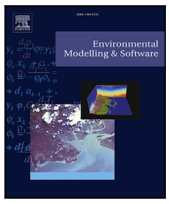
Environmental Modelling and Software 183 (2025) 106248

Contents lists available at ScienceDirect

Environmental Modelling and Software

journal homepage: www.elsevier.com/locate/envsoft



A conceptual data modeling framework with four levels of abstraction for environmental information

David Martínez [a], Laura Po [b], Raquel Trillo-Lado [c], José R.R. Viqueira [a,*]

[a] Departamento de Electrónica e Computación, Universidade de Santiago de Compostela, Santiago de Compostela, A Coruña, Spain
[b] Dipartimento di Ingegneria "Enzo Ferrari", Università degli studi di Modena e Reggio Emilia, Modena, Reggio Emilia, Italy
[c] Departamento de Informática e Ingeniería de Sistemas, Instituto de Investigación en Ingeniería de Aragón, Universidad de Zaragoza, Zaragoza, Zaragoza, Spain

ARTICLE INFO

Dataset link: https://data.europa.eu/, https://www.meteogalicia.gal, https://www.intecmar.gal

Keywords:
Conceptual data modeling
Environmental data
Data management
Meteorological data
Oceanographic data
Air quality data
Data integration

ABSTRACT

Environmental data generated by observation infrastructures and models is widely heterogeneous in both structure and semantics. The design and implementation of an ad hoc data model for each new dataset is costly and creates barriers for data integration. On the other hand, designing a single data model that supports any kind of environmental data has shown to be a complex task, and the resulting tools do not provide the required efficiency. In this paper, a new data modeling framework is proposed that enables the reuse of generic structures among different application domains and specific applications. The framework considers four levels of abstraction for the data models. Levels 1 and 2 provide general data model structures for environmental data, based on those defined by the Observations and Measurements (O&M) standard of the Open Geospatial Consortium (OGC). Level 3 incorporates generic data models for different application areas, whereas specific application models are designed at Level 4, reusing structures of the previous levels. Various use cases were implemented to illustrate the capabilities of the framework. A performance evaluation using six datasets of three different use cases has shown that the query response times achieved over the structures of Level 4 are very good compared to both ad hoc models and to a direct implementation of O&M in a Sensor Observation Service (SOS) tool. A qualitative evaluation shows that the framework fulfills a collection of general requirements not supported by any other existing solution.

Software availability

Software All the SQL code used for the implementation and evaluation of the current prototype of the present framework is available in the following url: https://github.com/cogradeusc/envdamof.

1. Introduction

The observation and prediction of the conditions of our environment is a keystone for the progress of many scientific disciplines. Scientists have been developing infrastructures that generate and store huge amounts of environmental data. The data storage and access subsystems range from very simple tools developed in the scope of specific research projects to very complex data hubs that integrate the deluge of information generated by sophisticated pubic data generation infrastructures and large scientific communities. Examples of such complex data hubs are the Copernicus Data Space Ecosystem,[1] the Global Earth Observation System of Systems (GEOSS) (Nativi et al., 2015), the NCAR Research Data Archive,[2] the NOAA OneStop portal[3] and the Hydroshare online collaboration environment.[4] Environmental datasets have critical importance for users with very specialized and high skills of science and engineering, however, they are also the basis over which general services for citizens may be developed (Viqueira et al., 2020).

Environmental data infrastructures must provide data storage structures and data discovery and access mechanisms. The design and implementation of the data model has much impact in the data storage and access efficiency and also in the data discovery efficacy. A data model must contain all the required data and metadata to implement effective data discovery. At the same time, the structures should be designed bearing in mind the relevant queries that will be implemented

* Corresponding author.
E-mail addresses: david.martinez.casas@usc.es (D. Martínez), laura.po@unimore.it (L. Po), raqueltl@unizar.es (R. Trillo-Lado), jrr.viqueira@usc.es (J.R.R. Viqueira).

[1] https://dataspace.copernicus.eu/
[2] https://rda.ucar.edu/
[3] https://data.noaa.gov/onestop/
[4] https://www.hydroshare.org/

https://doi.org/10.1016/j.envsoft.2024.106248
Received 16 May 2024; Received in revised form 10 October 2024; Accepted 11 October 2024
Available online 18 October 2024

in the data access mechanisms, to enable an adequate efficiency, even when the size of the datasets increase with time.

Conceptual modeling frameworks such as the Unified Modeling Language (UML) enable the definition of data structures using classes, properties of classes and different types of associations between classes. Classes are used to model sets of entities of the application domain (instances of the class). Thus, the city of "Santiago de Compostela" may be an instance of a class *City*. The model that defines the mechanisms available in a modeling framework is called a metamodel (model of the model) (Gonzalez-Perez and Henderson-Sellers, 2008). Model elements are instances of the metamodel. Thus, for example, class *City* will be an instance of metaclass *Class* and a property "name" of class *City* will be an instance of metaclass *Property*. Instances of the model (the data) are usually recorded using data storage technologies with specific data storage models (for example databases with the relational model). On the other hand, instances of the metamodel (the metadata) are usually recorded in catalogs (implemented also with data storage technologies), and are of key importance for data discovering tasks. Generalization/specialization associations between classes enable the definition of hierarchies where subclasses inherit the properties and associations of superclasses. As an example, a superclass *PopulationCenter* may be specialized in two subclasses, *City* and *Village*. Superclasses are abstract when they may not have instances. Models containing only superclasses are called abstract models and may be specialized for various specific purposes, enabling the reuse of data model structures.

Designing a good ad hoc model for each specific dataset is costly, however it is clearly the best approach to achieve efficient solutions in terms of data storage space and query response time. Currently, two broad data storage models are used for the development of such craft model based solutions for environmental information: the Unidata Common Data Model (CDM) and database models. Unidata CDM and its corresponding NetCDF (Network Common Data Form) file format[5] are broadly used to represent remote sensing observations and environmental model outputs, whose shape has the form of some grid of spatial and temporal dimensions. On the other hand, database models, most commonly either the relational model (Codd, 1970) or some extension of it, are broadly used for datasets generated by in-situ observation infrastructures. Such a craft model approach also has some important drawbacks. First, the cost of having to develop a new model for each specific dataset is high in terms of human resources. Besides, the quality of each model depends completely on the skills of the designers. Finally, designing independent models for different applications creates a level of heterogeneity that hinders the integration of information obtained from different datasets.

To address the limitations of ad hoc solutions, several generic models for environmental data have been proposed in the literature. Defining a single, universal data model for all applications is impractical due to the diverse nature of environmental data needs. However, models for specific application domains have been developed and applied successfully (Horsburgh et al., 2008; Mason et al., 2014; Abdallah and Rosenberg, 2019). The Open Geospatial Consortium (OGC) has established the Observations and Measurements (O&M) data model, which provides a framework for environmental observations (Cox, 2013). A key strength of O&M is its extensibility, allowing it to be tailored to the specific requirements of various applications. Numerous profiles specialized for different contexts have been developed based on O&M (Taylor et al., 2013; Wojda and Brouyère, 2013; Horsburgh et al., 2016; Blodgett et al., 2021).

O&M has been extensively used alongside web services such as the OGC Sensor Observation Service (SOS) to support the dissemination of environmental data (Bröring et al., 2012). Solutions like 52° North,[6] istSOS,[7] and PySOS[8] provide implementations of the SOS standard; however, these technologies often encounter performance challenges when working with large datasets, particularly regarding data retrieval and query efficiency (see Section 7.2).

An effort to specify standard ways to store environmental data in the Unidata CDM is done with the specification of the Climate and Forecast (CF) conventions (Eaton et al., 2023). CF also defines a standard vocabulary for various data model elements (properties, units of measure, etc.). A generic data modeling approach based on semantic web standards, called NGSI-LD, has been proposed for context sensor data on the internet of things and smart city areas (ETSI, 2023). Neither CF nor NGSI-LD represent all the basic concepts related to environmental observation, which are defined in the O&M model. In general, a review of existing environmental data modeling solutions shows a big challenge in achieving both completeness in covering all the requirements of all applications and efficiency in data storage and query (see Section 7.3).

In this paper, a framework for the conceptual modeling of environmental data is proposed. The framework defines data structures for the representation of the main concepts that arise in the scope of both observation and modeling infrastructures. Those structures may be specialized to incorporate the requirements of different domains and specific applications. Four levels of abstraction are considered in the framework. The first two levels provide, respectively, with a generic data modeling approach and with a generic solution for environmental information. In the third level of abstraction, general data models for specific application domains may be defined specializing the structures of the previous level. Models for specific applications are defined at the fourth level of abstraction, once again reusing the structures from the previous level. The extensibility of the framework enables its use in any environmental application domain. In spite of its broad scope, the evaluation of the framework has shown in general a very good performance in terms of query response time. The proposed framework evolves from the TAQE data modeling framework (Martínez et al., 2022) defined for traffic and air quality monitoring.

Based on the above, the main contributions of the paper can be summarized as follows.

- A complete set of requirements for the conceptual modeling framework are proposed, that were extracted from an extensive review of existing solutions and from the experience of the authors in projects.
- An abstract data model specialized on environmental applications largely based on O&M and a metamodel with support for multiple vocabularies.
- An illustration of the use of the framework to define specialized abstract data models in two application areas: (i) climate science and (ii) traffic and air quality monitoring in smart cities (already considered in TAQE). Eight specific use case datasets of the above areas were modeled and implemented with the framework.
- An efficient implementation of the data structures generated by the framework based on PostgreSQL, PostGIS and schema-less aggregates encoded in JSON data types. A performance evaluation shows that the implementation is in general as efficient as good ad hoc models designed in existing organizations and outperforms in most cases a reference direct implementation of the OGC O&M model used by a SOS tool. Different types of datasets were used in this evaluation, including simple data values obtained in static observation stations, trajectories generated by mobile platforms and vertical profiles obtained at specific locations in the sea.

[5] https://doi.org/10.5065/D6H70CW6

[6] 52° North Initiative for Geospatial Open Source Software GmbH https://52north.org/software/software-projects/sos/

[7] Sensor Observation Service for Water Information Systems https://istsos.org/

[8] https://github.com/manuGil/py4sos

- A qualitative evaluation of the fulfillment of the requirements posed for the present framework. Other twelve data modeling solutions were also evaluated with respect to the same requirements, showing that none of them achieves fully all of them.

The remainder of this paper is organized as follows. In Section 2, related work and existing approaches are reviewed in detail, paying special attention to the OGC O&M data model and data storage models based on it. The general framework structure and the set of requirements assumed for its design and implementation are described in Section 3. The data models and metamodels defined for the two first levels of abstraction are described in Section 4. In Section 5, a generic data model for climate science applications is proposed. Two of the use cases implemented with the framework are described in Section 6. The results of the evaluation of the framework are shown in Section 7, including an illustration of data integration among different datasets, a quantitative evaluation of the query performance and a qualitative evaluation of the fulfillment of the proposed requirements. Finally, some conclusions and lines of further work are depicted in Section 8. The paper is completed with an appendix (Appendix) that provides additional models and use cases related to traffic and air quality monitoring in smart cities.

2. Related work

Models are abstractions of real systems that are key artifacts during engineering of software products (Brambilla et al., 2017). Various data modeling paradigms are used at different levels of abstraction in the area of Data Management. Two main paradigms are used at the conceptual level: (i) Models based on entities (objects) and relationships (associations) among them (Chen, 1976; Blaha and Rumbaugh, 2005) and (ii) models based on dimensions and measurable facts (dimensional modeling) (Kimball and Ross, 2002). At a lower level of abstraction, currently, most applications still rely on implementations based on the relational model (Codd, 1970). However, in some specific cases, non-relational (Sadalage and Fowler, 2013) paradigms provide good performance, by supporting complex nested data types (aggregates) and large scale distributed architectures. All those non-relational models rely also in the lack of predefined schema, which brings advantages at data insertion and disadvantages at data querying. In particular, as applications cannot query the database catalog to get the schema, the schema must be encoded in the applications code, which is not the best place to be. Furthermore, the database cannot use the information of the schema to perform query optimization (Sadalage and Fowler, 2013).

Extensions to support both complex data types and distributed architectures are currently available for relational DBMSs, enabling their use with different data models and configurations. An example is the PostgreSQL DBMS[9] and its Citus Data extension[10] for distributed databases. However, the management of scientific arrays is still nowadays not efficient in DBMSs, thus, relevant applications have to use either specific file formats (Devys et al., 2019; The HDF Group, 2024) or specific array DBMSs (Baumann, 1994; Brown, 2010). A final general purpose data model paradigm is the Resource Description Framework (RDF) (Cyganiak et al., 2014) used in the semantic web and linked data scope.

Conceptual object-based and dimensional models are also used in the area of geospatial data management (Rigaux et al., 2001; Viqueira et al., 2005). Entities types (objects classes) of object-based models are called *Feature Types* (Kottman and Reed, 2009), and they may include geometric properties (Herring, 2020) to represent their location and shape on the Earth surface. Measures that change over geospatial and temporal dimensions are modeled using collections of mappings called Coverages (OGC, 2007). Feature Types and Coverages with sparse spatial domains are efficiently managed with either relational or non-relational approaches. On the other hand, dense coverages are usually represented with arrays of spatio-temporal dimensions, whose size might be very large. Their efficient management requires therefore the array specific solutions mentioned above. Few works have attempted the uniform management of Features and Coverages (Villarroya et al., 2016; de Bakker et al., 2017), and thus, relevant mature and efficient implementations have not been reached yet.

An important milestone towards the definition of a general conceptual data model for environmental observation data was the proposal of the Observations and Measurements (O&M) standard data model by the OGC (Cox, 2013). This conceptual model defines the main concepts involved in the representation of environmental observations (see Section 2.1 for more details). The O&M data model is a key part of standard interfaces defined by the OGC to access observation data through the web, such as the Sensor Observation Service (SOS) (Bröring et al., 2012) and the sensing part of the SensorThings API (Liang et al., 2021). Various tools for environmental data warehouses that implement the OGC SOS data access interface are currently available: 52° North SOS,[11] istSOS[12] and PySOS[13] are representative examples (see Section 2.1 for more details). O&M is also at the core of the Semantic Sensor Network (SSN) ontology (Haller et al., 2017; Compton et al., 2012) proposed by both OGC and the World Wide Web Consortium (W3C). SSN was used at the core of a generic model for the mediator component of a semantic integration federated architecture for environmental observation datasets (Regueiro et al., 2017).

Over the past decade, the hydrological research community has made significant progress in data modeling. Representative outputs of such effort are the Hydroshare infrastructure (Tarboton et al., 2024) and the WaterML profile (Taylor et al., 2013) of the O&M data model. One of the first solutions proposed was the Observations Data Model (ODM) (Horsburgh et al., 2008), designed for the storage of the data and metadata of in-situ observations of monitoring sites in a relational database. The object-oriented H^g_2O model (Wojda and Brouyère, 2013) for ground water data specializes the O&M model to include specific feature types for features of interest and sampling features. It includes also new structures for simple and complex observation types. The VOEIS Data Model (VODM) (Mason et al., 2014) extends ODM with data streams, datasets, users, roles, memberships, metatags and site data catalogs. The relational model proposed in Kim et al. (2015) is based on a geodatabase and enables the recording of both river observations and simulations. ODM2 (Horsburgh et al., 2016) may be considered a profile of O&M and it enables the modeling of discrete Earth observations, i.e., those that record a single value for the whole FOI (it does not support coverages as observation results). The relational Water Management Data Model (WaMDaM) (Abdallah and Rosenberg, 2019) restricts also to discrete observations and it was designed bearing in mind the following principles: modularity and extensibility, incorporation of networks of nodes and links as features, support for scenarios and version control, reusable context metadata, support for multiple data types, semantics specified with controlled extensible vocabularies, direct access support to subsets of data and metadata and open-source software environment. The mainstems (Blodgett et al., 2021) data model describe hydrologic networks using feature types proposed in part 3 of WaterML. Finally, the framework proposed in Salas et al. (2020) for open data and open modeling uses agents to integrate data obtained from different models.

Regarding oceanographic data, international hubs such as the U.S. Integrated Ocean Observing System (IOOS),[14] the European Marine Observation and Data Network (EDMODnet)[15] and the Copernicus Marine

[9] https://www.postgresql.org/
[10] https://www.citusdata.com/
[11] https://52north.org/software/software-projects/sos/
[12] https://istsos.org/
[13] https://sourceforge.net/p/pysos/
[14] https://ioos.noaa.gov/
[15] https://emodnet.ec.europa.eu/

Service (CMEMS),[16] rely mainly in the Unidata Common Data Model (CDM), implemented with NetCDF files to represent and store both in-situ and remote-sensed observations and model results. The Climate and Forecast (CF) convention (Eaton et al., 2023) is used to specify how NetCDF is used to represent specific dataset types and also to provide a standard vocabulary for observed and modeled properties.

Unidata CDM and the NetCDF standard is also the keystone for the representation of most dataset types for observation and modeling in climate science and meteorology. An interesting recent contribution in this area is the taxonomy of features for inputs and outputs of numerical models proposed in Harpham (2020). The taxonomy is based on standard spatial geometry types (Point, Multipoint, Polyline, etc.), where spatial grids and meshes are treated as specializations of Multipolygons. Temporal variations enable the modeling of data whose spatial fingerprint is defined by a single geometry, but it varies over time. Spatio-temporal variations support the changing in both value and spatial fingerprint at each time instant, i.e., it enables the modeling of tracks of different types (pointtrack, polylinetrack, etc.).

Environmental data is very important in the scope of smart cities and internet of things. In these areas, the European Telecommunications Standardization Institute (ETSI) has released a suite of specifications called NGSI (Next Generation Service Interfaces), which includes the NGSI-LD context information model (ETSI, 2023). The model is based on the W3C semantic web standards and it includes three layers: (i) a Meta Model defined on top of RDF/RDFS concepts, (ii) a Cross Domain Ontology that incorporates temporal and geographical properties and (iii) Domain Specific Ontologies that restricts the previous layer to a specific application domain and defines this way the specific structure and vocabulary. W3C SSN (Haller et al., 2017; Compton et al., 2012) could be incorporated as a new layer between the Cross Domain Ontology and Domain Specific Ontologies related to sensor observation.

Few efforts may be identified to achieve a uniform data model for air quality and traffic data in the context of smart city infrastructure development. The ontology proposed in Oprea (2009) was designed to model air pollutant concentrations at monitoring stations, together with their potential pollution sources. QBOAirbase (Galárraga et al., 2017) was built as a reduced version of the Airbase dataset provided by the European Environmental Agency (EEA), which gives access to pollutant concentrations registered at environmental stations through Europe. QBOAirbase has a dimensional model consisting of three dimensions (year, station and sensor) and measurements for different pollutants. The database is linked to other semantic web RDF data sources and it incorporates data provenance information by leveraging the use of the Prov-O ontology (Lebo et al., 2013). In Fernandez et al. (2016), the authors define an ontology to support intelligent transportation systems, which incorporates mechanisms to model vehicles, road infrastructure and sensors for traffic monitoring. Air quality prediction is an active research topic in the environmental modeling area (Johansson et al., 2022; Pisoni et al., 2024). However, specific data modeling solutions have not been proposed, to the best of these authors knowledge.

The TAQE data modeling framework (Martínez et al., 2022) considers four levels of abstraction to define data models for traffic and air quality at both local and regional scales. Data abstraction levels range from the completely general purpose level 1 to the completely specific level 4 of applications. Level 2 restricts to environmental applications and it is based on OGC O&M. Level 3 provides generic models for some application domains (air quality and traffic in this case). The solution proposed in the present paper evolves from that of TAQE, therefore, both proposals are completely compatible. The main difference is that TAQE was designed for traffic and air quality whereas the present model is more general. The level 3 model of TAQE includes remote sensing data sources, not considered in the relevant level 3 model shown in this paper, due to the focus in smart cities, i.e., only in the local scale for which remote sensing is not provided jet sufficient resolution. Main contributions of the present work with respect to TAQE are the following: (i) The O&M model considered in TAQE is now extended to provide uniform representations of observation subsamples and time evolving process properties. (ii) A metamodel based on the previous model is defined that supports the definition of data types and the incorporation of multiple vocabularies in the catalog, (iii) a data model of level 3 for climate science was provided for illustration purposes, (iii) use cases with meteorological and oceanographic data of common use were also tested, (iv) an efficient implementation with aggregate structures based on PostgreSQL, PostGIS and JSON was provided and (v) a detailed evaluation, which is both qualitative comparing to other solutions and of query performance was undertaken.

2.1. Environmental data storage implementations based on O&M

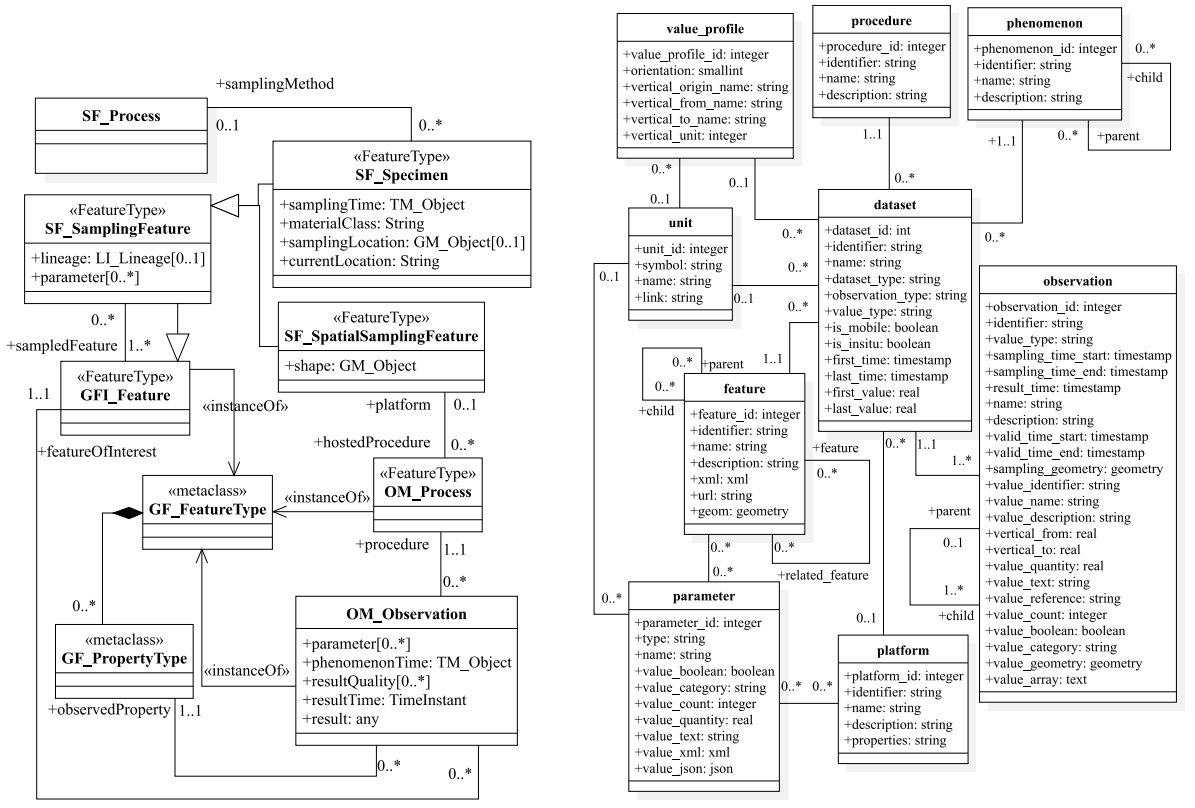
The main concepts considered by the OGC Observations and Measurements (O&M) conceptual model (Cox, 2013) are shown in the UML class diagram of Fig. 1(a). Instances of class *GFI_Feature* are used to model observed features, whose *Feature Types* are instances of metaclass *GF-FeatureType*. An example of an observed feature is the Spanish region of Galicia, and its feature type might be "Region". Properties of observed feature types, such as temperature and rainfall are represented by metaclass *GF_PropertyType*. In many cases, features may not be directly observed, at least in their whole extent, and some kind of sampling has to be performed. Class *SF_SamplingFeature* provides support for the representation of the different types of features relevant for those samplings. In particular, class *SF_SpatialSamplingFeature* supports the representation of samplings of the spatial extension of the observed feature. An example is the collection of locations of a network of meteorological stations that sample a specific region. Class *SF_Specimen* enables the representation of specimens captured to observe through them the ultimate observed feature. An example of a specimen is a bottle of water obtained from a specific location in a river, that is analyzed in a laboratory to get values of observed properties of the river. Class *OM_Process* is included to incorporate metadata of the processes used to generate observations. Finally, class *SF_Process* enables the description of the method used to capture the instances of *SF_Specimen*.

An observation (*OM_Observation*) records the value generated for the observed property (*result*) and it references its observed feature (*FeatureOfInterest*), its observed property and the process used to generate it (*procedure*). Mandatory temporal data of the observation consists of two elements: (i) the *phenomenonTime*, which records the time instant or time period during which the result applies to the property of the observed feature and (ii) the *resultTime* that records the instant when the procedure generated the observation. Optionally, an observation may record additional parameters, result quality metadata, and other types of general metadata. The result of an observation may be very simple, such as an integer value that represents some count. However, complex results are also possible, such as time series of records and different types of geospatial coverages.

An unusual characteristic of the O&M conceptual model is that it combines elements at both model and metamodel levels. A metamodel is a model whose instances are specific models (Gonzalez-Perez and Henderson-Sellers, 2008). Instances of the metamodel are usually recorded in metadata catalogs. Thus, classes *GFI_Feature*, *OM_Process* and *OM_Observation* are instances of metaclass *GF_FeatureType*. In Fig. 1(a) it is shown how the O&M model defines an association between class *OM_Observation* and metaclass *GF_PropertyType*. This unusual characteristic will lead to potential performance problems when specific models based on O&M are implemented.

Fig. 1(b) shows the main data storage structures of a representative implementation of the O&M data model developed to support a general purpose sensor observation server with a Sensor Observation Service

[16] https://marine.copernicus.eu/



(a) Main classes of the O&M conceptual model

(b) Main data storage structures of a representative SOS server implementation

Fig. 1. Illustration of a general purpose data storage model based on OGC O&M.

(SOS) (Bröring et al., 2012) data access interface. Such an implementation model enables the storage of environmental observation data in any application area with the same set of data storage structures, which eases data integration. Classes *feature*, *procedure* and *phenomenon* are used to represent, respectively, observed features, data generation processes and observed properties. *Parameter* enables the recording of feature properties that are not observed. Class *dataset* is incorporated in the model to provide support for complex observations, such as time series, vertical profiles and trajectories, using flat (non-nested) data structures. Observations are recorded in models like the one of Fig. 1(b) in SOS tools like 52° North SOS, istSOS and PySOS.

A main problem of this type of generic data storage solutions is that data query over those structures does not offer the required performance in many cases, as it will be shown in Section 7.2. Two main characteristics of the model are behind those performance issues: (i) The recording in the model of metamodel elements. In particular, observed and non observed properties and their relationships with observed features and observations are recorded as data items and not at part of the schema in the system catalog and (ii) the use of flat non-nested structures to represent components of complex observation results that are usually retrieved as a whole. The framework proposed in this paper avoids the two above characteristics while providing a general solution for environmental data storage.

3. Requirements and general framework structure

The framework proposed in this paper incorporates data models at four levels of abstraction.

Level 1: The UML object-oriented metamodel.

Leve 2: A metamodel and an relevant abstract data model for geospatial and environmental applications.

Table 1
Generic Requirements.

Requirement	Description
GR01	The framework must be flexible to be easily adaptable to different applications and application domains, enabling the definition of application specific structures and the reuse of common data modeling structures among them.
GR02	The framework must support the incorporation of controlled vocabularies, such as the Standard Names of the CF convention (Eaton et al., 2023) and the compliance with relevant content specific standards such as O&M and WaterML (Taylor et al., 2013).
GR03	The framework must provide models of different levels of abstraction to support the integration of data sources generated in different applications. Each model combines general elements inherited from the model of the previous level with specialized structures designed for the application area objective of the current level.
GR04	The framework must support the creation of data models that may be efficiently implemented, both in terms of data storage size and query response time, allowing the use of different scalable data management technologies in such implementation.

Level 3: Abstract data models specialized in different environmental application areas.

Level 4: Data models for specific applications inside each application area.

The methodology followed to design the above levels of the framework is described now. First, a wide collection of requirements was extracted from the review of related approaches in the literature and from the experience of the authors in various projects. A discussion on the evaluation of the fulfillment of such requirements by proposed framework and by other approaches is provided in Section 7.3.

Table 2
Level 1 requirements: General purpose data modeling functionality.

Requirement	Description
L1R01	The model must support the representation of objects (entities) and their properties.
L1R02	The model must support an extensible data type system for the values of the object properties. Primitive data types (integer, string, etc.) must be directly supported and they might be extended by user defined data types, including enumeration data types and data types with complex structure.
L1R03	The model must support the representation of associations (relationships) between objects.
L1R04	The model must support the classification of objects with the same properties and relationships into classes (entity types).

Table 3
Level 2 requirements: Geospatial and Environmental data modeling functionality.

Requirement	Description
L2R01	The model must provide data types for geometric objects, intervals of real values and temporal periods.
L2R02	The model must provide structures for the representation of Coverages with geospatial, temporal and vertical dimensions. Points, lines and surfaces must be supported in the geospatial dimensions, enabling multipoint, multicurve and multisurface coverages.
L2R03	The model must provide data types to represent measures and categories. A measure combines a real value with a unit of measure. A category combines a keyword with a reference to a vocabulary of possible values.
L2R04	The model must support the representation of data generation processes (including observation processes and modeling processes). The representation of the evolution with respect to time of the process properties must be supported, to be able to identify the conditions that applied during the generation of a specific result.
L2R05	The model must support the representation of objects generated during the spatial sampling of the features of interest. Examples of spatial samples are points, profiles, trajectories, scenes, swaths, etc.
L2R06	The model must support the representation of specimens collected to sample the features of interest. The processes used to capture those specimens should also be represented by the model.
L2R07	The model must represent the temporal context of the generated data, including the result and phenomenon time considered in the OGC O&M data model. Result time represents the instant when the process generated the data. Phenomenon time is related to the time when the generated data applies to the feature of interest.
L2R08	The model must support simple data values and also complex values resulting from spatial, vertical and temporal subsampling. Complex values include time series, vertical profiles, trajectories and coverages.

The generic requirements that guide the overall design of the framework are shown in Table 1. Broadly, according to those requirements the system must support the integration of different applications, but adapting to their singularities, achieving at the same time good performance in data storage and querying. Data integration is achieved by the reuse of common abstract structures that are specialized for different applications and by sharing common and standard vocabularies for property names. Data integration is illustrated in Section 7.1. Query performance is evaluated in Section 7.2 with respect to ad-hoc models and to the generic data storage solutions described in Section 2.1.

Requirements that are independent of any application (see Table 2) were considered to design Level 1 of the framework. Thus, the UML metamodel was chosen at this level, incorporating objects, classes, properties and associations, including generalization/specialization associations and composition. UML composition, represented with a black diamond, is used to represent a strong form of "has a" relationship, which is materialized in the implementation with dependencies of existence and identification from the parts to their whole (as in weak entity types in the Entity-Relationships model). Therefore, cardinality in the side of class with the role of "whole" is always 1..1 and it is not represented in the diagrams.

Next, the requirements that are specific to geospatial and environmental applications, but are at the same time independent of any of those applications were considered to design a second level of abstraction. Level 2 requirements are shown in Table 3. To design this level, OGC and ISO standards were used as a baseline. In particular, this level is largely based on OGC O&M (Cox, 2013). Level 2 consist of a metamodel that defines metadata catalog structures and an abstract model that will be specialized in subsequent levels.

At level 3, various abstract data models may be defined to incorporate data structures that are common in different application areas. This level is illustrated in the present paper by providing a model that might be reused by may different climate science applications (See Section 5). Other examples of level 3 abstract models for traffic and air quality monitoring at local scale in the context of smart cities are shown in Appendix. In general, communities of experts of different application areas may reach agreements in the form of models of level 3, which will be reused in many applications of those areas. The existence of those models will ease the integration of the data generated by different applications in those areas.

Finally, data models that define the data structures used to record datasets in specific applications are incorporated at level 4. These specific models specialize and reuse abstract data structures from level 3.

Defining data models that are specialized at various levels is an approach already followed by OGC. Thus for example, the WaterML model specializes the more general O&M data model for applications in the area of hydrology. Users may use directly data encodings based on O&M and WaterML to represent their data. However, users might also specialize further WaterML for specific applications. It is argued by the authors of this paper that once the users are familiar with the concepts of models from upper levels, the formulation of new models for their applications is simplified by reusing elements of those models. Additionally, in general, the quality of the designs gets improved and the integration of data among applications is also facilitated.

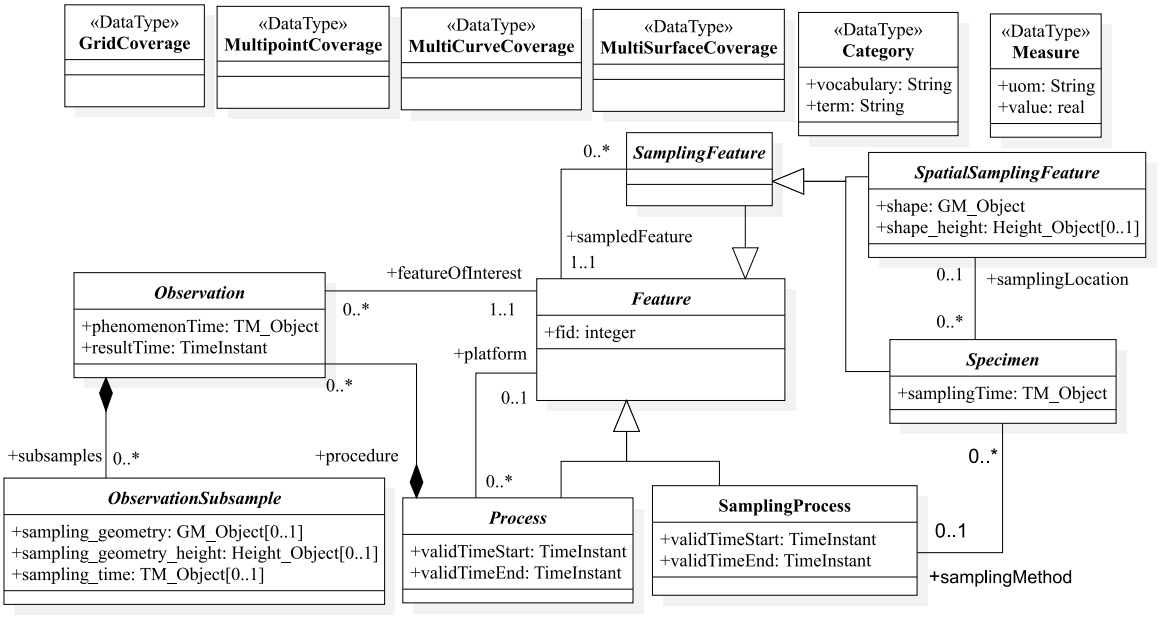
4. Level2: Abstract model and metamodel for environmental data

The abstract data model that contains the generic data structures for environmental data representation is described below in Section 4.1. Section 4.2 outlines the main characteristics of the current implementation of the data structures. Finally, the structures of the metadata catalog provided at level2 of the framework are shown in Section 4.3.

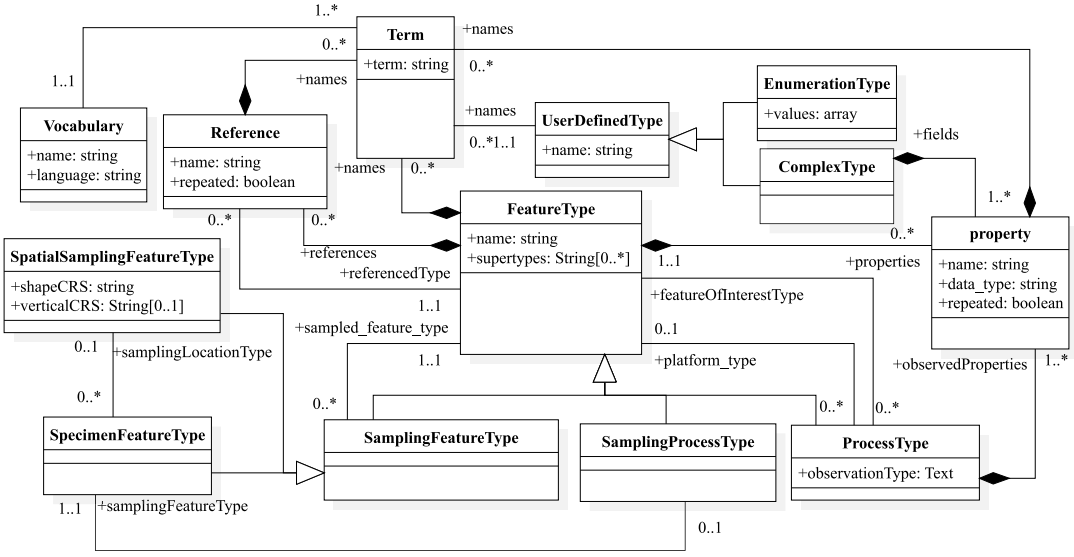
4.1. Abstract data model

The abstract data model of level 2 is shown in Fig. 2(a). It is noticed that this model is very similar to the O&M model already described in Section 2.1 (see Fig. 1(a)). Classes *Feature*, *Process*, *Observation*, *SamplingFeature*, *SpatialSamplingFeature*, *Specimen* and *SamplingProcess* are incorporated to support relevant concepts of O&M. The data types provided at the top of the figure are also supported by O&M and enable the representation of different types of observation results, including grid and discrete coverages, categories (terms available in vocabularies) and measures (real values with a unit o measure). The main differences between this abstract data model and O&M are resumed as follows.

- Observed properties are represented in O&M with associations between the observation and a metaclass of the metamodel. In the present framework, properties (either observed or not) are represented at level 2 only in the metamodel and thus available only in the metadata catalog (see Section 4.3). Values of observed properties will be recorded at level 4 either as values of properties of specific subclasses of *Observation*, or as values of properties of specific subclasses of *ObservationSubsample*, or as bands of coverages referenced in instances of some subclass of *Observation* .



(a) Level 2 Abstract Environmental Data Model



(b) Metadata Catalog Metamodel

Fig. 2. Level 2 abstract environmental data model and metadata catalog metamodel.

- The current version of the framework does not incorporate quality metatadata, which is optional in O&M.
- O&M enables the incorporation of a list parameter values in observations. In most cases, those parameters are used to record vertical coordinates and spatial and temporal properties of components of complex observations. The present framework does not provide support for generic parameters, but specific structures for temporal and geospatial subsamples of the observation are supported by class *ObservationSubsample*, which is not part of O&M.
- Vertical coordinates are represented in the present framework separately from geometric objects. Supporting height coordinates separated from geometric objects eases the integration with tools of the area of Geographic Information Systems (GIS). In some cases, it is needed to provide specific properties for different heights of the same geometric object (one example is a vertical profile). In these cases, supporting the height as a separate dimension helps in achieving more efficient solutions.
- Data types *Category* and *Measure* are parametric in the present framework. Thus, if the vocabulary or the unit of measure are specified during the declaration of the data type, then their values are recorded only in the metadata catalog. Otherwise, they have to be recorded with each observation result, as it is the case in O&M.
- Classes *Process* and *SamplingProcess* incorporate in the present model properties to represent periods of valid time (*validTimeStart* and *validTimeEnd*). The reason is that those classes use to have a highly dynamic nature, requiring the recording of evolution with respect to time of device configurations and/or model hyperparameters.

4.2. Data model implementation

The implementation of the current prototype of the framework is based on the PostgreSQL[17] database management system. PostgreSQL implements a relational model extensible with complex structures (aggregates). Those aggregates may be incorporated in the system in various different ways, including a combination of arrays and user defined types, XML and JSON. The support of XML and JSON structures transforms the underlying relational model to a hybrid SQL and NoSQL document-based model.

Geospatial data type implementation relies on the OGC and ISO standard data types (Herring, 2020) provided by the PostGIS extension.[18] Apart from the geometric objects, other complex values must be represented, including height intervals, time periods, coverages, categories, measures and subsamples. JSON structures are used to represent all those nested complex structures.

Coverage data types contain references to out-of-band representations of those coverages. In the current prototype of the framework implementation, grid coverages are stored in GeoTIFF files (Devys et al., 2019). Other types of geospatial coverages (multipoint, multicurve and multisurface) are recorded with tables in GeoPackage format (Yutzler, 2024). Temporal and height dimensions of coverages are represented in nested JSON structures under observation subsamples.

4.3. Metadata catalog structures

Catalog structures support the recording of metadata of each of the subclasses of the models defined at level 4. The schema of those structures is shown in the UML class diagram of Fig. 2(b). Class *FeatureType* records metadata of any feature type, including features of interest, processes, sampling processes and sampling features. For each feature type, the catalog records its name (which identifies the feature type), the set of properties, the set of references to other feature types, an optional set of names that denote the feature type in different vocabularies, and the set of supertype names. Supertypes are classes of some data model of Level 3. Each feature type of each specific application (Level 4), may be a subclass of one or more classes of Level 3. Tagging the feature type of level 4 in the catalog with the names of the superclasses provides general semantics that help in interpreting the semantics of its instances, easing the implementation of data integration applications, as it is illustrated in Section 7.1. Examples of these metadata are given in Section 6 for specific use cases.

Each *Property* of each *Feature Type* is identified by a name. Besides, the catalog records the name of its data type (*data_type*), an optional set of names that denote the property in different vocabularies and a boolean flag, called *repeated*, that points out whether the property has either just one value or various values.

The data type may be either a primitive one directly supported by the system or a user defined data type. Two classes of data types may be defined by users. An *EnumerationType* defines a set of possible text values for a property. On the other hand, a *ComplexType* defines an structure containing fields, each of them again with a name and a data type, which might be primitive or user defined. The names of the data types and fields obtained from vocabularies may also be recorded in the catalog.

A *Reference* represents a link between a source feature of a specific feature type to one or more destination features of another feature type. It enables the implementation of binary associations between feature types.

The metadata of each process type is recorded in subclass *ProcessType* of class *FeatureType*. In addition to the metadata already described for feature types, for each process type, the catalog records: (i) the name of the data structure of the model that records the observations generated by the process type (*observationType*), (ii) the name of the feature type that records the platform where the process is installed, (iii) the name of the feature type that records the FOIs of the generated observations and (iv) the set of properties that are observed by the process. It is reminded that, in spite of the use of the term "observation", process types may be used to represent observation and modeling processes.

Class *SamplingFeatureType* records metadata of each sampling feature type, including the name of the sampled feature type (the one recording the final FOI that is being sampled). Two types of sampling feature types are supported, a *SpatialSamplingFeatureType* to support the spatial sampling of the final FOI and a *SpecimenFeatureType* to support the sampling thought specimen capture. The names of the coordinate reference systems used for geospatial and vertical coordinates are recorded as metadata of the spatial sampling feature type. Regarding specimen feature types, the catalog records the name of the spatial sampling feature type that records the location where the specimen was captured. Besides, class *SamplingProcessType* records metadata of the processes used to capture the specimens.

Table 4
Level 3 requirement: Climate observation and modeling.

Requirement	Description
L3CS01	The data generated by observation infrastructures installed at specific locations, using either static or removable devices, and representing the height of the observed location whenever necessary.
L3CS02	The data generated along vertical profiles, vertical sections, linear transects and 3D trajectories, either at the ocean or at the atmosphere.
L3CS03	The data generated by remote sensing devices, including observations results for each point of a either regular or irregular grid. The data generation process might generate snapshots at specific or predefined scenes or data gusts following specific or predefined swaths.
L3CS04	The data generated by oceanographic and meteorological models, including nowcast, forecast and reanalysis systems.
L3CS05	The data generated by the indirect observation of environmental properties through the direct observation of captured specimens.

5. Level 3 data model for climate observation and modeling

The climate science application domain was chosen to illustrate and evaluate the framework due to the wide variety of different observation and modeling infrastructures available in this area. The requirements considered for our purposes are shown in Table 4.

The level 3 data model for climate observation and modeling applications is shown in the UML class diagram of Fig. 3. Processes for in-situ observation at specific locations are modeled as instances of subclasses of *CSInsituStaticProcess*. The observed locations (usually locations of environmental stations) are modeled with either class *CSSamplingLocation* or class *CSSamplingLocationHeight*, depending on whether the vertical offset has to be recorded or not. Requirement **L3CS01** is thus supported by the previous classes.

More complex types of in-situ observations are also supported by the model, to fulfill requirement **L3CS02**. In particular, classes are included to model process that generate vertical profiles, vertical sections, linear transects and trajectories. The spatial sampling feature type of a vertical profile *CSProfile* records the geosaptial location and the vertical interval. The observation result is complex (*CSProfileObservation*), and it has values of the observed properties at each vertical offset (*CSProfileObservationSubsample*). The shape of a vertical section is defined by a linestring and a vertical interval (*CSSection*). Each sample of its complex observation records the observed properties, a point inside the linestring and a vertical offset

[17] https://www.postgresql.org/
[18] https://postgis.net/

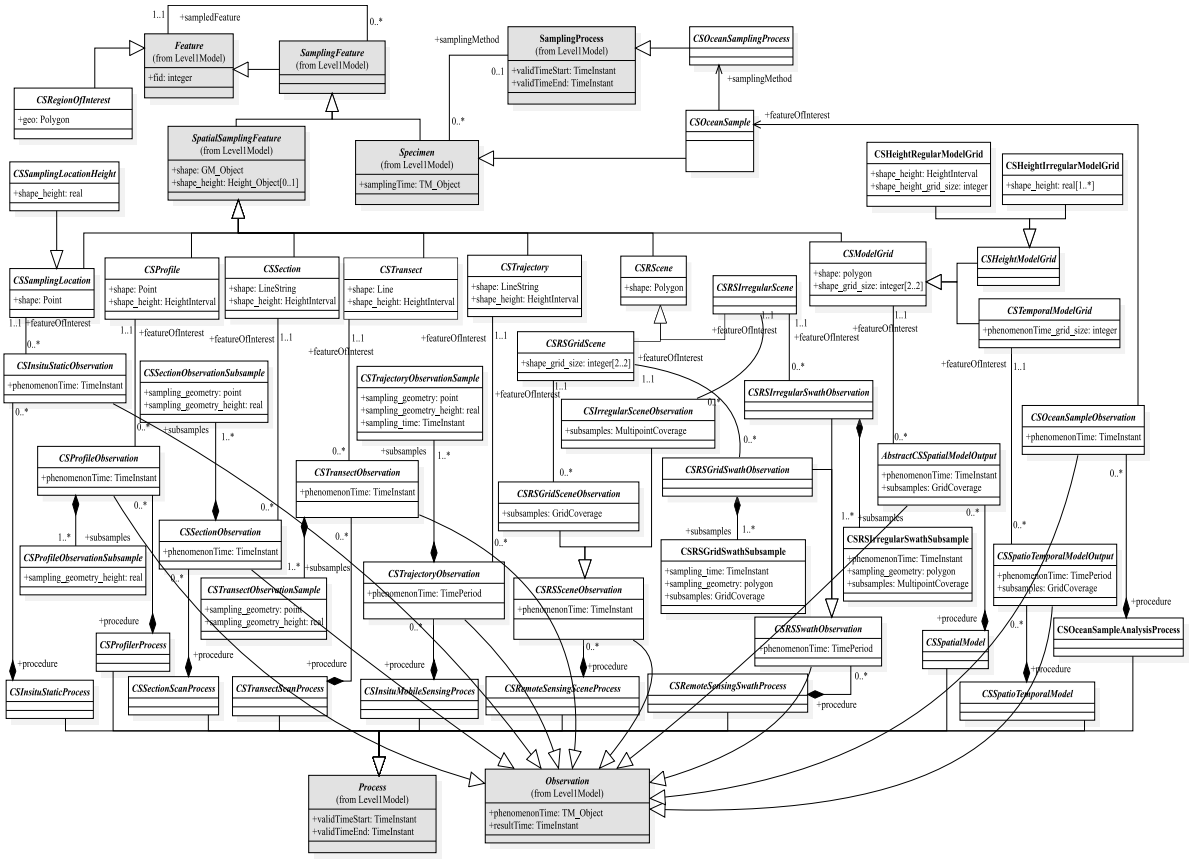


Fig. 3. Level 3 data model for climate science.

(*CSSectionObservationSubsample*). The shape of a transect is defined by a 3D straight line, stored with a line geometry and a height interval (*CSTransect*). The combined coordinates of the point and height recorded in each sample (*CSTransectObservationSample*) must lie inside the 3D straight line of the transect. Profiles, vertical sections and transects are associated to a single phenomenon time instant. Contrary to this, a trajectory is generated during a time period. Its shape is also a 3D line as in the case of a transect, but in this case, the line does not have to be straight. Thus, each sample (*CSTrajectoryObservationSample*) must record the point, the vertical offset and the specific phenomenon time instant, in addition to the observed properties.

Two types of remote sensing processes are supported by the model (**L3CS03**), namely *CSRemoteSensingSceneProcess* and *CSRemoteSensingSwathProcess*. The former generates observations at each location of a predefined or sporadic region, whereas the latter generates gusts of observations along the path defined by a spatial swath. Two types of spatial samplings are supported, regular grids (*CSRSGridScene*) and irregular samplings (*CSRSIrregularScene*). A *CSRemoteSensingSceneProcess* generates for each phenomenon time instant a complex result that is encoded as a spatial coverage. Grid coverages or multipoint coverages are used depending on the type of regular or irregular scene (see classes *CSRSGridSceneObservation* and *CSIrregularSceneObservation*). On the other hand, the observations generated by a *CSRemoteSensingSwathProcess* are even more complex. In fact, each observation, which applies to a period of phenomenon time, contains a time series of spatial coverages, whose combined geospatial fingerprint define a spatial swath. Therefore, each sample of each observation, records a coverage for a specific phenomenon time instant of the observed period (see the contents of classes *CSRSGridSwathSubsample* and *CSRSIrregularSwathSubsample*).

The model supports also the recording of results generated by environmental models (**L3CS04**). A spatial model (*CSSpatialModel*) generates an estimation of the values of the properties of interest for each cell of a spatial grid. The grid is always regular in the two geospatial dimensions (*CSModelGrid*). If the model generates estimations at different vertical offsets, then a height dimension must be added to the grid. Two types of vertical dimensions are supported. In a *CSHeightRegularModelGrid*, vertical offsets are placed at regular distances. On the other hand, the vertical offsets are not placed regularly in a *CSHeightIrregularModelGrid*. Spatial models are normally used to perform nowcasts, i.e., real time estimations of the properties at every discrete location of the region of interest. Forecast and reanalysis are supported by instances of some subclass of *CSSpatioTemporalModel*. Now, the output coverage must have also a temporal dimension, which is defined with a regular sampling at the observation phenomenon time period. Again, the spatial dimensions of the grid may include a vertical dimension, with a regular or irregular subsampling.

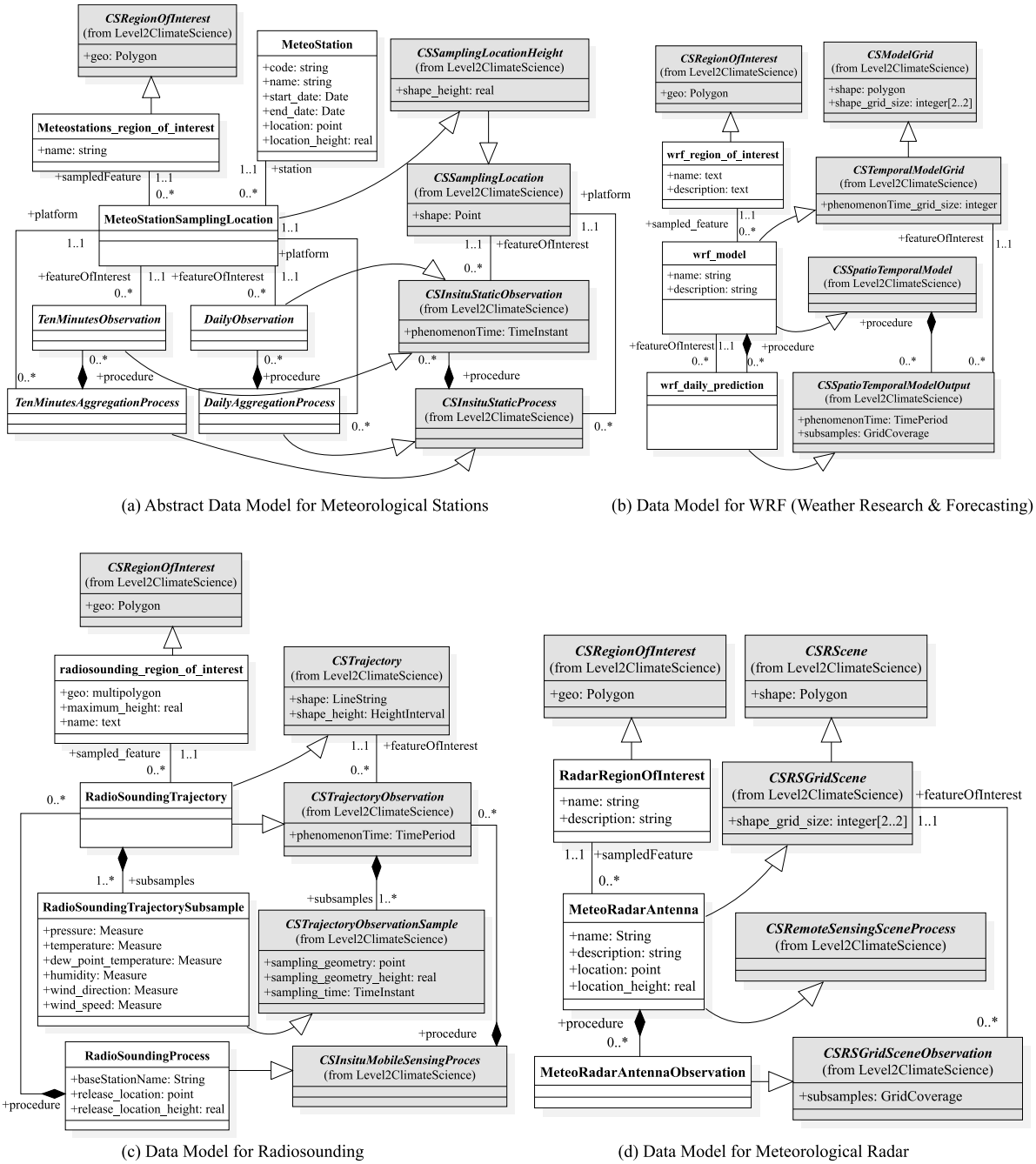
Class *CSOceanSampleAnalysisProcess* is used to support the observation thought the capture of specimens in the ocean (**L3CS05**). Examples of such specimens are samples of sea water obtained at specific locations and depths and samples of marine organisms fished at specific sea swaths. Collected specimens are modeled as instances of *CSOceanSample*. The process used to collect the specimens is modeled with class *CSOceanSamplingProcess*.

6. Use cases

In the following subsections, various data models of Level 4 are described to illustrate and evaluate the use of the framework in a variety of different applications.

6.1. Meteorological data in METEOGALICIA

Four specific Level 4 data models are described in this subsection to illustrate the use of the framework with meteorological data. All



(a) Abstract Data Model for Meteorological Stations

(b) Data Model for WRF (Weather Research & Forecasting)

(c) Data Model for Radiosounding

(d) Data Model for Meteorological Radar

Fig. 4. Level 4 data models for meteorological data in MeteoGalicia.

the models are used to represent data generated by MeteoGalicia, the meteorological agency of the Spanish region of Galicia.[19] These data include observations generated by a network of static meteorological stations, spatio-temporal grid coverages generated by a weather numeric prediction model, 3D trajectories of in-situ observations generated in radiosounding campaigns and spatial coverages of reflectivity observations generated by a meteorological radar.

The model shown in Fig. 4(a) may be specialized to record data generated by instruments installed in meteorological stations. The data of each station is recorded in class *MeteoStation*, whereas class *MeteoStationSamplingLocation* is used to model specific locations and heights inside the perimeter of the stations, where the instruments are installed. Observation process types that generate data with the maximum level of temporal resolution (10 min) are modeled as subclasses of *TenMinutesAggregationProcess*. Similarly, process types that generate daily aggregates are modeled as subclasses of *DailyAggregationProcess*. Respective observation subclasses of *TenMinutesObservation* and *DailyObservation* are also defined in the final model. Currently, 7 subtypes of each of the above parent process classes are defined in the model, whose observed properties are listed in Table 5. It is reminded that, in addition to the above data structures, the framework records also metadata in the catalog (see Fig. 2(b)) of these specific Feature Types and Process Types. Thus, for example, for each process type subclass of *TenMinutesAggregationProcess*, the catalog records its name, the set of properties, the observation class that records the generated observations, the platform type that records relevant platforms, the feature of interest type and the observed properties amongst other metadata items. The catalog records

[19] https://www.meteogalicia.gal/

Table 5
Properties observed by each type of meteorological process.

Process type	Ten minutes aggregate	Daily aggregate
Snow	snow height	snow height
Wind	wind direction, wind speed, wind gust direction, wind gust speed, wind direction standard deviation, wind speed standard deviation	wind speed, wind gust direction, wind gust speed, prevailing wind direction
Precipitation	rainfall	water balance, rainfall
Pressure	barometric pressure, sea level reduced pressure	barometric pressure, sea level reduced pressure
Solar radiation	sunshine duration, global solar radiation	insolation, sunshine duration, daily global irradiation
Tempera-ture/Humidity	relative humidity, mean air temperature, dew temperature	(maximum, minimum, mean) relative humidity, (maximum, minimum, mean) air temperature, dew temperature, evapotranspiration
Surface temperature	mean air temperature, soil temperature	mean air temperature, soil temperature

also the supertypes of each feature type. For example, for feature type MeteoStationSamplingLocation, the catalog records the supertype "cs_sampling_location_height". This way, applications know that each instance of this class must be interpreted as a sampling location at a specific height, which provides important semantics for the application and for its end users.

Fig. 4(b) shows a data model to support the recording of data generated by the Weather Research and Forecasting (WRF) model. The metadata of the model and the shape of the spatio-temporal grid is represented with class *wrf_model*. Model outputs are represented with instances of class *wrf_daily_prediction*. Each such instance contains a spatio-temporal coverage, with a spatial resolution of 1 km and with a temporal resolution of 1 h, that represents, at each spatial cell, the predicted values for wind direction and speed, air pressure at sea level, rainfall, relative humidity, snowfall amount, snow level, sea surface temperature and air temperature, for each hour during the next 4 days. It is reminded that, in the current implementation of the framework, a spatio-temporal grid coverage is represented as a series of spatial grid coverages, each of them recorded in a separate GeoTIFF file. The catalog records metadata for all the above feature types. Now for example, the supertypes of feature type *wrf_model* will record the array of values ["cs_temporal_model_grid", "cs_spatio_temporal_model"], as a *wrf_model* in this model is both a sampling feature that records the characteristics of the model grid and a process that records the model description.

Fig. 4(c) depicts the data model that supports radiosounding campaigns. The metadata of the observation process is recorded in class *RadioSondingProcess*, which includes the name of the base station from which the radiosondes are released and its geospatial and vertical coordinates. Each radiosonde generates a trajectory (class *RadioSoundingTrajectory*), which is both a spatial sampling geometry of LineString data type and a complex observation. Therefore, in the catalog, the supertypes field will record an array of values ["cs_trajectory", "cs_trajectory_observation"]. In each subsample of the trajectory, class *RadioSoundingTrajectorySubsample* represents the relevant geospatial and vertical location, the phenomenon time instant and the values of the atmospheric observed properties. It is reminded that these nested subsamples of the trajectory observation are implemented in the current prototype of the framework with a json substructure.

The data model of Fig. 4(d) enables the representation of the observations and related context data generated by a meteorological radar. The context data related to the data generation process and to the spatial sampling feature, i.e., the used geospatial grid, is recorded in class *MeteoRadarAntenna*. The grid coverages that are generated by the infrastructure every five minutes recorded in class *MeteoRadarAntennaObservation*.

6.2. Oceanographic data in INTECMAR

The use of the framework to model oceanographic data is illustrated with a couple of datasets available in INTECMAR,[20] the institute of the regional government of Galicia that is in charge of the control of the marine environment in this region of the northwest coast of Spain.

The first dataset consists of CTD (Current, Temperature, Depth) profiles generated at specific locations inside the Galician estuaries by periodical campaigns organized by INTECMAR. The Level 3 data model is shown in Fig. 5(a). The final observed feature of interest is the marine area of each observed estuary (see class *Estuary*). Each estuary is sampled at specific stations (class *CTDStation*), which are defined in the model as subclasses of the Level 3 *CSProfile*. The data is generated using CTD devices, which are moved to the sampling stations in ships. The hardware configuration of each CTD device and the ship in which it is installed at each campaign (period of time) is registered in class *CTDDevice*, which is a subclass of Level 3 *CSProfilerProcess*. Each generated vertical profile is represented by an instance of class *CTDObservation*, and values of the observed properties registered at each depth are recorded in class *CTDObservationSubsample*. Remember that the recording of those subsamples is implemented using JSON aggregates in the current prototype of the framework.

The model depicted in Fig. 5(b) enables the recording of the sea current coverages generated by a High Frequency (HF) Radar infrastructure. The hardware infrastructure is composed of a series of antennas located at strategic places along the coastline. Each antenna generates every hour a radial coverage of sea surface current velocities away from the instrument. The shape of the sampling grid of each antenna is a circular sector whose center is the antenna location. Observed subsamples are located in the circular sector at regular distances (ranges) and bearings. Those grids are irregular when they are represented using geographic coordinates, therefore the generated coverages are modeled as MultiPoint coverages in the model. The metadata of the antennas and of the relevant radial grids are represented in class *HFRadarAntenna*. Every hour, a data combination process merges the observations of all the antennas in the area to generate a combined observation of sea surface water velocities. Each version of this data combination process is modeled as an instance of class *HFRadarCombine*. The spatial shape of the result combined observation is a regular grid of geographic coordinates, whose metatadata is also recorded in the same class.

7. Evaluation

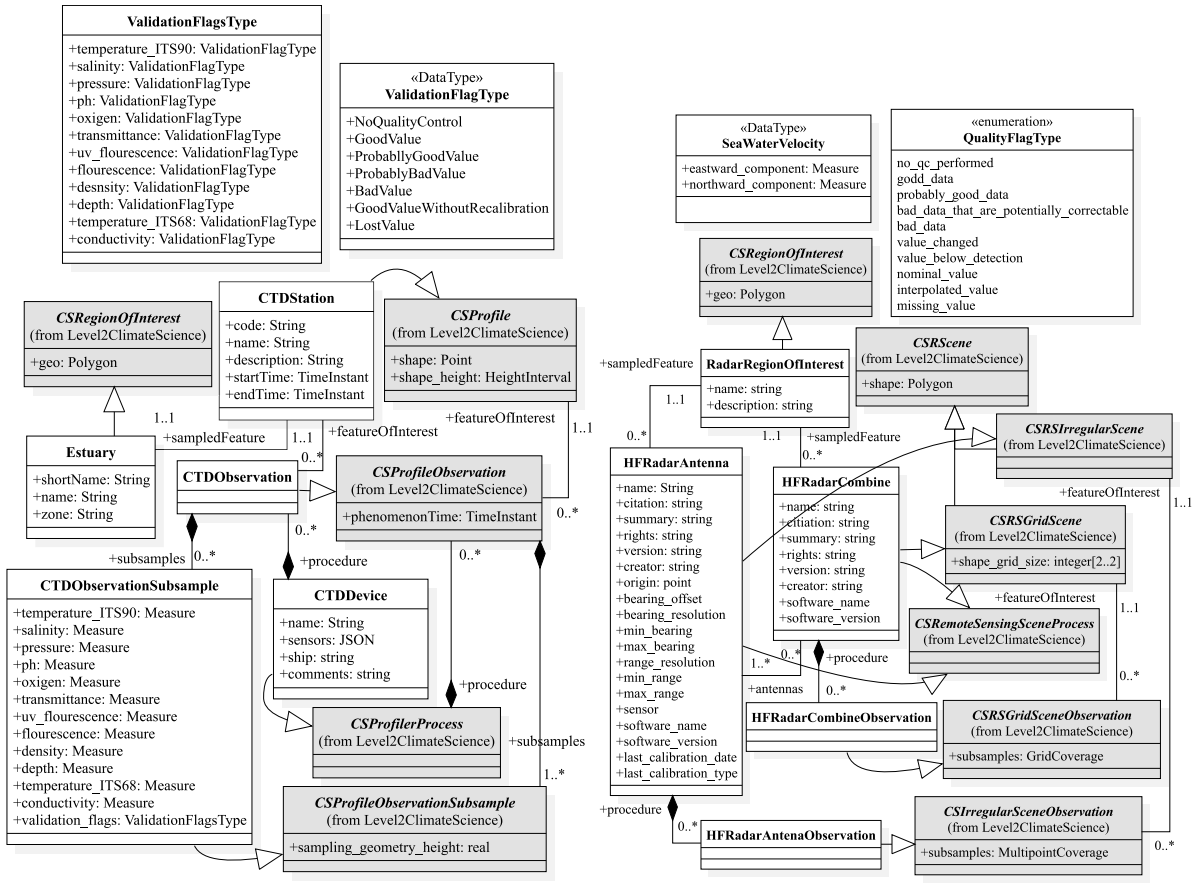
The evaluation of the framework is both quantitative, in terms of query performance, and qualitative, comparing the level of support provided for the assumed requirements with other approaches.

7.1. Illustration of data integration between applications

Let us consider a user that needs data of air temperature in a specific bounding rectangle R during a period of time from $t1$ to $t2$. A general purpose application could use the data structures of the present framework to access the required data. The sequence of tasks to be executed is the following.

1. Obtain the list of process types recorded in the frameworks catalog (see Fig. 2(b)), whose observed properties contain a property that has among its names for vocabulary "CF Standard Names" the value "air_ temperature". Considering the metadata of our uses cases, this query should retrieve the following process types:

[20] http://www.intecmar.gal/



(a) Data Model for CTD Profiles

(b) Data Model for Hight Frequency (HF) Radar

Fig. 5. Level 4 data models for oceanographic data in Intecmar.

radiosounding_process
temperature_humidity_10minutes_process
surface_temperature_10minutes_process
temperature_humidity_daily_process
wrf_model

2. After looking at the descriptions of the above processes, the user might choose to discard the *temperature_humidity_daily_process* because it generates daily aggregate data.
3. For the remainder process types, the system obtains their subtypes from the catalog. Thus, it is now aware that one of the process types is a "cs_insitu_mobile_process", other two are "cs_insitu_static_process" and the last one is a "cs_spatiotemporal_model". It obtains also the name of the *Observation* subclass that records the data for each process.
4. To access the data of the "cs_insitu_mobile_process", the system issues a query to the observation class, filtering by space using R and by time using the period $[t1, t2]$. Each observation is a trajectory that must be unnested to get the location, the phenomenon time instant and the temperature values. Again, those unnested elements must be filtered by space and time. Notice that the name of the field that records the temperature for this process type is also available in the catalog, and the names of the fields that record phenomenon time and location are standardized in the abstract model of level 2 of the framework.
5. To access the data of the two "cs_insitu_static_process", the system will query the observation table again filtering by space and time. The temperatures are directly obtained from a field of the observation table. The location is obtained from the shape of the feature of interest of the observation. Again, property names are described in the catalog and all the other fields and structures are standardized in level 2.
6. The observation table of the "cs_spatiotemporal_model' records references to spatiotemporal coverages. First, the system obtains all the references to coverages whose temporal period intersect with the period $[t1, t2]$ and whose feature of interest shape intersect with the query rectangle R. Depending on the system implementation of coverage access, the system may retrieve the whole coverage in some raster format such as NetCDF, or it might issue a new query to a specific data access service such as a standard OGC Web Coverage Service (WCS).

Notice that the use of standard vocabularies in the catalog enables the system to obtain the processes that observe air temperature, independently of the specific names that the observed properties might have in each dataset. Besides, the specialization of common abstract models of previous levels provides at some extent a common model template for all the applications. Having a common data model is requisite for integrated querying of different datasets, either in data warehouses or in federated systems (Levy, 2000). More precisely, the existence of those common abstract data models forces to have specific fields with specific names to record context and provenance metadata. Thus for example, the geometry of any spatial sampling feature will always be called "shape" in all the level 4 models. The combination of these common structures with the metadata available in the catalog enables application independent querying and therefore, it eases data integration among applications.

7.2. Performance evaluation

In general, it is expected that a data model designed for a specific application may reach a better performance than a model that is more

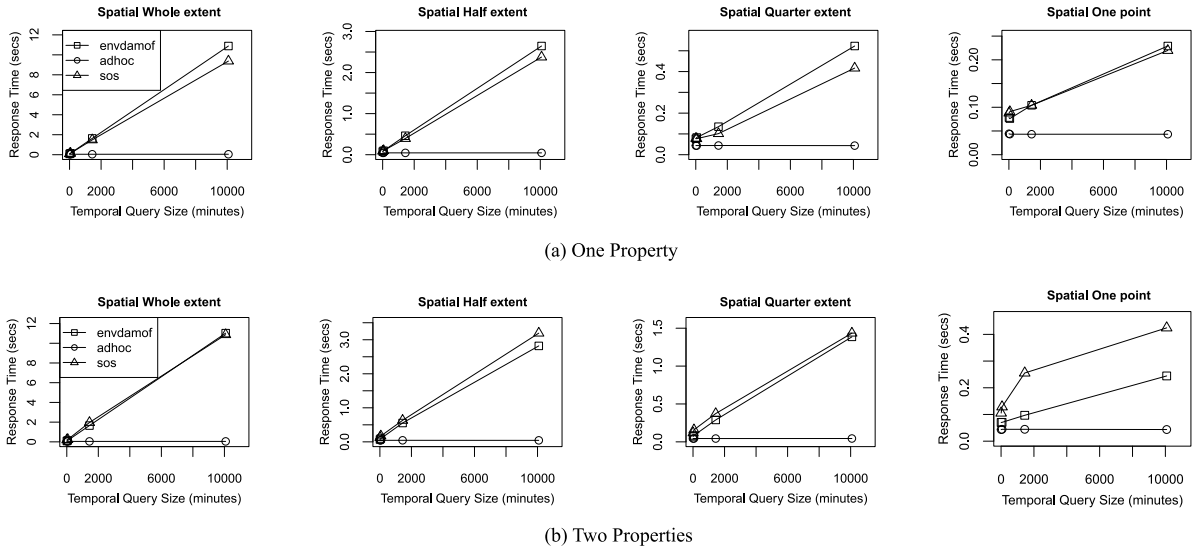


Fig. 6. Query performance: Sensor traffic observations.

generic and common to many applications. On the other hand, general approaches bring benefits in terms of development time and quality of the final solution. The hypothesis to check in the performance evaluation undertaken is that the current prototype implementation of the present framework, despite of being of general purpose, enables achieving a query performance similar to the one achieved by ad-hoc solutions, and in general, better than the one achieved by a reference implementation of an O&M data model in a Sensor Observation Service (SOS) tool. In particular, the query performance of the three following data modeling approaches was evaluated.

envdamof The current prototype of the environmental data modeling framework described in the present paper.

ad hoc A data model designed ad hoc for the relevant dataset, with the same characteristics of the one used in the source organization or project.

sos The relational implementation of the O&M data model provided by a SOS implementation (see Fig. 1(b)).

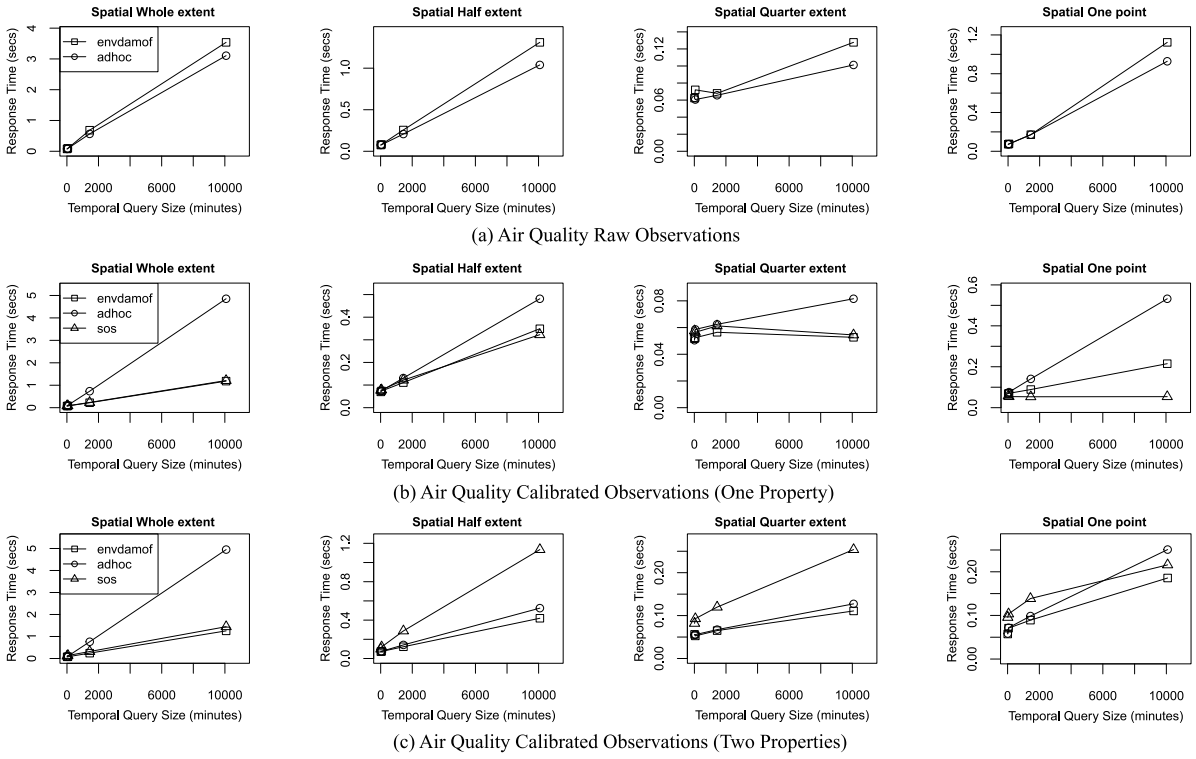
The hardware characteristics of the server used to perform all the experiments are the following: CPU 2 × Intel Xeon E5-2630 v4 (2,2 GHz 10c), 384 GB of RAM: 12 × 32 GB RDIMM 2400MT/s, 32 TB HDD: 8 × 4TB 7.2k SATA 6Gbps in JBOD. Version 15.3 of PostgreSQL with version 3.3 of PostGIS was used for the three approaches.

Various experiments were undertaken where many SQL queries were performed over different datasets stored in each of the above three types of data models. Indexing structures were created in all the datasets of all the models on identifiers of processes and features, geometric fields and temporal fields. Different types of queries were defined based on the selectivity of the spatial, temporal and vertical (when applicable) filter. Spatial and vertical ranges ranged from the whole extent of the dataset to just one spatial or vertical point. Temporal ranges ranged, in general, from just one time instant to 10k minutes of data. The result of each query includes one or various observed properties (depending on the query type), the identifier of the data generation process and the spatial and temporal context of the observation. Ten random queries were generated and executed for each combination of selectivity values (spatial, vertical and temporal selectivity). The average query response time for each type of query was computed after eliminating outliers (detected using the average and the standard deviation).

The first experiment used a subset of the traffic observations generated during the TRAFAIR project. The dataset contains more than 650k observations, each of them with two properties, traffic flow and occupancy. The query response times for each of the queries are shown in Fig. 6. The ad hoc implementation achieves a better performance with all the queries. This is due to the fact that the data model is simpler in the representation of the process data, as it does not support the recording of the temporal evolution of process properties. The SOS implementation does not support such temporal evolution neither, in spite of being a general application independent solution. The SOS implementation represents properties as rows in a table of phenomena and each observation represents the data of a single property. Two observations (rows) must be retrieved and joined therefore to obtain the data of the two properties. The SOS implementation achieves a slightly better performance when only one property is needed, since it is not needed to project out not required properties from the stored table. On the other hand, its performance degenerates when two properties are required. It is reminded that the proposed framework represents observation properties with metadata in the catalog and with separate columns in the observation table of each process type. Thus, all the properties of each process may be obtained accessing a single observation.

In a second experiment, queries similar to the ones of the previous experiment were executed over two datasets of air quality observations. The first dataset contains more than 5.6 million raw observations generated by the low cost sensors used in the TRAFAIR project. Three observed properties were retrieved by each of the evaluated query types. The query performance results are shown in Fig. 7(a). The SOS implementation does not provide with results with reasonable response times retrieving 3 observed properties from such large dataset. The ad hoc model performs slightly better in all the queries due to the specific approach followed to represent the changes in the configurations of the sensors. However, despite of being a model specialized from a general purpose one, the current prototype provides results with response times with almost the same performance.

The second dataset contains around 1.5 million calibrated observations generated by the calibration models defined for the low cost sensors of the TRAFAIR project. The query performance results for queries returning a single observed property are shown in Fig. 7(b). If the spatial filter returns more than one spatial sampling feature of interest (sensor location), then the two generic solutions (current prototype of present framework and SOS implementation) achieve similar response times, faster than the ad hoc solution. This is so despite of the fact that the SOS implementation does not support different temporal versions of the same process. The lower performance of the



(a) Air Quality Raw Observations

(b) Air Quality Calibrated Observations (One Property)

(c) Air Quality Calibrated Observations (Two Properties)

Fig. 7. Query performance: Air quality observations.

ad hoc solution is due to the combination of a couple of reasons that complicate the required SQL statement: (i) the link between the calibration algorithm (process) and its location is not directly recorded and (ii) each version of the sensor status is stamped with a single time value and not with a time period, which complicates obtaining the status corresponding to a specific instant. When the spatial filter restricts to one specific spatial sampling feature identifier (spatial one point), the SOS model performs better than the current prototype of the present framework. This is due to the fact that the SOS model is good in querying short time series of simple observations (of just one property) at sampling locations. Fig. 7(c) shows the query performance results when two observed properties are retrieved. Now, the current prototype of the present framework outperforms the other two solutions in all cases. The performance of the SOS implementation degenerates when spatial filters are used, because the spatial restriction has to be tested for more rows, due to the fact that each property is represented in a different row.

The third experiment considered a dataset of around 600k observations generated by the wind observation instruments of a network of meteorological stations. The query performance results achieved by the queries that retrieve just one property are shown in Fig. 8(a). In this case, the SOS implementation obtains the best or near the best performances in all cases. The model used by the ad hoc solution is very similar to the one used by the SOS implementation, as it represents also properties as rows in a table of parameters and each value of each property is represented in a different row in the table of observations. The reason why the performance is worse for the ad hoc model is that is not easy to identify wind instruments in the database, as the type of instrument is encoded in a text field that has to be parsed. In any case, all the models offer performances in the same range of magnitude for one property. When two properties are requested, the model provided by the current prototype of the present framework outperforms the other two models. This is due to the need of two rows to represent each observation in the ad hoc and SOS implementations, which have to be joined. The proposed model has better performance with queries of two properties despite of supporting various temporal versions for the same process. Notice that the correct version of the process has to be found for each observation, and this has a cost.

A common characteristic of all the above experiments is that they evaluated the queries over datasets with simple observations (without subsamples). Two more experiments were performed to evaluate the models in a scenario of complex observations with potential subsample querying. Different query types were executed over a dataset of 49 radiosounding trajectories, with a total of near 20k subsamples, using the three models. The SOS model represents trajectories as parent complex observations whose children observations (trajectory subsamples) are recorded as separate observations also in the same observations table. The location of each subsample is recorded in the observations table under a *sampling_geometry* field. The line geometry of the whole trajectory is recorded in the feature of interest and it is linked to the parent (trajectory) observation. Regarding the ad hoc model, it is a very simple model designed to ease the recording of the data. It has only three tables. A first table is used to record radiosounding stations (locations from where radiosondes are released). Another table is used to represent radiosounding trajectories, recording only the time instant of release, a reference to the station and an identifier for the trajectory inside its corresponding station. The values of the observed properties are recorded in a third table, which contains a reference to the trajectory, an identifier of the measure in the trajectory, the time instant of the measure, latitude, longitude and height coordinates and one field for each observed property. The results for the queries that obtain just one of the observed properties, are shown in Fig. 9(a). All the three models offer very similar query performance figures, with a slightly better performance of the ad hoc model, due to its simplicity. Fig. 9(b) shows the results for queries retrieving two observed properties. Now, it may be observed how the performance of the SOS model is penalized by the fact of having to access and join two rows (one for each property) to obtain the data of each subsample. Again, the performance of the current prototype of the present framework and the simple ad hoc model are very similar.

Fig. 10(a) shows the results of spatial queries over the trajectories, to retrieve the whole trajectory (with all the properties). The response

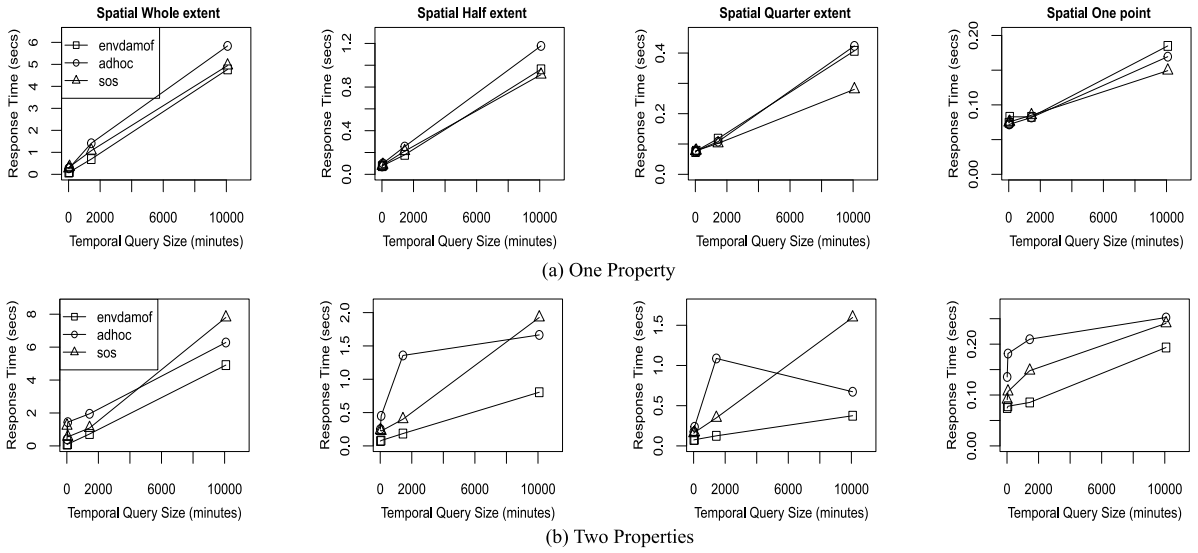


Fig. 8. Query performance: Meteorological stations observations.

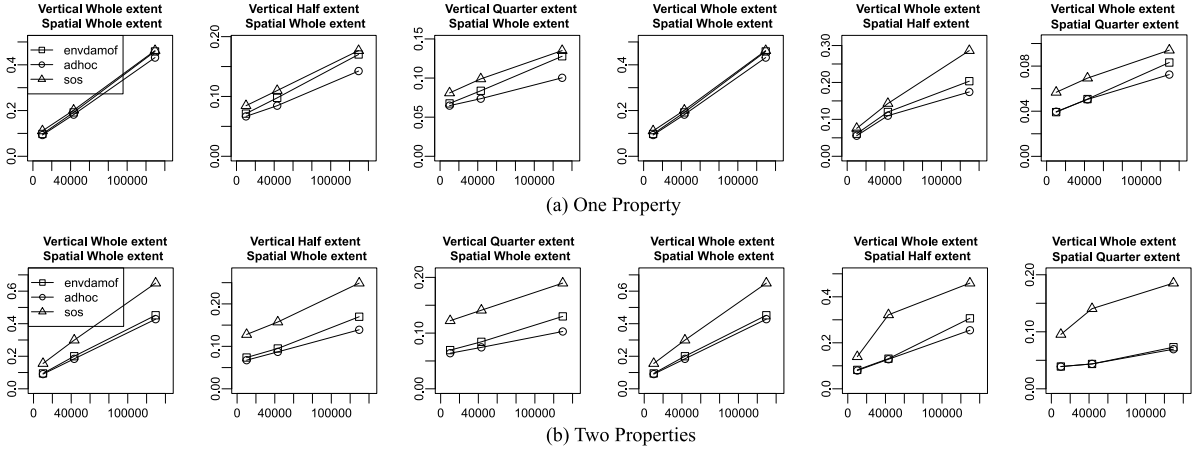


Fig. 9. Query performance: Radiosounding subsamples (X axis: Response time in seconds. Y axis: temporal query size in minutes).

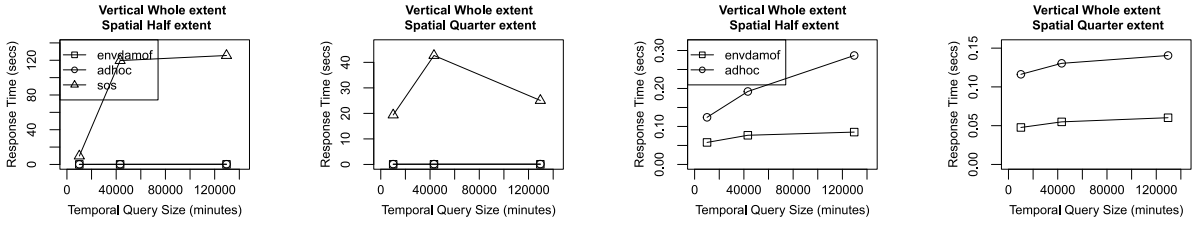


Fig. 10. Query performance: Radiosounding whole trajectory.

times of the SOS model are now orders of magnitude worse than those of the other two approaches, which disables its use for this type of queries. A zoom to the response times of only the two best performing models is shown in Figure Fig. 10(b). It may be appreciated now that the current prototype of the present framework has better performance than the ad hoc model when the whole trajectory has to be queried and retrieved. This due to the fact that the line geometry of the trajectory is already precomputed in the model of the present framework, whereas it has to be computed in query time in the simple ad hoc model. Additionally, the recording of the trajectory in a single aggregate JSON document offers also advantages when it has to be retrieved as a whole.

Fig. 11 shows the query response times for different query types over a dataset of near 19k ctd profiles with a total of more than 2 million subsamples (around 112 depths per profile in average). The ad hoc model has structures to record stations (sampling features), profiles (observations), devices (process), measures, parameters (observed properties) and data elements. A profile has one measure per parameter and each measure has a data element per depth level. Observed properties are therefore treated as data rows and not as columns, as it is the case of the SOS model. Profiles are treated as parent observations in the SOS model, having a collection of related child observation subsamples. A specific field of the observation table is used to determine if an observation is either a profile or a subsample. Vertical coordinates

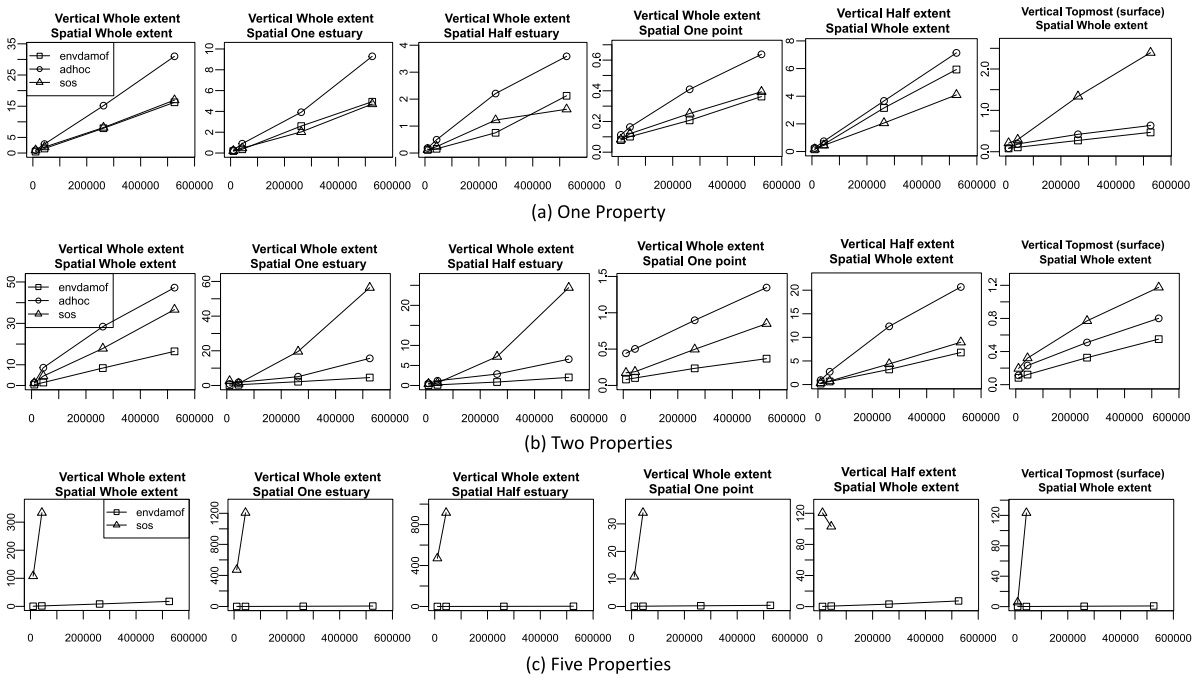


Fig. 11. Query performance: CTD profiles (X axis: Response time in seconds. Y axis: temporal query size in minutes).

are recorded in the SOS observation table. Response times of queries returning a single property are shown in Fig. 11(a). In general, the performance of the current prototype of the present framework is either the best or near the best. The ad hoc model has the worst performance in spatial queries. With vertical range queries, the SOS has a good performance, since vertical coordinates are directly used. In this case, the JSON aggregate used by the current prototype implementation has to be unnested and filtered, which degenerates its performance with respect to the SOS model. However, when the topmost vertical level is required (observation at sea surface), the SOS model suffers a performance problem, since it records vertical coordinates and not a vertical level. This is not the case of the ad hoc model, which records both vertical level in one field and depth as an observed property. The use of a JSON array structure in the current prototype of the present framework enables also the direct access to the first element, without having to compare coordinates to find the topmost. When two properties are required, the current implementation of the present framework outperforms the other two models, as it is shown in Fig. 11(b). Fig. 11(c) shows how the current implementation of the present framework maintains a very good performance when five properties are retrieved. The SOS model may be used only to retrieve few time instants, and even in that case with a too poor performance. The ad hoc model is even worse, and it is not shown in the figure due to too high response times.

As a synthesis of the above experiments and results, it has been shown that the SOS model analyzed suffers from important performance issues when it has to be used to retrieve various observed properties or when the size of the dataset is large (millions of observations). This is so despite not supporting various temporal versions of the same data generation process. The performance of the ad hoc models depends on the objectives considered during their design. Overall, if the required query types are considered during the design phase, then it will reach a performance that may be considered as a reference to be achieved by more generic models. In practice, ad hoc models are not always the best possible models. Finally, the models designed, and the data storage systems constructed with the current prototype of the present framework offered very good query performance figures in all the analyzed cases. As a final remark, it has to be noted that raster datasets were not considered during evaluation since their performance is mainly determined by the external raster data storage approach, which is out of the scope of this work.

7.3. Qualitative evaluation

The model of Level 1 of the present framework and some other relevant approaches are evaluated in this subsection with respect to the requirements specified in Section 3. A synthesis of the results of this evaluation is shown in Table 6. The justification and discussion corresponding to each requirement is provided below.

GR01 Designing models at Level 4 of the present framework enables the reuse of structures of the previous levels among different application areas and specific applications. Thus, the structures of Level 4 combine application specific structures designed for specific requirements with general purpose ones. This requirement is also fulfilled by specific models and profiles defined under the OGC O&M model, such as WaterML, H_2^gO, ODM2 and Mainstems. Models ODM, VODM and WaMDaM provide with general structures that may be used in many application domains and they have considered the linking with application specific structures, but only for the representation of spatial sampling features of interest. The NGSI-LD ontology is of general purpose and it may be reused in any application. That is not the case of the AIR_POLLUTION_Onto and QBOAirbase ontologies, which have been designed for air quality observations at static stations.

GR02 Various vocabularies may be used in each model of Level 4 in the present framework. Various names may be provided for feature types and properties in the catalog, specifying the vocabulary of each of them. Vocabularies in the data values of the instances may be specified using the Category data type. Standards related to the data structures may be incorporated by defining those structures in the Level 4 data model. As an example, it is possible to incorporate ISO 19115 or Dublin Core metadata elements in the definition of process types to incorporate those metadata at the level of dataset, however, the framework does not force this as mandatory. Other models

Table 6
Evaluation of related approaches with respect to the specified requirements supported by the present framework (Y = Yes, N = No, P = Partially).

Requirement	O&M (Cox, 2013)	WaterML (Taylor et al., 2013)	ODM (Horsburgh et al., 2008)	H^g_2O (Wojda and Brouyère, 2013)	VODM (Mason et al., 2014)	ODM2 (Horsburgh et al., 2016)	WaMDaM (Abdallah and Rosenberg, 2019)	Mainstems (Blodgett et al., 2021)	CDM-CF (Eaton et al., 2023)	NGSI-LD (ETSI, 2023)	AIR-_POLLUTION-_Onto Oprea (2009)	QBOAirbase (Galárraga et al., 2017)
GR01	Y	Y	P	Y	P	Y	P	Y	Y	Y	N	N
GR02	P	P	P	P	P	P	Y	P	Y	Y	Y	Y
GR03	Y	Y	N	Y	N	Y	N	Y	N	P	N	N
GR04	N	N	N	N	N	N	N	N	Y	–	–	–
L1R01-L1R04	Y	Y	P	Y	P	Y	P	Y	Y	Y	N	N
L2R01	Y	Y	P	Y	P	Y	P	Y	N	Y	N	N
L2R02	Y	Y	N	Y	N	N	N	Y	Y	N	N	N
L2R03	Y	Y	Y	Y	Y	Y	Y	Y	P	N	N	N
L2R04	P	P	P	P	P	P	P	P	N	N	N	N
L2R05	Y	Y	P	Y	Y	Y	P	Y	N	N	N	N
L2R06	Y	Y	Y	Y	Y	Y	N	Y	N	N	N	N
L2R07	Y	Y	P	Y	P	Y	P	Y	N	P	P	P
L2R08	P	P	P	P	P	P	P	P	P	N	N	N

enable the use of various vocabularies to name data elements. Thus, NetCDF attributes may be used in CDM-CF to incorporate standard CF names. Semantic web based solutions such as NGSI-LD, AIR_POLLUTION_Onto and QBOAirbase may also incorporate different terminologies in their ontologies. On the other hand, all the other evaluated approaches enable the use of only one vocabulary for the specification of the feature type and property names. Regarding the possibility of incorporating other standard data structures, this is possible in those approaches based on the OGC O&M model (O&M, WaterML, H^g_2O, ODM2, Mainstems) and also in CDM-CF and NGSI-LD, but it is not possible in all the other.

GR03 Data integration may be done in the present framework at Levels 1, 2 and 3, by using common data structures defined at those levels. Thus, at Level 1, information systems may be implemented to provide with functionality of complete general purpose over features types and their relationships (as the one provided by generic DBMS clients). Generic environmental data tools may be implemented assuming the model of Level 2, providing now specific functionalities related to data generation processes, features of interest, observations and their geospatial and temporal context. Systems that may integrate different datasets coming from specific application areas may be implemented assuming the models at Level 3 and specific applications with their specific functionalities use the models of level 4. An example of data integration has been shown in Section 7.1. The above options may also be enabled by all the models based on OGC O&M (O&M, WaterML, H^g_2O, ODM2, Mainstems), by defining specific profiles of the proposed models. NGSI-LD considers only two levels of abstraction, and all the other define only one data model.

GR04 Performance evaluation results were discussed in Section 7.2. It was shown that the current prototype of the present framework has in general a good performance, only clearly beaten by some ad hoc simple models in specific cases. On the other hand, the direct implementation of the OGC O&M model in a SOS tool has shown poor query response times in many cases, specifically when either the dataset is very large or various properties have to be retrieved or the observation type is complex. The key characteristic of such a direct implementation of the OGC O&M data model that causes bad performance is the use of various rows to represent various observation components (properties and/or subsamples). This characteristic is shared by all the models based on O&M (O&M, WaterML, H^g_2O, ODM2, Mainstems) and by ODM, VODM and WaMDaM. It is obvious that specific models implemented with CDM-CF may yield good performance. The performance of the sensor web based solutions (NGSI-LD, AIR_POLLUTION_Onto and QBOAirbase) was not tested, due to the lack of implementations that support the considered datasets.

L1R01-L1R04 The general metamodel of Level 1 of the present framework provides with support for representation of entities with their relevant properties and relationships. The model supports also user defined data types. This is also the case of all the models based on the OGC O&M (O&M, WaterML, H^g_2O, ODM2, Mainstems), since they are defined as specializations of the general OGC model for geosptial features (Kottman and Reed, 2009). The CDM-CF approach provides also the general purpose structures (dimensions, attributes and variables) that enable the support of the above features. NGSI-LD is based on the RDF model (Cyganiak et al., 2014), which provides the required general purpose data representation capabilities. Other models are more specific. In particular, ODM, VODM and WaMDaM enable the incorporation of any structure, but only as a spatial sampling feature of interest type. Finally, AIR_POLLUTION_Onto and QBOAirbase are specific models for air quality observed in stations.

L2R01 OGC standards are used in the present framework to model geometric objects and time periods. Following a similar approach, vertical intervals have also been added. The same, except for the lack of specific support of vertical intervals, applies also to all the solutions based on OGC O&M (O&M, WaterML, H^g_2O, ODM2, Mainstems). VODM supports geometries but does not define periods and intervals. Geometries may also be used in NGSI-LD by incorporating geometric RDF representations (Car et al., 2024). Only points represented by geographic coordinates are supported in ODM, WaMDaM, AIR_POLLUTION_Onto and QBOAirbase.

L2R02 The representation of geospatial coverages of different types is supported by relevant data types of the model of Level 2 in the present framework. In the current prototype, those representations reference out-of-band external data structures that efficiently store the coverages. Different encodings are considered in OGC standards to represent coverages, and they may be incorporated in all the solutions that extend OGC O&M (O&M, WaterML, H^g_2O, ODM2, Mainstems). All the other approaches do not consider the representation of coverages as an objective.

L2R03 Data values that record measures with units of measure and categories of specific vocabularies are supported in the present framework as values of data types Measure and Category. These data types are inherited from OGC standards, thus, all the approaches based on OGC O&M follow the same approach (O&M, WaterML, $H_2^g O$, ODM2, Mainstems). Measures and categories are also supported by ODM, VODM and WaMDaM. CDM-CF use NetCDF attributes with CF standard names to represent the units of measure of variables, but it does not have a standard mechanism to represent categories (only for data quality flags). No relevant specific support is provided in NGSI-LD, AIR_POLLUTION_Onto and QBOAirbase.

L2R04 Class *Process* of the Level 2 data model of the present framework, and relevant subclasses in models of lower levels, are used to represent context data of data generation processes and their evolution with respect to time. No specific and standard support for evolution with respect to time is provided in any other model, although many of them provide some kind of structure to represent processes (O&M, WaterML, ODM, $H_2^g O$, VODM, ODM2, WaMDaM, Mainstems).

L2R05 Class *SpatialSamplingFeature* of Level 2 provides support for spatial sampling in the current framework. General spatial sampling context data is also incorporated by all the models based on OGC O&M (O&M, WaterML, $H_2^g O$, ODM2, Mainstems). ODM, AIR_POLLUTION_Onto and QBOAirbase support only sampling points, which are also supported in WaMDaM in the context of a hydrographic network.

L2R06 Specimens are represented in the present framework with instances of class *Specimen*. The methods used to collect those specimens are described with instances of class *SamplingProcess*. The recording of context data of specimens and related processes are supported in all the models specialized from the OGC O&M model (O&M, WaterML, $H_2^g O$, ODM2, Mainstems). Data of specimens may also be recorded in ODM and VODM. This type of sampling is not considered explicitly in any of the other analyzed models.

L2R07 Both result and phenomenon time stamping are supported in the results of the present framework. This is also the case of other models based on OGC O&M (O&M, WaterML, $H_2^g O$, ODM2, Mainstems). Only one time instant is considered in many approaches, which is usually interpreted as phenomenon time (ODM, VODM, WaMDaM, NGSI-LD, AIR_POLLUTION_Onto and QBOAirbase). CDM-CF is of more general purpose, thus, it does not assume any specific semantics for time stamps.

L2R08 Class *ObservationSubsample* of Level 2 and relevant subclasses in lower Levels of the present framework are used to support values with complex structure. Such complex values may be incorporated also in other models based on OGC O&M, although they are not explicitly part of the model. Some kinds of subsamples have been considered in ODM2 (time series, sections and transects) and WaMDaM (time series and electronic files). CDM-CF defines structures for grid coverages, points, time series, trajectories, profiles, time series of profiles and trajectories of profiles. Complex values are represented with groups of simple ones in ODM and with groups and datasets in VODM. No specific support has been defined in NGSI-LD, AIR_POLLUTION_Onto and QBOAirbase.

8. Conclusions and future work

A conceptual data modeling framework for environmental information was proposed. The framework provides models at four levels of abstraction. Levels 1 and 2 define a general environmental data model based on the OGC O&M standard. At this level, the framework provides also a metadata catalog that supports the use of multiple vocabularies. Defining generic data models of Level 3 enables the adaption of the framework to the specific needs of different application areas. The general purpose structures of the above levels of abstraction may be reused in many applications, combining general purpose structures with application specific ones. The reuse of such common structures and the potential use of standard vocabularies eases the integration of different and heterogeneous datasets at various levels of abstraction. The framework supports the representation of context data related to data generation processes, features of interest, sampling features and generated values, which include simple values and complex ones, such as time series, trajectories, transects, profiles and different types of spatial and spatio-temporal coverages.

General purpose designs may be used directly by end-users without any design effort to be done, however they use to fail in providing the efficiency that may be achieved by good ad hoc designs. It was shown that the general purpose direct O&M implementations provided by SOS tools have important performance problems in many cases. The current framework provides a general design at level 2, but it requires some design effort to produce levels 3 and 4. On the other hand, the performance achieved is in general close to that of ad hoc models (or even better when the design of those ad hoc models is not good). Application domain experts with some data modeling skills should agree in the design of level 3 models. This is not a simple task in general, but when it is achieved, the benefits are very important. Next, end-users may benefit from the existence of those level 3 models to ease the design of their level 4 models. They need to have some data modeling knowledge, however much data modeling experience would not be required to achieve good solutions since they are already supported by existing models of level 3.

Direct O&M implementations provided by SOS tools and some other approaches do not require a previous declaration of the properties of features and of the observed properties measured by processes. This is close to the idea of not requiring schema definition of many NoSQL databases (Sadalage and Fowler, 2013). On the other hand, the user must declare the properties of feature types and the observed properties of processes in the proposed framework during the design of level 4 models. Those declarations are recorded together with other parts of the schema in the catalog defined at level 2. Advantages and disadvantages of working with and without schema have largely been studied by the database community. In general, the lack of schema eases the incorporation of schema changes during data insertion. However, the lack of schema brings important problems for applications that perform data querying, since the changes in the schema are not documented in the system.

Directions of future work are mainly oriented to the developing of general technologies for the searching and interactive exploration of environmental datasets modeled with the present framework. Besides, research work is still needed to adapt the framework to support various versions of the models of levels 3 and 4, enabling the evolution of the database schema. Finally, a step forward in the data modeling line will be the proposal of a metamodel that enables the specification of application specific solutions without the need of intermediate levels of abstraction.

CRediT authorship contribution statement

David Martínez: Software, Investigation. **Laura Po:** Writing – review & editing, Investigation, Conceptualization. **Raquel Trillo-Lado:** Writing – review & editing, Investigation, Conceptualization. **José R.R. Viqueira:** Writing – review & editing, Software, Investigation, Conceptualization.

Table 7
Level 3 requirements: Traffic in Smart Cities.

Requirement	Description
L3TF01	The data generated by the automatic monitoring of traffic at specific locations of the road network.
L3TF02	The real time traffic conditions estimated by traffic reconstruction models at each section of the road network.
L3TF03	The future traffic conditions predicted by traffic models for each section of the road network.

Table 8
Level 3 requirements: Air Quality in Smart Cities.

Requirement	Description
L3AQ01	The data generated by static air quality monitoring stations.
L3AQ02	The data generated by in-situ removable devices, installed at ground static platforms, ground mobile platforms and flying platforms.
L3AQ03	The data generated by nowcast and forecast models for air quality.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was partially supported by the following projects. TRAFAIR project (2017-EU-IA-0167), co-financed by the Connecting Europe Facility of the European Union. Galicia Marine Science programme, which is part of the Complementary Science Plans for Marine Science of Ministerio de Ciencia, Innovación *y* Universidades included in the Recovery, Transformation and Resilience Plan (PRTR-C17.I1), funded through Xunta de Galicia with NextGenerationEU and the European Maritime Fisheries and Aquaculture Funds. EarthDL-USC (PID2022-141027NB-C22) and NEAT-AMBIENCE (PID2020-113037RB-I00) projects, funded by Agencia Estatal de Investigación, Ministerio de Ciencia e Innovación, through the national plan of scientific and technical research and innovation 2021–2023.

Appendix. Data models for traffic and air quality in smart cities

A.1. Level 3 requirements and data models

In the scope of the development of smart city infrastructures, it is usual to include services related to the monitoring of traffic conditions and air quality. In this section, for illustration purposes, requirements and abstract models are proposed for such application area. In particular, Tables 7 and 8 show the requirements for traffic and air quality models, respectively.

The abstract data model of level 3 for traffic monitoring in the context of smart cities is shown in Fig. 12. Broadly, the model defines structures to support the recording of observations generated by traffic sensors located at specific locations of the road network (**L3TF01**). It incorporates also structures to record outputs of two types of models: (i) models that estimate the traffic in each section of the road network in real time ((**L3TF02**)) and (ii) models that predict the evolution with respect to time of the traffic in the road network ((**L3TF02**)).

Fig. 13 shows the abstract data model of level 3 that supports the recording of data obtained from the monitoring of air quality in smart cities. The model provides structures to support all the requirements defined in Table 8.

A.2. Traffic and air quality data in the TRAFAIR project

The main objective of the TRAFAIR project (Po et al., 2019) was the monitoring and prediction of air quality at high scale inside cities. Traffic in the cities was monitored at specific locations using traffic sensors (Bachechi et al., 2022b). Using those traffic observations, a traffic reconstruction model was executed to provide real time estimations of traffic flow in each road section (Bachechi et al., 2022a; Bilotta and Nesi, 2021). Fig. 14(a) shows a Level 4 data model that supports the recording of the traffic data generated in the TRAFAIR project. Classes *road_section*, *road_segment* and *road_node* are used to record the road network obtained from OpenStreetMap.[21] Traffic sensors (Class *traffic_sensor*) are located at specific segments and are used to generate traffic observations. Notice that class *traffic_sensor* has the role of both a *TrafficStation* and a *TrafficPointObservationProcess*. Each traffic observation (Class *sensor_traffic_observation*) provides a value for the traffic flow, i.e., number of vehicles per hour, and for the traffic occupancy (percentage of time when the sensor was detecting a vehicle), every 5 min. The road network was filtered and transformed in the project to generate a graph of main street road arcs (Class *road_arc*). The traffic reconstruction model (class traffic_flow_model) provides an estimation of the traffic flow every 15 min for each road arc, which is recorded in class *traffic_flow_model_output*.

Air quality monitoring was performed with low cost sensors (Rollo et al., 2023; Casari and Po, 2024) and calibration models built with machine learning (Bachechi et al., 2024). Air quality prediction was performed with the GRAL pollutant dispersion model,[22] using traffic emission estimations as main pollutant sources. Fig. 14(b) shows a Level 4 data model that enables the recording of the air quality data generated in the TRAFAIR project. Air quality legal stations (class *aq_legal_station*) provide with observations with legal coverage inside the city. These features model both air quality stations (*AQStation*) and air quality in-situ static processes (*AQInSituStaticProcess*), since their sensors are not moving through different locations during their lifetime. Each observation (class *aq_legal_station_observation*) contains the concentration of various gases recorded in different observed properties. Low cost sensors (class *sensor_low_cost*) are used at different locations during the project. In fact, they must be collocated with a legal station during some periods to generate training datasets that enable the generation of the calibration models. Each raw observation generated by a low cost sensor (class *sensor_raw_observation*) refers to a specific sampling location (class *sensor_low_cost_feature*) and it contains a battery voltage measure, a humidity measure, a temperature measure, and a couple of raw voltage measures, generated by an electrochemical cell, for each of the following gases: CO, NO, NO_2, O_3. Low cost sensor raw voltage observations are transformed to generate gas concentrations using a calibration model (class *sensor_calibration*). A calibration model is generated for a specific sensor and it has an algorithm for each gas. The algorithm consists of a machine learning model trained with a dataset generated during the collocation of the sensor with a legal station. All the metadata of each algorithm is recorded in a JSON structure, as it is shown in the diagram, and it includes the reference to the legal station, the training period, the input variables used by the model (including at least sensor voltages), a reference to the model implementation used (*sklearn.ensemble.RandomForestRegressor* was used in the project) and the values of the used hyper-parameters. Calibrated observations (*sensor_calibrated_observation*) provide with estimations for gas concentrations at the spatial sampling features. Finally, the data model enables also the recording of the outputs generated by the GRAL pollutant dispersion model, which have the form of spatiotemporal coverages of NO_x concentration values. Class *air_quality_model* records the metadata of the GRAL configuration used to generate the predictions and also the characteristics of the spatio-temporal grid considered. Model outputs, i.e., spatio-temporal coverages of NOx values are recorded in class *air_quality_model_output*.

[21] https://www.openstreetmap.org/
[22] https://gral.tugraz.at/

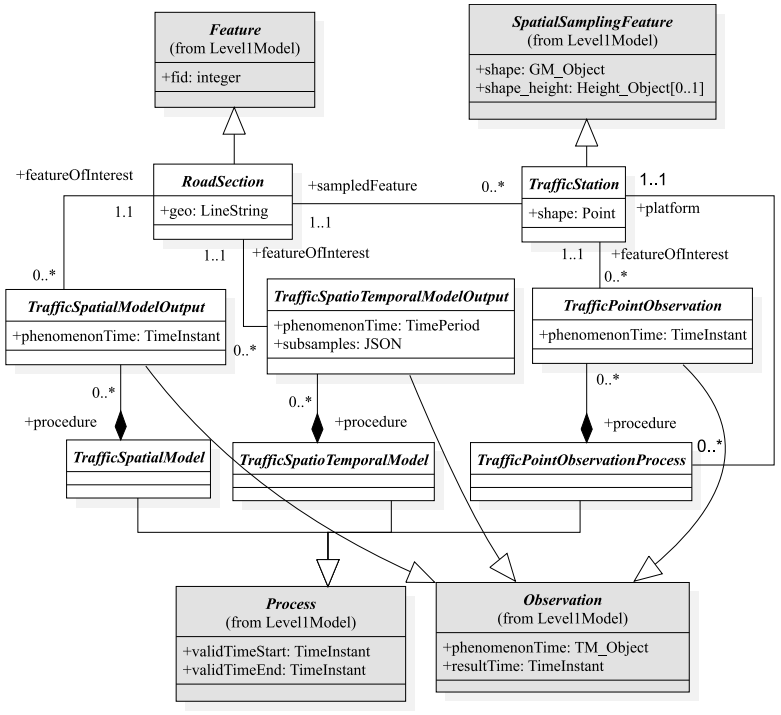


Fig. 12. Level 3 data model for traffic.

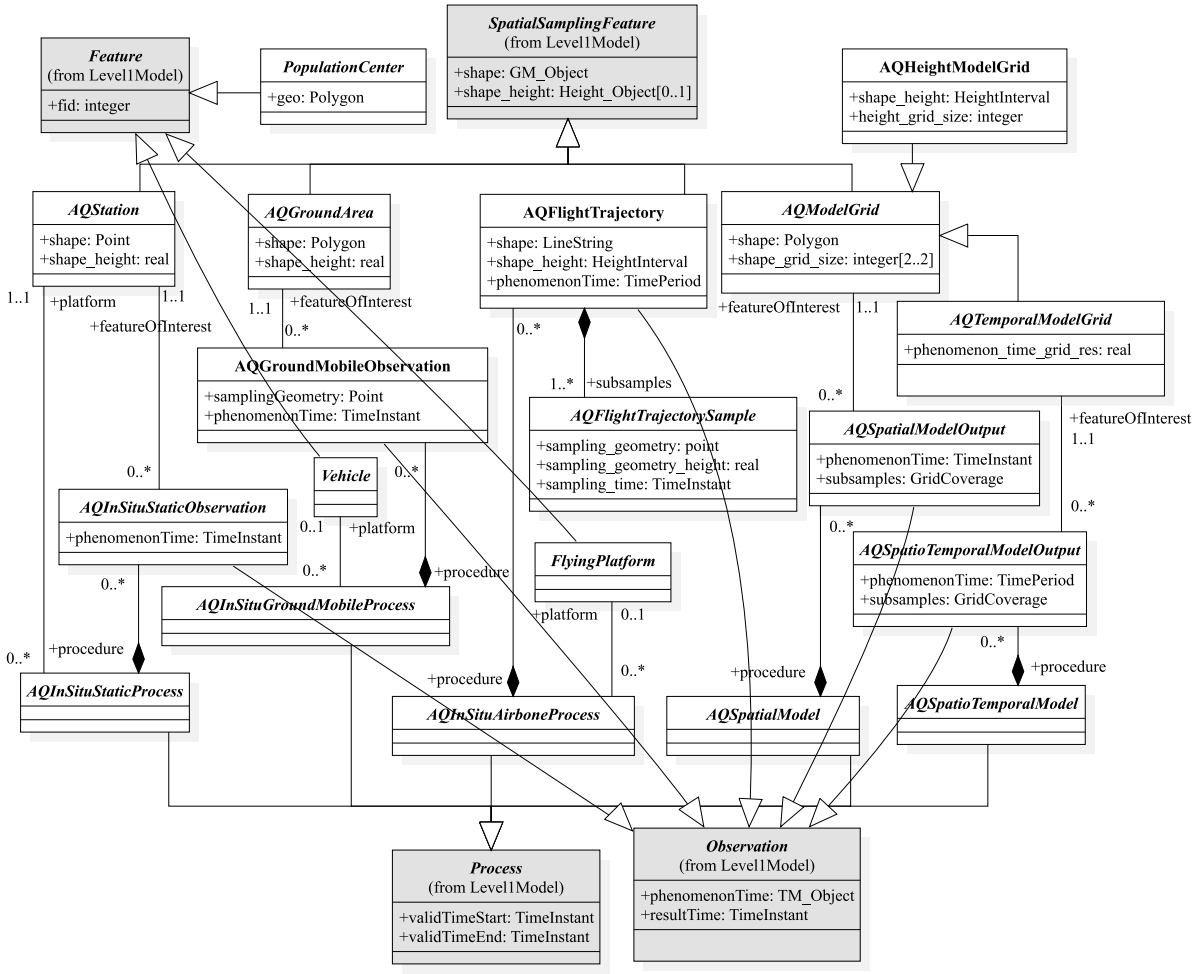
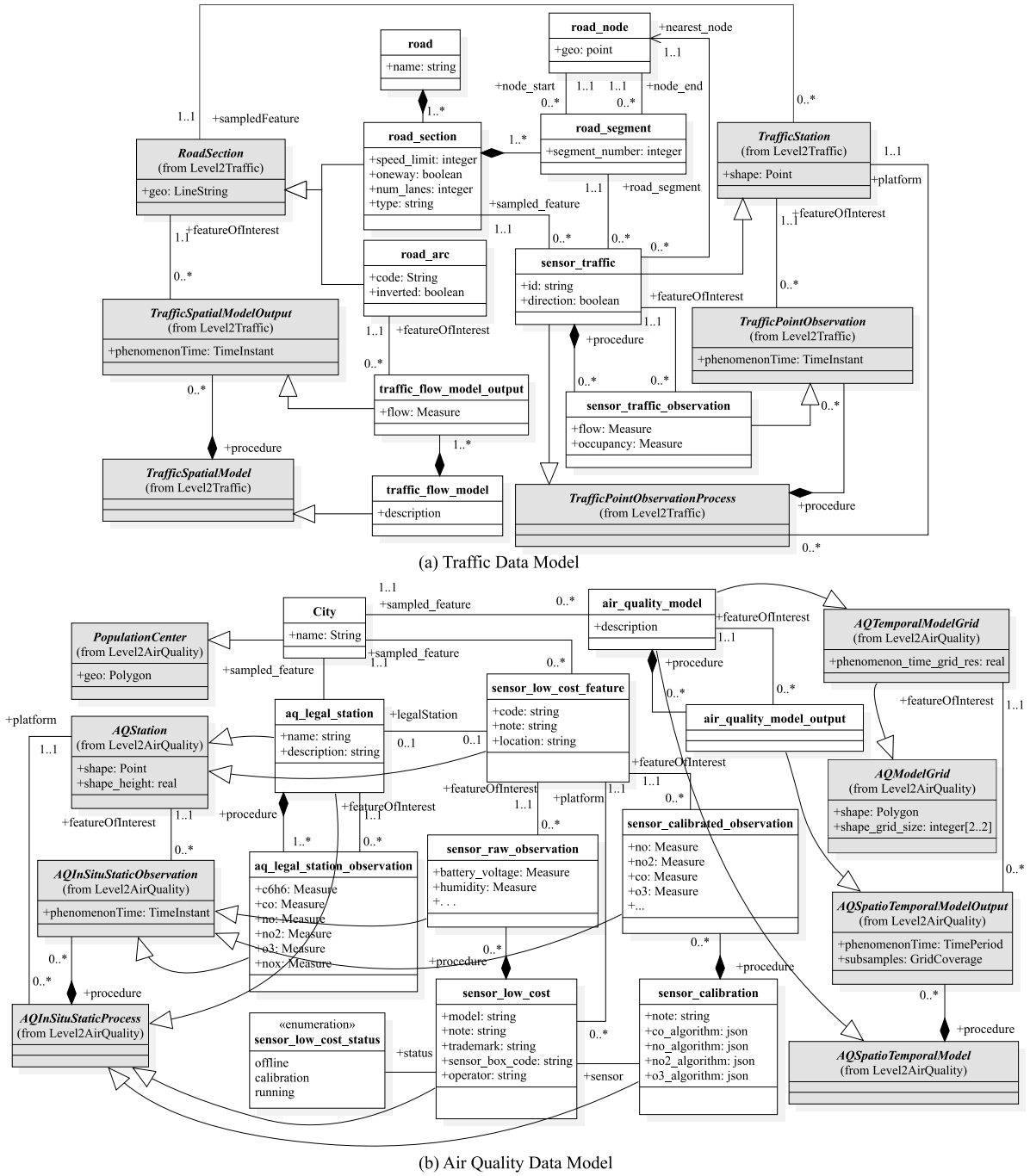


Fig. 13. Level 3 data model for air quality.



(a) Traffic Data Model

(b) Air Quality Data Model

Fig. 14. Level 4 data models for traffic and air quality data in the TRAFAIR project.

Data availability

The data of all the datasets is publicly available in their relevant data producer organizations. Traffic and air quality data of the city of Santiago de Compostela may be downloaded from the European Data Portal https://data.europa.eu/. The meteorological data is available at open data infrastructure of MeteoGalicia https://www.meteogalicia.gal. The oceanograhic data is available at the open data infrastructure of Intecmar https://www.intecmar.gal. To reproduce the experiments undertaken during the evaluation of the framework, CSV files adapted to the schema of each of the approaches may be provided upon request to the corresponding author of this paper.

References

Abdallah, A.M., Rosenberg, D.E., 2019. A data model to manage data for water resources systems modeling. Environ. Model. Softw. 115, 113–127. http://dx.doi.org/10.1016/j.envsoft.2019.02.005.

Bachechi, C., Po, L., Rollo, F., 2022a. Big data analytics and visualization in traffic monitoring. Big Data Res. 27, 100292. http://dx.doi.org/10.1016/j.bdr.2021.100292, URL: https://www.sciencedirect.com/science/article/pii/S221457962100109X.

Bachechi, C., Rollo, F., Po, L., 2022b. Detection and classification of sensor anomalies for simulating urban traffic scenarios. Cluster Comput. 25, 2793–2817. http://dx.doi.org/10.1007/s10586-021-03445-7.

Bachechi, C., Rollo, F., Po, L., 2024. HypeAIR: A novel framework for real-time low-cost sensor calibration for air quality monitoring in smart cities. Ecol. Inform. 81, 102568. http://dx.doi.org/10.1016/j.ecoinf.2024.102568, URL: https://www.sciencedirect.com/science/article/pii/S1574954124001109.

Baumann, P., 1994. Management of multidimensional discrete data. VLDB J. 3 (4), 401–444. http://dx.doi.org/10.1007/BF01231603.

Bilotta, S., Nesi, P., 2021. Traffic flow reconstruction by solving indeterminacy on traffic distribution at junctions. Future Gener. Comput. Syst. 114, 649–660. http://dx.doi.org/10.1016/j.future.2020.08.017.

Blaha, M., Rumbaugh, J., 2005. Object-oriented Modeling and Design with UML. Pearson Education.

Blodgett, D., Johnson, J.M., Sondheim, M., Wieczorek, M., Frazier, N., 2021. Mainstems: A logical data model implementing mainstem and drainage basin feature types based on WaterML2 part 3: HY features concepts. Environ. Model. Softw. 135, 104927. http://dx.doi.org/10.1016/j.envsoft.2020.104927.

Brambilla, M., Cabot, J., Wimmer, M., 2017. Model-Driven Software Engineering in Practice: Second Edition, second ed. Morgan & Claypool Publishers.

Bröring, A., Stasch, C., Echterhoff, J., 2012. OGC Sensor Observation Service Interface Standard. Open Geospatial Consortium Inc., OpenGIS Implementation Standard. https://www.ogc.org/standard/sos/.

Brown, P.G., 2010. Overview of sciDB: Large scale array storage, processing and analysis. In: Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data. SIGMOD '10, New York, NY, USA, ACM, pp. 963–968. http://dx.doi.org/10.1145/1807167.1807271.

Car, N.J., Homburg, T., Perry, M., Knibbe, F., Cox, S.J., Abhayaratna, J., Bonduel, M., Cripps, P.J., Janowicz, K., 2024. OGC GeoSPARQL - A Geographic Query Language for RDF Data. Open Geospatial Consortium Inc., OpenGIS Implementation Standard. http://www.opengis.net/doc/IS/geosparql/1.1.

Casari, M., Po, L., 2024. MitH: A framework for mitigating hygroscopicity in low-cost PM sensors. Environ. Model. Softw. 173, 105955. http://dx.doi.org/10.1016/j.envsoft.2024.105955, URL: https://www.sciencedirect.com/science/article/pii/S1364815224000161.

Chen, P.P.-S., 1976. The entity-relationship model—toward a unified view of data. ACM Trans. Database Syst. 1 (1), 9–36. http://dx.doi.org/10.1145/320434.320440.

Codd, E.F., 1970. A relational model of data for large shared data banks. Commun. ACM 13 (6), 377–387. http://dx.doi.org/10.1145/362384.362685.

Compton, M., Barnaghi, P., Bermudez, L., García-Castro, R., Corcho, O., Cox, S., Graybeal, J., Hauswirth, M., Henson, C., Herzog, A., Huang, V., Janowicz, K., Kelsey, W.D., Phuoc, D.L., Lefort, L., Leggieri, M., Neuhaus, H., Nikolov, A., Page, K., Passant, A., Sheth, A., Taylor, K., 2012. The SSN ontology of the W3C semantic sensor network incubator group. J. Web Semant. 17, 25–32. http://dx.doi.org/10.1016/j.websem.2012.05.003.

Cox, S., 2013. Observations and Measurements. Verion 2.0. Open Geospatial Consortium Inc., The OpenGIS Abstract Specification. http://www.opengis.net/doc/as/om/2.0.

Cyganiak, R., Wood, D., Lanthaler, M., 2014. RDF 1.1 Concepts and Abstract Syntax. World Wide Web Consortium, W3C Recommendation. https://www.w3.org/TR/rdf11-concepts/.

de Bakker, M.P., de Jong, K., Schmitz, O., Karssenberg, D., 2017. Design and demonstration of a data model to integrate agent-based and field-based modelling. Environ. Model. Softw. 89, 172–189. http://dx.doi.org/10.1016/j.envsoft.2016.11.016, URL: https://www.sciencedirect.com/science/article/pii/S1364815216305898.

Devys, E., Habermann, T., Heazel, C., Lott, R., Rouault, E., 2019. OGC GeoTIFF Standard. Open Geospatial Consortium Inc., OGC Implementation Standard. http://www.opengis.net/doc/IS/GeoTIFF/1.1.

Eaton, B., et al., 2023. NetCDF climate and forecast (CF) metadata conventions. https://cfconventions.org. (Online; Accessed 14 April 2024).

ETSI, 2023. Context Information Management (CIM);NGSI-LD Information Model. European Telecommunications Standardization Institute (ETSI), Group Specification (GS). https://www.etsi.org/deliver/etsi_gs/CIM/001_099/006/01.02.01_60/gs_CIM006v010201p.pdf.

Fernandez, S., Hadfi, R., Ito, T., Marsa-Maestre, I., Velasco, J.R., 2016. Ontology-based architecture for intelligent transportation systems using a traffic sensor network. Sensors 16 (8), http://dx.doi.org/10.3390/s16081287.

Galárraga, L., Mathiassen, K.A.M., Hose, K., 2017. QBOAirbase: The European air quality database as an RDF cube. In: Nikitina, N., Song, D., Fokoue, A., Haase, P. (Eds.), Proceedings of the ISWC 2017 Posters & Demonstrations and Industry Tracks Co-Located with 16th International Semantic Web Conference. ISWC 2017, Vienna, Austria, October 23rd - To - 25th, 2017, In: CEUR Workshop Proceedings, vol. 1963, CEUR-WS.org, URL: https://ceur-ws.org/Vol-1963/paper507.pdf.

Gonzalez-Perez, C., Henderson-Sellers, B., 2008. Metamodelling for Software Engineering. Wiley Publishing.

Haller, A., Janowicz, K., Cox, S., Phuoc, D.L., Taylor, K., Lefrançois, M., 2017. Semantic Sensor Network Ontology. Open Geospatial Consortium and World Wide Web Consortium, W3C Recommendation. https://www.w3.org/TR/vocab-ssn/.

Harpham, Q., 2020. A simple taxonomy for describing the spatio-temporal structure of environmental modelling data. Environ. Model. Softw. 133, 104810. http://dx.doi.org/10.1016/j.envsoft.2020.104810.

Herring, J., 2020. Topic 1: Spatial Schema.. Open Geospatial Consortium Inc., The OpenGIS Abstract Specification. https://www.ogc.org/standards/as/.

Horsburgh, J.S., Aufdenkampe, A.K., Mayorga, E., Lehnert, K.A., Hsu, L., Song, L., Jones, A.S., Damiano, S.G., Tarboton, D.G., Valentine, D., Zaslavsky, I., Whitenack, T., 2016. Observations data model 2: A community information model for spatially discrete Earth observations. Environ. Model. Softw. 79, 55–74. http://dx.doi.org/10.1016/j.envsoft.2016.01.010.

Horsburgh, J.S., Tarboton, D.G., Maidment, D.R., Zaslavsky, I., 2008. A relational model for environmental and water resources data. Water Resour. Res. 44 (5), http://dx.doi.org/10.1029/2007WR006392, URL: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2007WR006392.

Johansson, L., Karppinen, A., Kurppa, M., Kousa, A., Niemi, J.V., Kukkonen, J., 2022. An operational urban air quality model ENFUSER, based on dispersion modelling and data assimilation. Environ. Model. Softw. 156, 105460. http://dx.doi.org/10.1016/j.envsoft.2022.105460, URL: https://www.sciencedirect.com/science/article/pii/S1364815222001657.

Kim, D., Muste, M., Merwade, V., 2015. A GIS-based relational data model for multi-dimensional representation of river hydrodynamics and morphodynamics. Environ. Model. Softw. 65, 79–93. http://dx.doi.org/10.1016/j.envsoft.2014.12.002.

Kimball, R., Ross, M., 2002. The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling, second ed. John Wiley & Sons, Inc., USA.

Kottman, C., Reed, C., 2009. Topic 5: Features. Open Geospatial Consortium Inc., The OpenGIS Abstract Specification. https://www.ogc.org/standards/as/.

Lebo, T., Sahoo, S., McGuinness, D., 2013. PROV-O: The PROV Ontology. World Wide Web Consortium, W3C Recommendation. http://www.w3.org/TR/prov-o/.

Levy, A.Y., 2000. Logic-based techniques in data integration. In: Minker, J. (Ed.), Logic-Based Artificial Intelligence. Springer US, Boston, MA, pp. 575–595. http://dx.doi.org/10.1007/978-1-4615-1567-8_24.

Liang, S., Khalafbeigi, T., van der Schaaf, H., 2021. OGC SensorThings API Part 1: Sensing Version 1.1. Open Geospatial Consortium Inc., OGC Implementation Standard. http://www.opengis.net/doc/is/sensorthings/1.1.

Martínez, D., Po, L., Trillo-Lado, R., Viqueira, J.R.R., 2022. TAQE: A data modeling framework for traffic and air quality applications in smart cities. In: Braun, T., Cristea, D., Jäschke, R. (Eds.), Graph-Based Representation and Reasoning. Springer International Publishing, Cham, pp. 25–40.

Mason, S.J., Cleveland, S.B., Llovet, P., Izurieta, C., Poole, G.C., 2014. A centralized tool for managing, archiving, and serving point-in-time data in ecological research laboratories. Environ. Model. Softw. 51, 59–69. http://dx.doi.org/10.1016/j.envsoft.2013.09.008.

Nativi, S., Mazzetti, P., Santoro, M., Papeschi, F., Craglia, M., Ochiai, O., 2015. Big data challenges in building the global earth observation system of systems. Environ. Model. Softw. 68, 1–26. http://dx.doi.org/10.1016/j.envsoft.2015.01.017.

OGC, 2007. Topic 6: Schema for Coverage Geometry and Functions. Version 7.0.0.. Open Geospatial Consortium Inc., The OpenGIS Abstract Specification. https://www.ogc.org/standards/as/.

Oprea, M.M., 2009. AIR_POLLUTION_Onto: an ontology for air pollution analysis and control. In: Iliadis, Maglogiann, Tsoumakasis, Vlahavas, Bramer (Eds.), Artificial Intelligence Applications and Innovations III. Springer US, Boston, MA, pp. 135–143.

Pisoni, E., De Marchi, D., di Taranto, A., Bessagnet, B., Sajani, S.Z., De Meij, A., Thunis, P., 2024. SHERPA-cloud: An open-source online model to simulate air quality management policies in Europe. Environ. Model. Softw. 176, 106031. http://dx.doi.org/10.1016/j.envsoft.2024.106031, URL: https://www.sciencedirect.com/science/article/pii/S1364815224000926.

Po, L., Rollo, F., Viqueira, J.R.R., Lado, R.T., Bigi, A., López, J.C., Paolucci, M., Nesi, P., 2019. TRAFAIR: Understanding traffic flow to improve air quality. In: 2019 IEEE International Smart Cities Conference. ISC2, pp. 36–43. http://dx.doi.org/10.1109/ISC246665.2019.9071661.

Regueiro, M.A., Viqueira, J.R., Stasch, C., Taboada, J.A., 2017. Semantic mediation of observation datasets through sensor observation services. Future Gener. Comput. Syst. 67, 47–56. http://dx.doi.org/10.1016/j.future.2016.08.013.

Rigaux, P., Scholl, M., Voisard, A., 2001. Spatial Databases With Application to GIS. Morgan Kaufmann.

Rollo, F., Bachechi, C., Po, L., 2023. Anomaly detection and repairing for improving air quality monitoring. Sensors 23 (2), http://dx.doi.org/10.3390/s23020640, URL: https://www.mdpi.com/1424-8220/23/2/640.

Sadalage, P.J., Fowler, M., 2013. NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence. Addison-Wesley, Upper Saddle River, NJ.

Salas, D., Liang, X., Navarro, M., Liang, Y., Luna, D., 2020. An open-data open-model framework for hydrological models' integration, evaluation and application. Environ. Model. Softw. 126, 104622. http://dx.doi.org/10.1016/j.envsoft.2020.104622.

Tarboton, D.G., Ames, D.P., Horsburgh, J.S., Goodall, J.L., Couch, A., Hooper, R., Bales, J., Wang, S., Castronova, A., Seul, M., Idaszak, R., Li, Z., Dash, P., Black, S., Ramirez, M., Yi, H., Calloway, C., Cogswell, C., 2024. HydroShare retrospective: Science and technology advances of a comprehensive data and model publication environment for the water science domain. Environ. Model. Softw. 172, 105902. http://dx.doi.org/10.1016/j.envsoft.2023.105902.

Taylor, P., Cox, S., Walker, G., Valentine, D., Sheahan, P., 2013. WaterML2.0: development of an open standard for hydrological time-series data exchange. J. Hydroinform. 16 (2), 425–446. http://dx.doi.org/10.2166/hydro.2013.174.

The HDF Group, 2024. HDF5 API reference. https://docs.hdfgroup.org/hdf5/v1_14/. (Accessed: 23 February 2024).

Villarroya, S., Viqueira, J.R.R., Regueiro, M.A., Taboada, J.A., Cotos, J.M., 2016. SODA: A framework for spatial observation data analysis. Distrib. Parallel Databases 34 (1), 65–99. http://dx.doi.org/10.1007/s10619-014-7165-7.

Viqueira, J.R.R., Lorentzos, N.A., Brisaboa, N.R., 2005. Survey on spatial data modelling approaches. In: Manolopoulos, Y., Papadopoulos, A., Vassilakopoulos, M. (Eds.), Spatial Databases: Technologies, Techniques and Trends. Idea Group, pp. 1–22.

Viqueira, J.R.R., Villarroya, S., Mera, D., Taboada, J.A., 2020. Smart environmental data infrastructures: Bridging the gap between earth sciences and citizens. Appl. Sci. 10 (3), http://dx.doi.org/10.3390/app10030856.

Wojda, P., Brouyère, S., 2013. An object-oriented hydrogeological data model for groundwater projects. Environ. Model. Softw. 43, 109–123. http://dx.doi.org/10.1016/j.envsoft.2013.01.015.

Yutzler, J., 2024. OGC GeoPackage Encoding Standard. Open Geospatial Consortium Inc., OGC Encoding Standard.. http://www.opengis.net/doc/IS/geopackage/1.4.