontology

14. CarBooking Ontology

This section presents an overview of the AURORAL CarBooking module, explaining and defining its main classes. To develop the CarBooking ontology, a set of 8 requirements [59] was captured. These requirements were taken from a use case provided by the Austria Pilot

The main concepts and relations of the Marketplace ontology are presented below.

- book:Reservation: This class defines the main attributes related to a reservation such as the identifier, the date, the beginning and the end of the reservation ,etc.
- book:Vehicle: This class defines the information of the vehicle to be reserved such as the identifier, the label, the model, etc.
- core:User: This class presents the information related to the user that will make the reservation.
- book:Address: This class presents the address of the user, and the vehicle like the city and the code zip.

Error! Reference source not found. describes the main classes and properties of the AURORAL CarBooking ontology.

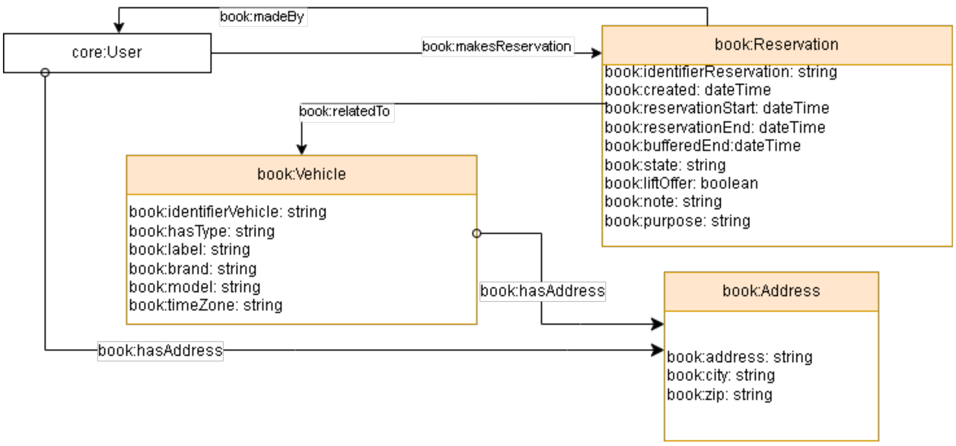


Figure 20 - General overview of the main classes and properties of the AURORAL CarBooking ontology

15. Mobility Ontology

This section presents an overview of the AURORAL mobility module, explaining and defining its main classes. To develop the mobility ontology, a set of 18 requirements [61] was captured. These requirements were taken from a use case provided by the Austria Pilot. A set of 21 tests [62] were defined to evaluate the mobility ontology. I

The main concepts and relations of the mobility ontology are described below. This ontology reuses diverse concepts from the transmodel vocabularies [28] that is related to the transport systems.

- mob:StopPoint: it determines the stop points of the bus during a tourism activity.
- mob:EstimatedCall: this class defines the estimated call of a bus in a specific stop point.
- mob:quay: defines the quay points of the estimated call.
- mob:ScheduledTime: presents the scheduled time (arrival and departure times) of the estimated call.
- mob:JourneyService: defines the journey service of the estimated call.
- tmjourney:JourneyPattern: defines the journey patterns of the journey service.
- tmjourney:Line: defines the line of a journey pattern.
- tmjourney:MeanOfTransport: defines the different means of transport (bus, metro, rail, gondola, ferry, etc.)

Error! Reference source not found. describes the main classes and properties of the AURORAL Mobility ontology.

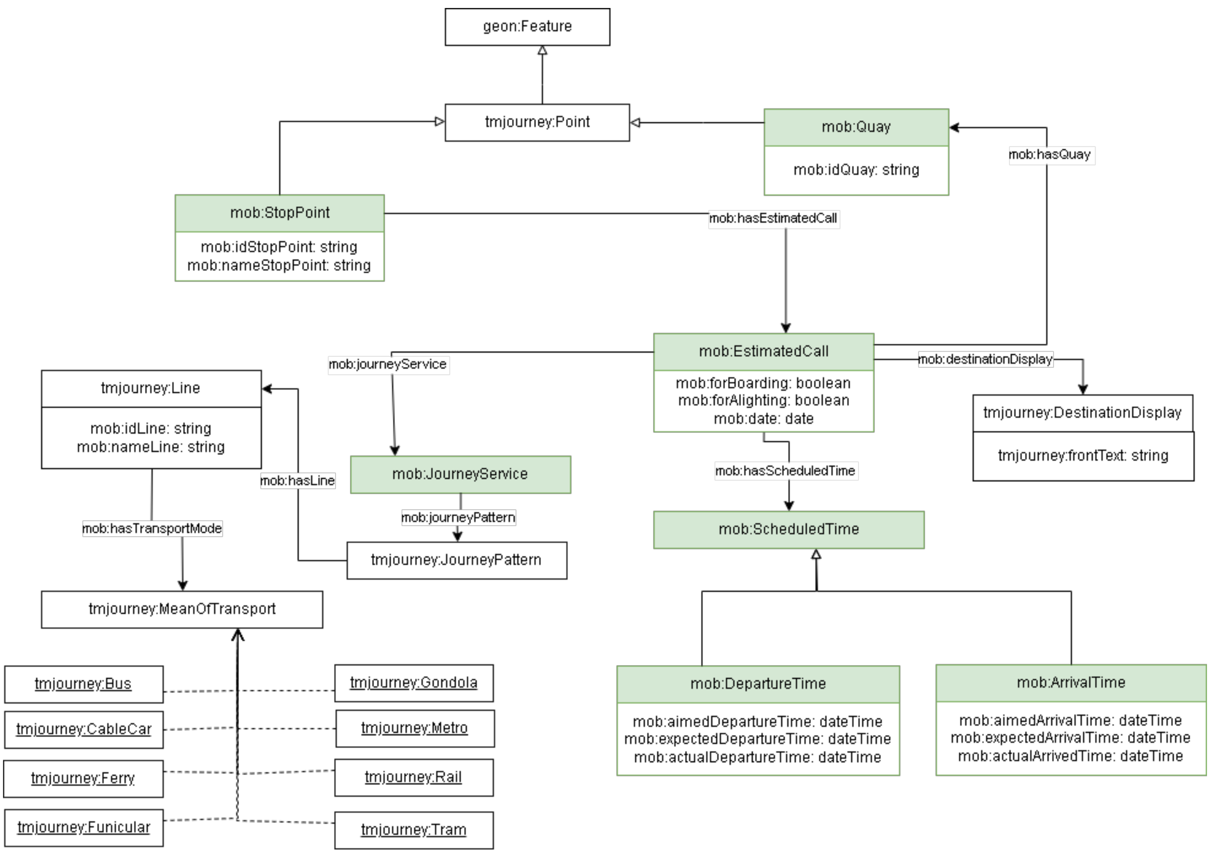


Figure 21 - General overview of the main classes and properties of the AURORAL Mobility ontology

16. Farming Ontology

This section presents an overview of the AURORAL Farming module, which presents information about the dairy production of cows. To develop the farming ontology, a set of 14 requirements [63] were taken from a use case provided by the Italian Pilot. A set of 20 tests [64] were defined to evaluate it.

The main concepts and relations of the farming ontology are described below.

- saref:featureOfInterest: This class is reused from the SAREF ontology to represent the information of the feature of Interest in the farming domain which is the cow and the heifer.
- cow:Animal: This class presents the information of the animal which is the feature of Interest. This class has two sub-classes: the cow:Cow and the cow:Heifer.
- cow:Activity: this class presents information related to the activity of the animal.
- cow:Breeding: This class presents the reproduction of the animal.
- cow:Lactation: This class defines the lactation days of the cow.
- cow:ReproductiveStatus: This class presents the reproductive status of the animal that could be: Pregnant, to undergo artificial insemination, does not need artificial insemination, recent delivery and inseminated. These statuses are defined as sub-classes of the cow:ReproductiveStatus class.
- core:Sensor: This class is reused from the core ontology to represent the Accelerometer which is assigned to the animal to monitor its activities.
- saref:Property: This class is reused from the SAREF ontology to present the measured property for the sensor assigned to the Animal. These properties are the adapters:XAxisAcceleration, the adapters:YAxisAcceleration and the adapters:ZAxisAcceleration .

Error! Reference source not found. describes the main classes and properties of the AURORAL Farming ontology.

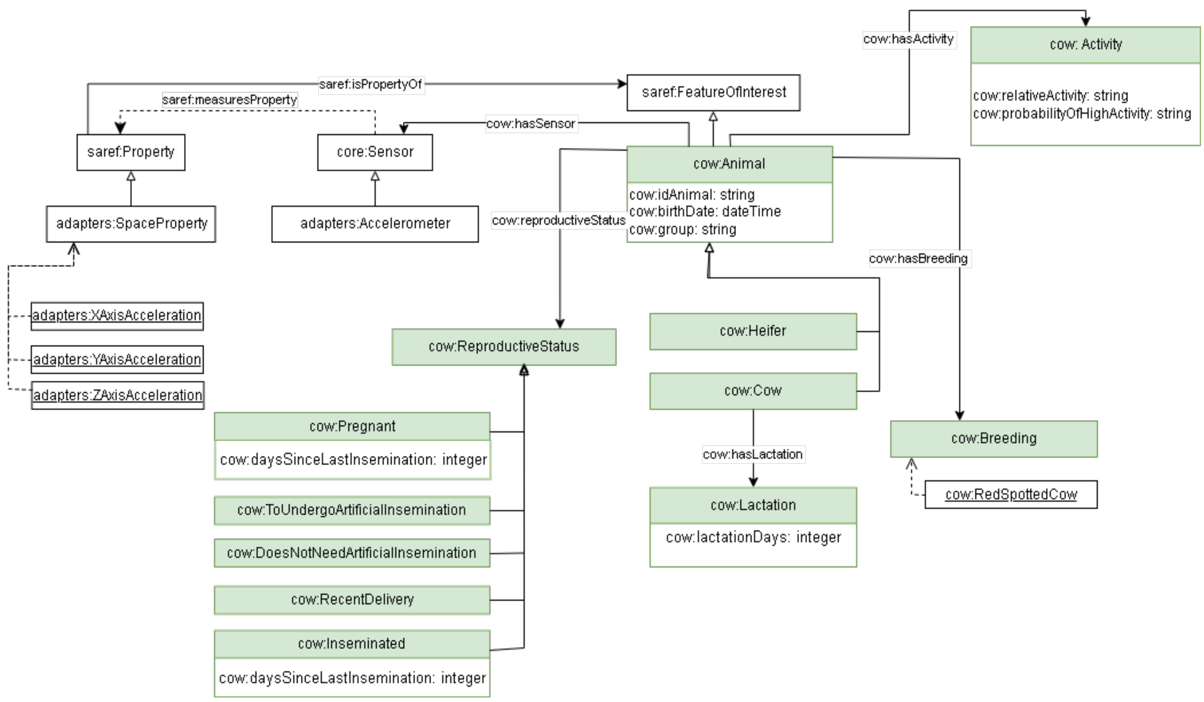


Figure 22 - General overview of the main classes and properties of the AURORAL Farming ontology

17. Cell-Tower Ontology

This section presents an overview of the AURORAL Cell-Tower module. To develop the cell-tower ontology, a set of 8 requirements [65] were taken from a use case provided by the Austrian Pilot. A set of 13 tests [66] were defined to evaluate it.

The main concepts and relations of this ontology are described below.

- core:Device presents information about a physical device. In this context the device is the cell:CellTower.
- cell:Range: This class presents the range of the cell tower.
- cell:Operator: this class presents information related to the operator of the cell tower.
- cell:Network: this class presents the network provided by the cell tower. The types of the network are: LTE, GSM, UMTS, 4G, 5G, and CDMA.
- cell:Country: this class defines the country where the cell tower is located.
- geo:Point: this class presents the geographical position of the cell tower.

Error! Reference source not found. describes the main classes and properties of the AURORAL Cell-Tower ontology.

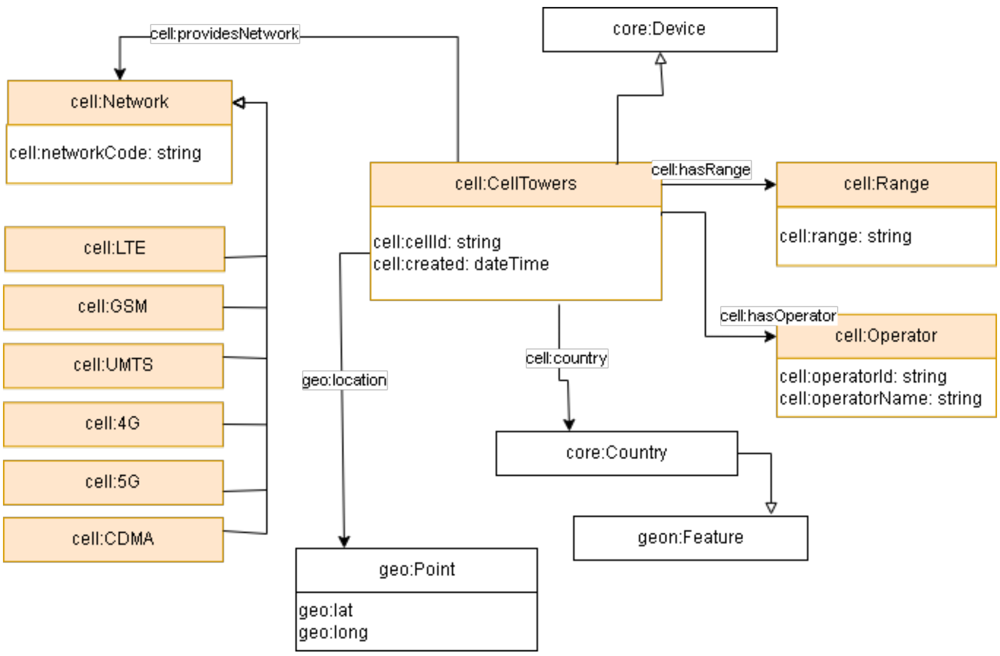


Figure 23 - General overview of the main classes and properties of the AURORAL Cell-Tower ontology

18. Electrical-Vehicle Charger Ontology

This section presents an overview of the AURORAL electric vehicle charger module. To develop this ontology, a set of 23 requirements [67] were taken from a use case provided by the Austrian pilot. A set of 19 tests [68] was defined to evaluate it.

The main concepts and relationships of this ontology are described below.

- ev:EvCharger: presents the related information of the electrical vehicle (ev) charger device.
- ev:DataProvider: defines the data provider of the ev-charger.
- book:Address: this concept is reused from the car-booking ontology to present the address of the ev-charger that belongs to a country and has a geographical point.
- ev:UsageType: defines the usage type of the ev-charger.
- cell:Operator: defines the information related to the operator of the ev-charger. This concept is reused from the cell-tower ontology.
- ev:Connector: defines the connectors of the ev-charger: The connector has two different types: the current types and the Connector types.
- saref:Property: is reused from the SAREF ontology to present the properties that are detected from the ev-charger. These properties are the Energy Property and the Status Property.

Error! Reference source not found. describes the main classes and properties of the AURORAL EV-charger ontology.

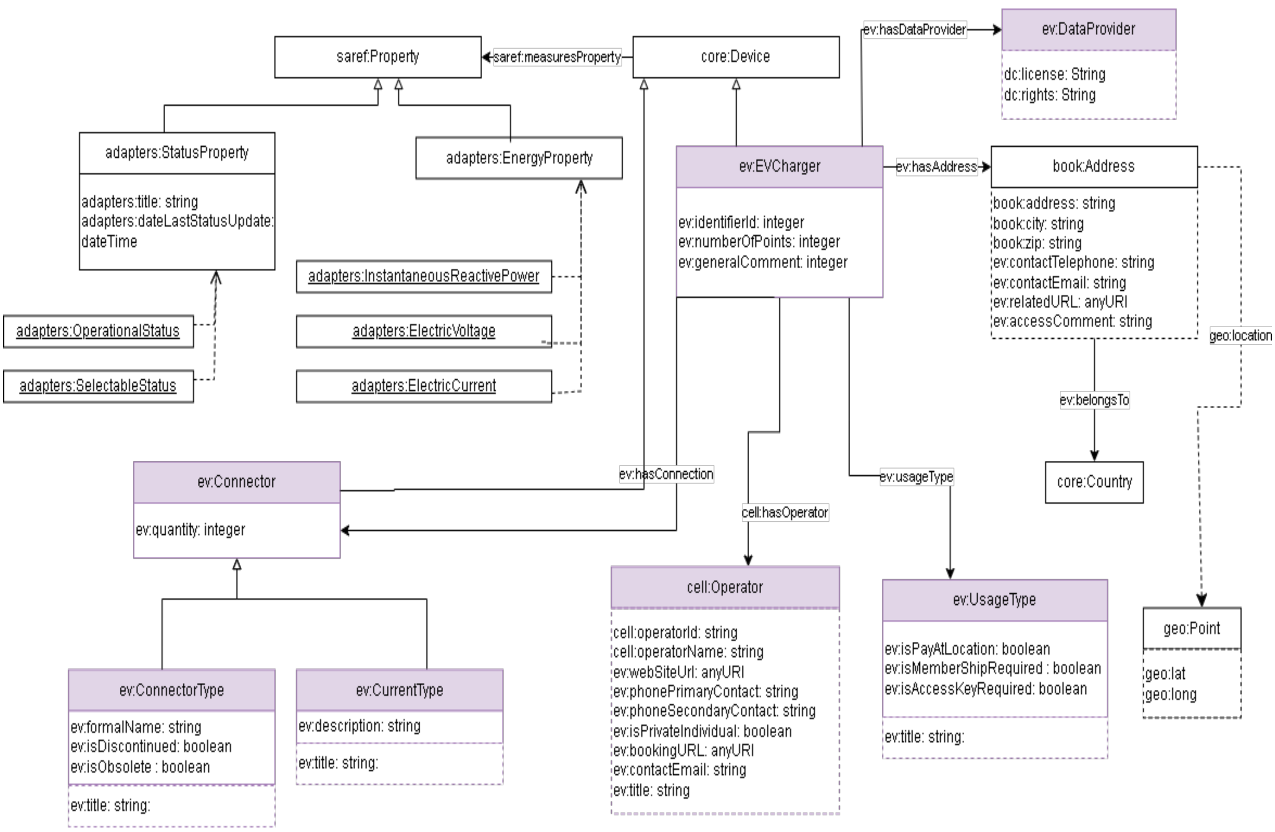


Figure 24 - General overview of the main classes and properties of the AURORAL EV-Charger ontology

19. Conclusion

This document describes the methodology and technological infrastructure used to create the AURORAL ontology network. In addition, it proposes a governance model designed to guarantee the interoperability, discoverability and comprehensibility of the ontologies developed. The governance model includes fundamental principles and best practices that guide the entire process of creating the AURORAL ontology network.

The AURORAL ontology network comprises thirteen modules that have been described along this document including the existing ontologies being reused during the AURORAL ontology network development. These modules have already been designed, implemented and published in various versions as can be observed online in the AURORAL ontology portal.

The "core" and "privacy" modules are at a more generic level than the other modules. The core ontology contains general information about the users, organizations, services, devices, etc. that are considered in the project. The privacy ontology models needed information to allow the confidentiality of data exchanged via the auroral platform. The adapters module contains information on the sensors used and their properties in various use cases. The other modules are more specific and are defined for a particular use case.

To ensure that the modules developed cover all the necessary requirements, a set of tests was developed. Themis was used to carry out this evaluation. In addition, to ensure that the ontology is well established, OOPS! was used to detect any pitfalls during ontology implementation. In addition, a set of payloads was defined for each module to facilitate understanding and use.

20. References

[1] M. Poveda-Villalón, A. Fernández-Izquierdo and R. García-Castro, « LOT: An industrial oriented ontology engineering framework » Engineering Applications of Artificial Intelligence, 111, 104755 2022: http://doi.org/10.5281/zenodo.2539305.

[2] M. C. Suárez-Figueroa, A. Gómez-Pérez and M. Fernández-López, «The NeOn Methodology for Ontology Engineering,» de Ontology engineering in a networked world, Springer, Berlin, Heidelberg, 2011.

[3] R. García-Castro, A. Fernández-Izquierdo, C. Heinz, P. Kostelnik, M. Poveda-Villalón and F. Serena, «D2.2. Detailed Specification of the Semantic Model,» 2017.

[4] S. Chávez, F. Bosché, N. Bountouni, S. Fenz, R. García-Castro, G. Giannakis, S. González-Gerpe, S. Kollmer, S. Kousouris, D. Ntalaperas, T. Thanos, F. Lampathaki, M. Poveda-Villalón, E. Valero, D. Vergeti and F. Tavakolizadeh, «D4.2 BIMERR Ontology & Data Model,» 2020.

[5] A. Fernández-Izquierdo, A. Cimmino, R. García-Castro, M. Poveda-Villalón, S. Terzi and C. Patsonakis, «T1.3 DELTA Ontology and Data Modelling Framework,» 2019.

[6] COGITO ontologies https://cogito.iot.linkeddata.es/

[7] easyTV ontology https://w3id.org/def/easytv

[8] Alhassan, I., Sammon, D., & Daly, M. (2016). Data governance activities: an analysis of the literature. Journal of Decision Systems, 25(sup1), 64-75.

[9] International, D. (2017). DAMA-DMBOK: data management body of knowledge. Technics Publications, LLC.

[10] Iglesias-Molina, Ana, Bernabé-Díaz, José Antonio, Deshmukh, Prashant, Espinoza-Arias, Paola, Ayllón-Benítez, Aarón, Fernández-Izquierdo, Alba, Ponce-Bernabé, José María, Pérez, Sara, Ruckhaus, Edna, Corcho, Oscar and Sánchez-Fernández, José Luis (2022). Ontology Management in an Industrial Environment: The BASF Governance Operational Model for Ontologies (GOMO). In: "ISMB 2022", July 10–14, Madison

[11] Alobaid, A., Garijo, D., Poveda-Villalón, M., Santana-Pérez, I., Fernández-Izquierdo, A., & Corcho, O. (2019). Automating ontology engineering support activities with OnToology. Journal of Web Semantics, 57, 100472.

[12] D. Garijo, «WIDOCO: a wizard for documenting ontologies,» de International Semantic Web Conference, 2017.

[13] Poveda-Villalón, María, Asunción Gómez-Pérez, and Mari Carmen Suárez-Figueroa. "Oops!(ontology pitfall scanner!): An on-line tool for ontology evaluation." International Journal on Semantic Web and Information Systems (IJSWIS) 10.2 (2014): 7-34.

[14] Chávez-Feria, S., García-Castro, R., Poveda-Villalón, M. (2022). Chowlk: from UML-Based Ontology Conceptualizations to OWL. In: , et al. The Semantic Web. ESWC 2022. Lecture

Notes in Computer Science, vol 13261. Springer, Cham. https://doi.org/10.1007/978-3-031-06981-9_20

[15] Jackson, R., Matentzoglu, N., Overton, J. A., Vita, R., Balhoff, J. P., Buttigieg, P. L., ... & Peters, B. (2021). OBO Foundry in 2021: operationalizing open data principles to evaluate ontologies. *Database*, *2021*, baab069.

[16] KARRAY, Mohamed, OTTE, Neil, RAI, Rahul, *et al.* The industrial ontologies foundry (IOF) perspectives. 2021.

[17] Fernández-Izquierdo, A., & García-Castro, R. (2019). Themis: a tool for validating ontologies through requirements. In *SEKE* (pp. 573-753).

[18] https://www.w3.org/2006/time#

[19] https://saref.etsi.org/

[20] http://www.w3.org/2003/01/geo/wgs84_pos#

[21] https://www.w3.org/2019/wot/td#

[22] http://www.wurvoc.org/vocabularies/om-2/

[23] https://protege.stanford.edu/

[24] https://www.topquadrant.com/

[25] https://www.poolparty.biz/

[26] https://chowlk.linkeddata.es/

[27] https://obofoundry.org/principles/fp-000-summary.htm

[28] https://app.diagrams.net/

[29] https://www.w3.org/TR/vocab-org/

[30] https://www.w3.org/TR/odrl-model/

[31] http://xmlns.com/foaf/0.1

[32] http://lexvo.org/ontology

[33] https://www.w3.org/2019/wot/hypermedia

[34] https://www.geonames.org/ontology#

[35] https://w3id.org/transmodel/journeys#

[36] http://www.w3.org/ns/dcat#

[37] https://github.com/oeg-upm/auroral-core-ontology

[38] https://ontoology.linkeddata.es/

[39] https://github.com/dgarijo/Widoco

[40] https://oops.linkeddata.es/response.jsp

[41] https://themis.linkeddata.es/tests-info.html

[42] https://auroral.iot.linkeddata.es/requirements/requirements-core.html

[43] https://github.com/oeg-upm/auroral-core-ontology/tree/master/tests

[44] https://auroral.iot.linkeddata.es/payload.html

[45] https://auroral.iot.linkeddata.es/requirements/requirements-priv.html

[46] https://github.com/oeg-upm/auroral-privacy-ontology/tree/master/tests

[47] https://auroral.iot.linkeddata.es/requirements/requirements-tourism.html

[48] https://github.com/oeg-upm/auroral-tourism-ontology/tree/main/tests

[49] https://auroral.iot.linkeddata.es/requirements/requirements-adapters.html

[50] https://github.com/oeg-upm/auroral-adapters-ontology/tree/main/tests

[51] https://auroral.iot.linkeddata.es/requirements/requirements-biomass.html

[52] https://github.com/oeg-upm/auroral-biomass-ontology/tree/main/tests

[53] https://auroral.iot.linkeddata.es/requirements/requirements-shipment.html

[54] https://github.com/oeg-upm/auroral-shipmentBiomass-ontology/tree/main/tests

[55] https://auroral.iot.linkeddata.es/requirements/requirements-marketplace.html

[56] https://github.com/oeg-upm/auroral-marketplace-ontology/tree/main/tests

[57] https://auroral.iot.linkeddata.es/requirements/requirements-energy.html

[58] https://github.com/oeg-upm/auroral-energy-ontology/tree/main/tests

[59] https://auroral.iot.linkeddata.es/requirements/requirements-carbooking.html

[60] https://github.com/oeg-upm/auroral-Car-booking-ontology/tree/main/tests

[61] https://auroral.iot.linkeddata.es/requirements/requirements-mobility.html

[62] https://github.com/oeg-upm/auroral-mobility-Ontology/tree/main/tests

[63] https://auroral.iot.linkeddata.es/requirements/requirements-cow.html

[64] https://github.com/oeg-upm/auroral-cow-ontology/tree/main/tests

[65] https://auroral.iot.linkeddata.es/requirements/requirements-cellTower.html

[66] https://github.com/oeg-upm/auroral-cellTower-ontology/tree/main/tests

[67] https://auroral.iot.linkeddata.es/requirements/requirements-EV-charger.html

[68] https://github.com/oeg-upm/auroral-VehicleCharger-ontology/tree/main/tests