Ecological Informatics 24 (2014) 261–272

Contents lists available at ScienceDirect

Ecological Informatics

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

A conceptual model for data management in the field of ecology

Javad Chamanara *, Birgitta König-Ries [1]

Heinz Nixdorf Endowed Chair for Distributed Information Systems, Ernst-Abbe-Platz 2, 07743 Jena, Germany

ARTICLE INFO

Article history:
Received 14 December 2012
Received in revised form 4 October 2013
Accepted 8 December 2013
Available online 14 December 2013

Keywords:
Conceptual model
Scientific data model
Biodiversity domain model
Standardization of ecological data models
Ecological data management

ABSTRACT

Conceptual models play an important role in identifying the domain under study and establishing an interoperability framework between different scientific groups and tools working on the same or neighboring domains. The importance comes from the fact that the conceptual models describe the target domain in a technology agnostic manner, using domain terminology, considerations, and rules.

In this paper we introduce a highly flexible data and metadata structure for biodiversity (and related fields) information management. The model incorporates important concepts needed to develop a proper domain model for managing biodiversity data, e.g., data, data structure, metadata, metadata structure, and semantic descriptions of model elements. The model is designed in UML using the object oriented analysis paradigms. The data management teams of several large collaborative projects as well as those of two research institutes were actively cooperating in the design of the model, thus ensuring that all aspects relevant for these very different projects and institutions are considered and that a high acceptance of the model will ensue.

The model supports and encourages reuse and sharing of different elements, making the cross dataset syntheses, comparison, merging and searches easier. The incorporated semantic package helps to annotate dataset's variables and metadata attributes by means of ontologies, taxonomies or thesauri. These annotations can be used for standardization, localization and also for managing the variety of meanings of same or similar variables among community members.

The model is currently undergoing its implementation phase and will replace the model used in the current version of BExIS, a data management platform for biodiversity research, when finished.

1. Introduction

The rapid loss of biodiversity over the last few decades threatens the survival of humankind. Ecology in general and biodiversity research in particular can help to understand the causes of this loss and help find ways to mitigate it. To deal with these issues, environmental sciences like ecology are getting more and more data driven, and data brings computers and software into any kind of solution. Like any other scientific and engineering task, environmental data processing needs a well-defined model of the data. The model can act as a basis for data gathering, description, and annotation as well as a communication medium between computer specialists, ecologists, and the scientists of other related communities.

Scientific data needs to be captured, transferred, processed, and interpreted for immediate use, as well as stored and managed to support future reuse. The value of data increases when all researchers within a community are able to share and interact with each other's knowledge (Michener and Jones, 2012).

Valle (Valle, 2012) has broken down the subject of scientific data management in ten areas. Among them the following can be named: Creation of logical collections which tries to abstract the physical data into logical collections, physical data handling in order to establish a mapping between the physical and the logical data views, persistence in order to define the data lifetime and deployment of mechanisms to counteract technology obsolescence, interoperability support to data location autonomy and putting various data collections together, security support for data access authorization and change verification, data ownership to define who is responsible for data quality and meaning, metadata collection, management and access, knowledge and information discovery, data dissemination and publication.

Thus, there should be a shared and accepted understanding to bring all the cooperating parties onto the same page. To achieve this common understanding and facilitate agreement, a way to capture the domain requirements including all major data generation and consumption functions is needed. The result needs to be understandable by its audience, i.e., scientists from the different communities involved, and needs to be sufficiently formal to avoid misunderstandings and differing interpretations. We believe that a conceptual model is the right vehicle to achieve this.

A domain (conceptual) model is a type of model used to make explicit the structural elements and their constraints within the domain

* Corresponding author. Tel.: +49 3641 946444.
E-mail addresses: javad.chamanara@uni-jena.de (J. Chamanara), birgitta.koenig-ries@uni-jena.de (B. König-Ries).
[1] Tel.: +49 3641 946430.

1574-9541/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.ecoinf.2013.12.003

of interest.[2] A domain model includes various entities, their attributes and relationships, plus the constraints governing the integrity of the elements comprising that domain.

Conceptual models play an important role in identifying the domain under study, especially in software development efforts. They can act like a contract making it easy to establish an interoperability framework between different scientific groups and tools working on the same or neighboring domains. The importance comes from the fact that the conceptual models describe the target domain in a technology agnostic manner, using the domain terminology, considerations, and rules (Fowler, 1997).

Defining a conceptual model for biodiversity data management is challenging as ecology by its nature is a multidisciplinary domain of science. The range of required sources of data spread over different fields in earth and life sciences and recently computer science. Ecologists use multi-dimensional data (Kattge et al., 2011a) especially when they are going to perform complex analyses on large spatial or temporal scales or addressing broad questions.

In this paper, we present a conceptual model for a biodiversity research information system that shows what are the requirements of such a model, what are the elements of the solution and how they satisfy different aspects of scientific data management.

We believe this paper to be useful for a variety of audiences: First, anyone looking for a data management system for their project will be supported in identifying all relevant aspects and in judging whether a particular implementation meets their needs. Second, the published model may guide those groups who are deciding to develop their own models/tools. And finally, the model provides an insight into the web-based software system, BExIS, that we develop based on this model. So reading the paper helps in deciding to choose the software as a scientific data management tool.

In this paper we first explain our motivation in Section 2 and review related work in Section 3. Section 4 explains remarkable requirements and the features that should be considered in the development of the model. The proposed model is described in Section 5 in the form of UML[3] diagrams supported by text descriptions and finally the conclusion is provided in Section 6.

1.1. Motivation and requirements

In order to answer the "big" questions of ecology, over the last decade, ecological research has changed from small scale studies to a strong focus on synthesis and integration of data. Research in ecology increasingly relies on the integration of small, focused studies to produce larger datasets that allow for more powerful, synthetic analyses (Madin et al., 2007). Consider as an example the Biodiversity Exploratories, a large scale German project, which aims at understanding the relationship between biodiversity of different taxa and level, the role of land use and management for biodiversity, and the role of biodiversity for ecosystem processes. Clearly, no single project can even attempt to answer these questions. Instead, data from many sub-projects of different disciplines collected over a long period of time needs to be synthesized in order to address these challenging problems. (Birkhofer et al., 2012) is an example of such a work. Another example of integrating small scale studies resulting in a synthesis work is done by (Lachat et al., 2012), in which data from 988 trap catches from 209 sites in 7 European countries has been integrated in order to study the relationship between beetles, dead-wood amount, and temperature. The TRY database (Kattge et al., 2011b) as another example, is a plant trait database for quantifying and scaling global plant trait diversity. It provides a global archive for plant traits and supports the design of global vegetation models. The repository contains about three million trait entities for 69000 plants from 93 contributing databases. Researchers are able to extract local project based data from the TRY database and conduct larger scale synthesis and analysis works.

In addition, a clear need for comprehensive new programs to energize synthesis is declared by (Carpenter et al., 2009) in order to accelerate pure and applied advances in ecology and environmental sciences. Standardization of methods, development of robust metadata, reproducibility of analyses, and executable workflows are recommended by (Reichman et al., 2011). All of this emphasizes the need for a joint understanding. However, the multidisciplinary characteristic of the ecological data (Carpenter et al., 2009) results in *multidimensional data*. Temporal, spatial, taxonomic, institutional, and observational aspects are just a few of these dimensions. Observations are complex data structures consisting of data elements such as measured value and unit, measuring device settings, precision and calibration, environmental and independent variables and so on. Different research projects have numerous study purposes causing high diversity of the types of variables required resulting in highly variable data schemas.

Furthermore, dealing with scientific data is not just about storing it somewhere and providing a unique identifier for later access. Generally, data progresses through different stages, in its life cycle, in order to be converted into information or knowledge and be published (Michener and Jones, 2012). Scientific data should be *discoverable*, in addition to being *identifiable*, which means the meaning of data items and their relationships should be described in a machine readable manner. They should be equipped with some supporting information even if it is not part of the main subject of the study, i.e., project, location, climate data at the time/location of the study. These kinds of *metadata* allow for composing combined queries containing criteria from different perspectives of the data, i.e., GIS data, project data and attributes of the primary data.

Describing the *semantic relationships* between the names and the meanings of different domain and non-domain related terms like entities and measured characteristics will add a valuable capability towards automatic reasoning and more in-depth data discovery.

Data curation is a common task which applies to any scientific data especially in collaborative long-term projects. It affects primary data in two major categories; *data modification* and *quality improvement*. The first is more about the changing, adding, formatting or standardization of, i.e., units of measurements while the latter is more focused on improving the quality of the current data by providing, e.g., identify missing values, remove outliers, annotate data, perform semi or fully automated statistical quality controls and so on. These two curation procedures bring *data versioning* and staging into the scope of the problem. The staging or quality improvement (QI) in turn introduces workflow into the domain as in different projects or organizations, scientists use different QI procedures.

Scientific data should be *published* like other publishable materials such as scientific papers, designs, multi-media contents, etc. (Chavan and Ingwersen, 2009). Published datasets need to be and stay identifiable over a relatively long time period. So any modeling effort should take the integration to identification and archiving mechanisms into account, too, although these features seem to fall in the scope of data centers or long term archives.

The need for such a highly flexible data and metadata structure led us to develop a domain model which is described in this paper. The model incorporates the most important concepts needed to develop a proper domain model for managing biodiversity data, e.g., data and data structure, metadata and its structure, and their semantic description. The model is designed in UML using the object oriented analysis paradigms. The data management teams of the Biodiversity Exploratories,[4] Jena Experiment[5] and MPI-BGC Jena[6] were actively cooperating in the design of the model, ensuring almost all aspects relevant for

[2] http://en.wikipedia.org/wiki/Conceptual_model_(computer_science), visited September 2013.
[3] Unified Modeling Language, http://www.uml.org/
[4] http://www.biodiversity-exploratories.de/
[5] http://www.the-jena-experiment.de/
[6] http://www.bgc-jena.mpg.de/

these different projects and institutions are considered and that a high acceptance of the model will ensue.

Although the model is designed with the requirements of BExIS[7] in mind, it is generic enough to be incorporated in other neighboring domains and any other domain whose main data acquisition is based on observations and measurements standard (INSPIRE Cross Thematic Working Group on Observations, Measurements, 2011).

1.2. Related work

During the literature study we have found numerous tools supporting different subdomains of ecology from various perspectives. Most of them were designed to fulfill their projects' needs and relying on data conversion and transfer to/from other tools for the rest of the functionalities. This may work for individual projects or small research groups, but the problem will grow with the size of the user community, the degree of collaborations between scientists, the number of projects the scientists are working on, the complexity and variety of the requirements over time and the interoperability with other tools. In this section, we describe the results of our survey on existing systems supporting ecological research having more common and generic design. Also those tools that are used in multiple communities or cover multi-disciplinary domains are investigated. All the works mentioned here have inspired the development of our model.

By utilizing BExIS (Lotz et al., 2012) scientists and researchers with different research topics are able to store their research data in a common repository. Research works can be organized as small individual projects that PhD students perform towards their thesis or can be long term regional of federal projects involving many researchers from different institutes.

In BExIS, datasets are composed of primary data and metadata. Primary data is a collection of "observation" entities so that each observation record is a set of values related to a specific observation. Metadata is a hierarchical structure of related metadata attributes and their corresponding values. The metadata contains attributes such as ownership, location, time, and project and so on, as well as the structure of the data in the associated observation entities. The data structure introduces the list of variables, so that each variable at least has a name, data type and a description. The observations and the metadata are stored in XML format in complying with predefined XML schemas. BExIS keeps track of all editing and deletions of the observations of datasets by means of a versioning mechanism. Submitted data to BExIS can be the result of surveying, observation, processing, or simulation and so on as long as it is organized as a tabular dataset, a matrix or an unstructured file based data. A matrix in BExIS is a data structure like a dataset having an additional variable as the row indicator. Data items in the matrix are cells referenced by the row indicator and one of the column variables.

Diversity Workbench (Weiss et al., 2012) is a modular system that manages different aspects of specimen data management. In this system collected *specimens* are described by some *attributes*. In addition, they are related to *projects* as any specimen may be included in more than one. *Agents* are people or groups responsible for the collection of a specimen. The procedure of collecting specimens can be described by *events*. They can be organized as collections or as a series of collections. Also events can store *properties*. Any specimen can have parts or duplicates, stored in different places. The specimens, their parts or duplicates undergo some analyses and the *results* are recorded.

BEFdata is a software platform providing support for interdisciplinary data sharing and harmonization for collaborating distributed research projects (BEFData, 2012). BEFdata allows the harmonization of naming conventions by generating *category* lists based on the primary data. It provides a secure environment during on-going analysis. Ordinary data management features such as importing and exporting data, exporting metadata to standard formats, and editing functionalities are available. In BEFdata data is organized as a spreadsheet (BEFData, 2012). The sheet is decomposed into its sheet-cells at the database level so that each and every single primary data value is stored independently in a table row. Values have imported and accepted versions to provide a mean of curation. They are associated with a data type and a category in addition to their data column. A collection of columns then establishes a dataset. In order to keep track of matching values of same rows, sheet cells stores their corresponding row number. Columns are connected to data groups which have a set of related categories. This interconnection provides a base for sheet cells to be associated with a category among the available set. Categories can be used as a basis for developing semantic services on datasets, columns and primary data values (Nadrowski et al., 2012).

TRY is a relational database trying to standardize trait definitions and their associated attributes in addition to observation and measurements of those traits on a global scale (Kattge et al., 2011a,b). It proposes a generic structure for those plant trait databases in which *Observation* is the central table of the conceptual framework. Each observation can be characterized by *measurements* in n dimensions (traits and ancillary data). Each measurement is characterized by a value, precision, characteristic, and a measurement standard (as proposed by OBOE). Traits and ancillary data are defined in the *Characteristic* table. Measurements on the same object in time are aggregated to an observation. Observations are embedded in a hierarchy of observations on different levels. Each Observation is related to a real world object, called entity. In the presented database structure, all of the data are identically treated as measurable characteristics of specific objects, whether they are primary or ancillary.

DataBank is a web-accessible database system designed for canopy researchers to integrate database technology into their research (Cushing et al., 2002). It simplifies data sharing within close collaborations and facilitates data archiving. By providing the building blocks for database design and to use metadata source tables, it tries to make field data documentation easier. DataBank introduces so called "templates" as commonly recurring domain-specific data structures to act as building blocks for new databases. These templates can be used for importing data into a warehouse, and for composing cross study queries. DataBank databases are designed for single use during fieldwork and analysis. It is not a replacement for existing archives such as the canopy crane site databases and the LTER repositories. It spans over several sites in contrast to canopy crane databases. Also it differs from LTER repositories by providing specialized services for one community and provide help in research design, although its metadata requirements are compliant with data deposition at LTER sites.

Metacat (Berkley et al., 2001; Biocomplexity (KNB), T.K.N., 2013a) as part of the Knowledge Network for Biocomplexity (KNB) is an open source, multi user, web based XML based metadata and data repository for ecology and environmental, but not limited to, sciences. Metacat decomposes the XML dataset to its constituent nodes and stores them in a relational database table in addition to indexing them in another table in order to make information search easier and faster. It allows for maintaining local autonomy over metadata and datasets while permitting data sharing with others through a Metacat Server. Metacat supports one and bi-directional replication schemes between collaborating servers, automatic EML[8] document processing and full support of the DataONE[9] Member Node interface. It also can be customized to use Life Science Identifiers to make every data record uniquely identifiable. In combination with Morpho (Biocomplexity (KNB), T.K.N., 2013b) scientists are able to create metadata, locate authorized datasets on Metcat Servers and view them, as well as sharing their data through the KNB. Morpho is a software and Data Package is an entity/term used in the context of Morpho. The data files are in tabular form having

[7] Biodiversity Infromation System, see Related work

[8] http://knb.ecoinformatics.org/software/eml/eml-2.1.1/index.html

[9] http://www.dataone.org/

rows and columns. The metadata can act as a documentation of the whole package or fine grained annotation of tables, rows, or columns.

An important category of tools affecting data life cycle modeling are workflow management systems. They are used in data processing pipelines, task orchestration and data or process distribution, data consumption in different formats and data production or result visualization as well as data provenance.

Kepler (Kepler, 2012) is a scientific workflow management system which is able to ingest data from different sources, process them by using ready-made or user defined components, or external data processing facilities and also to output the result in forms of data and visualization. Kepler uses an *actor* metaphor to model the steps of workflows. Currently it has more than 350 different actors to accomplish data ingestion, processing, analyzing and visualizing in different scientific areas like ecology, electrical engineering, mathematics and statistics. The application of Kepler in ecology is discussed by (Pfaff et al., 2012).

VisTrails is an open-source scientific workflow and provenance management system that provides support for simulations, data exploration and visualization (VisTrails, 2012). It is more focused on the documentation of change than the repetition of workflow instances. VisTrails benefits from a provenance infrastructure that maintains a detailed history of steps followed and data derived in the course of an exploratory task. This information is persisted as XML files or in a relational database. It allows users to navigate workflow versions, to undo changes, to visually compare workflows and their results, and to examine the actions that leads to a result. It allows the combination of loosely-coupled resources, specialized libraries, grid and Web services. It is extensible by means of contributed packages.

Taverna (Taverna, 2012) is an open source scientific domain independent workflow management tool suite to design and execute workflows. It is able to fetch data from local and remote resources through provided or custom services. It provides provenance functionalities according to Janus (Missier et al., 2010) and Open Provenance Model (OPM, 2012). Taverna tries to provide a common model for workflows and means for sharing and reusing them across the borders of individual working groups. To leverage the existing infrastructure, the computational model strongly focuses on web-services, so one of the major building blocks is a web-service registry. It provides an API as well as a web interface to access data about various web-services. This data includes functional, operational, profile and provenance attributes. A more complete list of standards, institutes, tools and software is compiled in (Hernández Ernst et al., 2010).

1.3. Proposed model

The fundamental purpose of the paper is to compile an integrated conceptual model which illustrates all the structural information elements from all different components of a generic ecology data management. The model details required objects, their attributes and interrelationships, but it does not describe the implementation decisions or guidelines. In addition, it is neutral towards the distribution of the packages or elements among different software components, tools or machines. Instead, it is more about what should be considered in the implementation of data management organizations or systems and leaves the "how to do it" to the implementers. They can be all implemented in an integrated software or be separately implemented as data management, search and discovery, publishing and archiving, for example.

O&M (INSPIRE Cross Thematic Working Group on Observations, & Measurements, 2011) introduces the notion of *observation* as an event whose *result* is an estimation of the value of some *property*(*ies*) of a *feature-of-interest*, obtained using a specified *procedure*. It allows for observation and data modeling, meaning that its users not only are able to describe features and properties but also to organize and store data. OBOE is an ontology to describe the observations using its core components; *observation*, *entity*, *characteristic*, *standard*, and *protocol* (Madin et al., 2007). Regardless of their design purpose, these forms of modeling are basically similar when they are used for data modeling, and can be easily realized in a tabular format in that, the rows are the observations, columns are the characteristics/properties and standards and protocols can be modeled as annotations to the columns.

The model that we describe in this paper complies with O&M for data modeling and uses the concepts of OBOE for semantic description purposes. In this section, first we introduce high level packaging and the role of each package, and then we describe every package of the proposed solution in more detail. The packages are identified by categorizing various data lifecycle patterns such as (Valle, 2012) and (Michener and Jones, 2012) and grouping of data items by their relevance and the amount of interrelationships.

The central concept of the model shown in Fig. 1, is the *data* package which contains datasets and relies on the *data structure* package in order to define the organization and the meanings of the data. Each dataset has the ability to be described by a set of metadata attributes. Metadata has its own structure, too as described in the *metadata structure* package.

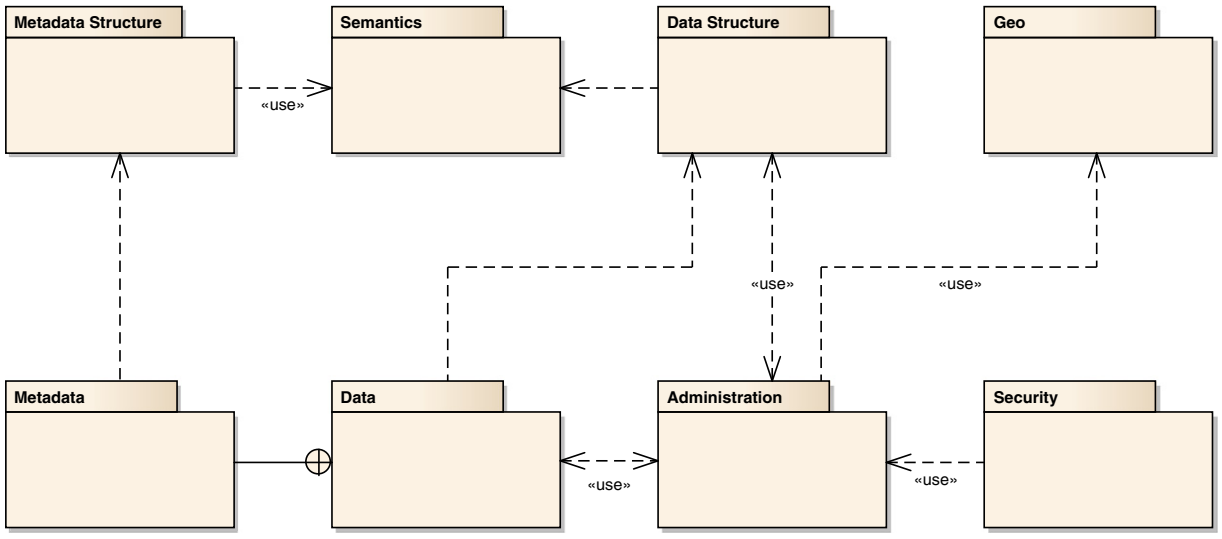

Fig. 1. The high level concepts of the conceptual model.

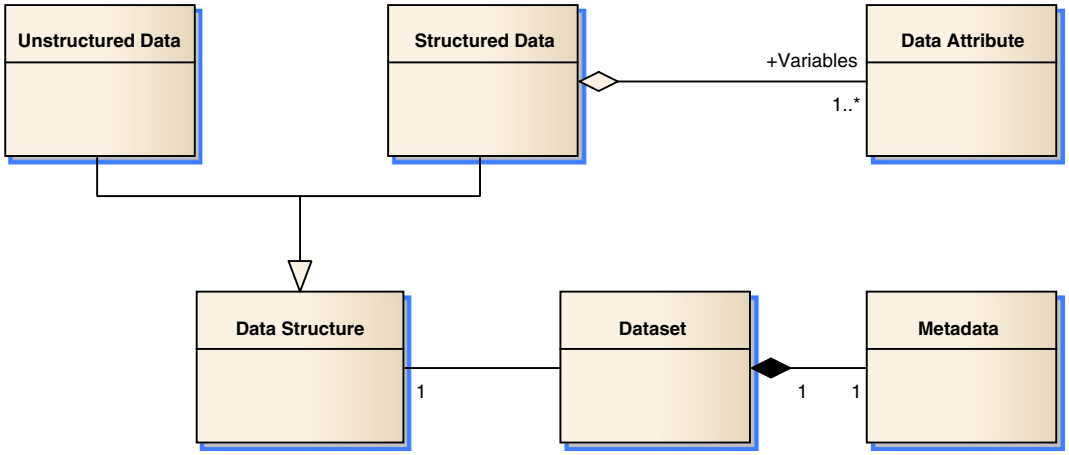

Fig. 2. An overall view on main classes of data and data structure packages.

The data structure package utilizes the *semantics* package in order to annotate its data attributes with semantic information e.g., ontology terms, taxonomies, thesauri or controlled vocabularies. These semantic annotations establish a ground for the synthesis, multi dataset analyses, and cross dataset search. The *administration* package provides contextual information regarding the management, ownership, institutional policies, locations, activities, and projects to both data structure and data packages.

1.4. Data components

Collected scientific primary data is held in a *dataset*[10] which is the heart of the model. Every dataset must comply with a specific *data structure*, which makes it possible to understand, process, or manipulate its content. In addition, the datasets have their own metadata. Based on the nature of the primary data, the datasets are divided into two main sub types, *structured data* and *unstructured data* as shown in Fig. 2. Structured data are in a tabular form with columns of variables and rows of coherent data tuples. A tuple can be interpreted as equivalent to the result of an observation.

Unstructured data does not mean that the data is really unstructured! Instead it means that the structure of the data is not known or is not of interest to the system. In both cases the implication is that no special processing or manipulation is going to be done on these kinds of data. By putting unstructured data into the model, scientists are able to store the actual primary data files out of the model and provide the model with enough information to access the data. For example, if a scientist needs to store a NetCDF file in the system (a software that has implemented this model), she is able to create an unstructured dataset, extract some of the NetCDF file's header information and put them in the dataset's metadata, and store the file itself to a local or remote storage provider. Finally, she should provide the address of the stored file as an instance of the *content descriptor* class. In this example the primary data of the dataset contains no information.

Structured datasets must be associated to a *data structure*. As depicted in Fig. 3, a data structure is a set of so called *data attributes* where each data attribute is the formal definition of one of the data columns in the dataset. Data attributes can play the role of variables or parameters depending on how they are used in a specific data structure. A *variable* is considered to be something that its value may be changed or manipulated during an experiment. Parameters help scientists in recording auxiliary information about the variable values, e.g., location, time and environmental conditions. Each variable and its associated parameters can be seen as a compound measurement. For example, in an experiment a scientist may need to measure soil nitrogen capacity 10 to 30 cm below the surface. For this purpose she measures the soil nitrogen capacity as the variable. In addition, she may record the actual depth, the humidity and the time as auxiliary data in order to understand the measured soil nitrogen capacity better.

Having the flexibility of designing data attributes in advance and using them in different data structures as variable or parameter allows data structure designers to create a set of highly reusable generic data attributes, draw more clear boundaries between independent and dependent variables, and reduces the total number of required data attributes.

Thanks to the *data container* base class as depicted in Fig. 4, every data and metadata attribute is able to have its own *data type*, *measurement scale*, list of *constraints*, *methodology*, *processing functions*, *unit* of measurement and *globalization* information. By associating with the *semantic description* coupling class, the data containers can be semantically described. The *extended property* class is designed to handle the need for custom attributes.

The many-to-many relationship between *data container* and *constraint* class allows the data structure designer to define or reuse default and missing values and validators to apply to the data and metadata attributes. By applying the *domain value* class it is possible to restrict the values of a (meta) data attribute to a predefined list of items, e.g., “Group A”, “Group B”, “Group C”, “Group D”, “Group E”, and “Group H” as domain (valid) values for a Köppen classification based climate variable. Also, it is possible to restrict the value of the containers by imposing one or more *validators* on them. The validators can control different aspects of data such as formats (like a date as mm/dd/yyyy for the US or dd.mm.yyyy for Germany), contents (like a number greater or equal to zero, min = x, max = y) or comparisons (e.g., Sample name must be alphanumeric starting with a leading English letter). Applying these kinds of constraints make controlling the quality of data easier. The constraints can be provided internally or they can be provided by an external source like a DBMS or a web service. External providers are considered in case the valid values of variables, parameters and metadata attributes change very often or are provided by other applications.

It is notable that every data container object can be a *value container* or a *reference container*. A reference container is a container that instead of holding the actual data, holds a pointer (reference) to the data. For example the “Owner” attribute can be a reference to a user of the application or a person defined outside. Reference containers help in data reuse and integrity so that by deleting their values in a dataset, the actual entities (in this example user or person) do not get affected. Also by updating the referenced entity all the references keep pointing to the right object.

[10] Note that we italicize classes to distinguish them from more general concepts, e.g., *observation* (in italic form) denotes a class in the model whereas ‘observation’ denotes its usual meaning.

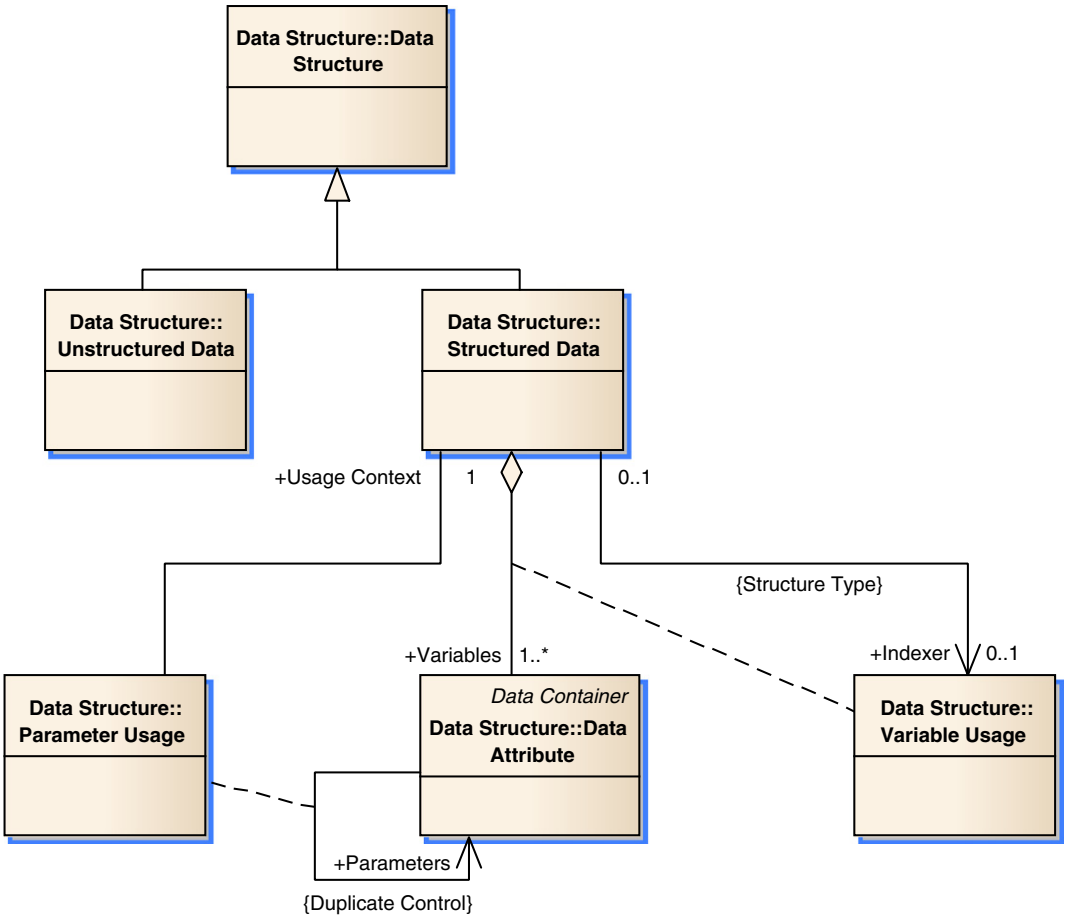

Fig. 3. Data structures and the usage of data attributes in the role of variables and parameters.

Beside the core structure, three levels of customization are also available. The first level is provided by the e*xtended property* class (Fig. 5), which allows defining some additional properties for data attributes allowing them to have different values in different datasets. For example, every researcher may like to provide a more familiar local name for variables she is working on.

In some cases, scientists may need to add extra values to a tuple, but not to all. They may even need to add different types of information to different tuples. The *amendment* class is designed to provide this second level of customization. It contains the value that is to be amended to the chosen tuple. As shown in Fig. 5, it is possible to amend any tuple with as many amendments as needed. The only restriction is that amendments can be associated only with parameters, since variables are normally connected to other parameters making amendments more complex.

V*iews* as shown in Fig. 6, are the third customization level to reduce the amount of visible data horizontally and vertically. *Variable selection criteria* determine visible columns of the target dataset and *content selection criteria* restrict the tuples. Although one of the usage scenarios of views can be security and information sharing, it can also be used in other scenarios, e.g., when the scientist needs to focus on some parts of the data, or to process a small portion of the whole dataset.

A *spanning view* is a special type of view that spans its criteria over all datasets belonging to a designated data structure. In other words, in cases that data from more than one dataset is supposed to go under the same filtering and/or projection, spanning views are used. Spanning views need to bind to one (and only one) data structure in order to be able to apply variable and content selection criteria effectively.

The view class is linked to the data structure class, which is the base of structured and unstructured data structures. This way it is possible to create views on both types of data structures and then it is the responsibility of the processing application to interpret the criteria and produce the actual view output. Cutting a spreadsheet to a range like A1:D20 or choosing “from second 15 to second 65” from a recorded visual observation are examples of applying views on unstructured data. As mentioned before the model itself has no intention to process unstructured data, but applications developed on top of this model or external tools can obtain enough information from the metadata and these view specifications in order to filter the datasets in proper ways.

1.5. Metadata components

The *dataset* class and its related classes contain no information about attributes like owner of data, project, involved researchers, and so on. Instead they are intentionally designed to focus on modeling the data and no more. These kinds of extra information are mainly modeled by means of metadata. Metadata can vary by individual users, projects, institutions, and publication purposes. So instead of conforming to any special metadata and/or data schema standard like EML, DarwinCore,[11] DublinCore[12] or ABCD,[13] we have decided to create a flexible metadata structure that is able to cover almost every standard or user defined metadata schema.

Fig. 7 shows that any metadata belongs to a metadata structure. *Metadata structure* is a hierarchical class which is able to contain as many as needed *metadata packages*. Metadata packages are aggregations of *metadata classes* and *metadata attributes*. A ‘*metadata class*’ is an entity that groups highly relevant metadata attributes together e.g., personal

[11] http://rs.tdwg.org/dwc/
[12] http://dublincore.org
[13] http://www.bgbm.org/TDWG/CODATA/Schema/default.htm

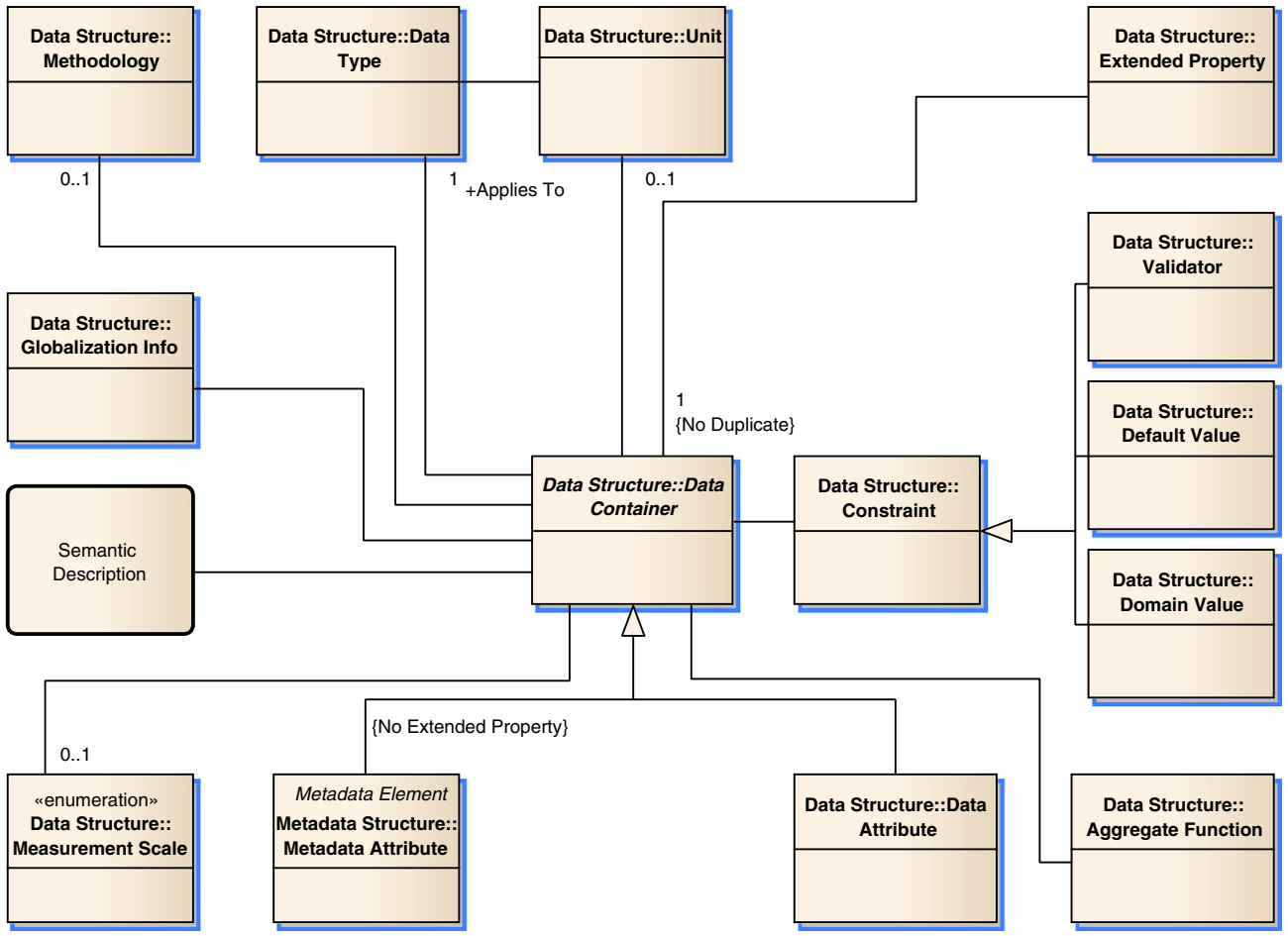

Fig. 4. The data container class and its capabilities.

information, authorship, contact information. The *metadata attribute* class defines what kind of data is considered to be used as metadata for datasets. The actual value of the attribute is stored in the *metadata attribute value* class. When a metadata structure is associated with a dataset, the structure hierarchy and all relevant packages and classes are traversed and a list of attributes is compiled. The user should provide value for all the required attributes, although s/he can provide data for optional attributes, too.

The *metadata attribute* class is a subtype of *data container* class which provides it with data type, constraints, unit, and semantic description and so on. Having semantic descriptions of metadata attributes, enables them to take part in semantic queries. Researchers can find similarity between attributes' values and have more power with metadata search capabilities to discover desired datasets.

1.6. Semantics component

Inspired by (Madin et al., 2007) a special *semantic package* is designed in order to describe data and metadata attributes in a semantic way. This semantic information can form ontologies, taxonomies or thesauri to be used to consolidate variables, parameters and attributes that have similar meaning, described in different ways or in different languages.

Based on Fig. 8, an *ontology* is a set of *terms* and their interrelationships which one of the terms act as the root. Ontologies can be hierarchical to allow ontology consumers to connect to sub parts, if they do not need the whole. The central concept in the ontology package is *term* class which describes a phrase in the target domain. A term can describe an "Entity" like bird, tree, worm, etc. It can describe a "Characteristic" of an entity or to be a generic entity e.g. diameter of a tree or just diameter. Also a term can describe a "Unit". Normally units apply to characteristics of entities, but here the ontology designer is free to define them independently or in association with characteristics. To make these terms more meaningful, they can take part in relationships together with other terms. The general pattern of these relationships, in accordance with RDF,[14] is (Subject → Predicate → Object) so that subject and object are two terms and predicate shows the meaning of the relationship between them, e.g. Bird → "is an" → Animal. The predicate is modeled as a *term*, too. This allows predicates to act as a subject or object for other predicates. For example it is possible to describe statements like "is a → is equal to → is an" or "is a → is equal to → is a subset of". The package allows realizing taxonomies, controlled vocabularies, thesauri, and any other hierarchical or graph based structure.

1.7. Administration component

Usually, research activities are performed in the context of projects. Projects are funded, supported or managed by organizations, institutes, agencies, and so on. In addition, ecology related research activities are usually related to geographical areas. The role of administration package as shown in Fig. 9, is to provide managerial, spatial, and administration information contexts for the datasets.

The *observation unit* class specifies the geographical extent that measurements are done at. It can be a very large exploratory, a surveying site or a small plot. *Work package* class models the entities like project, sub project and any important activity. Indeed it describes the effort

[14] http://www.w3.org/RDF/

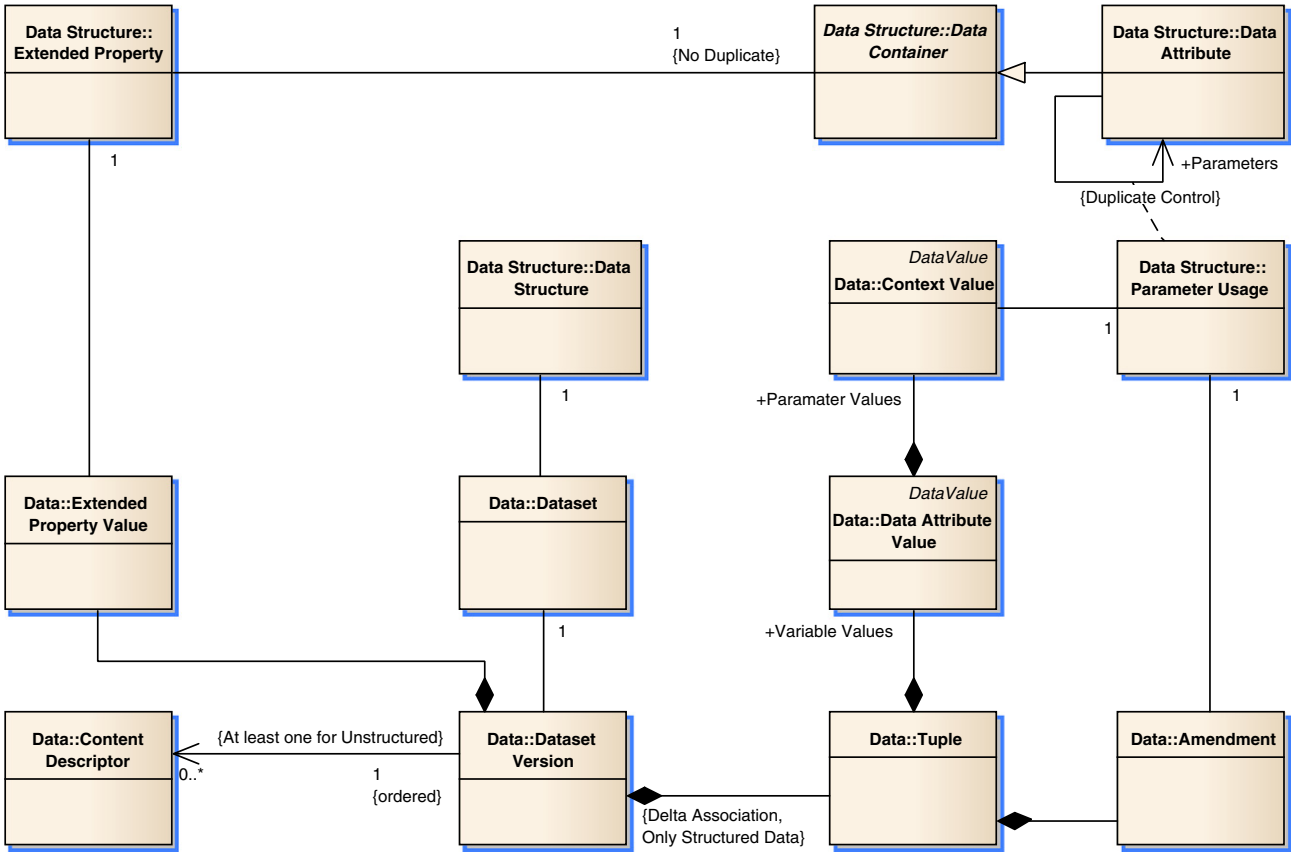

Fig. 5. Organization of the data in a dataset version.

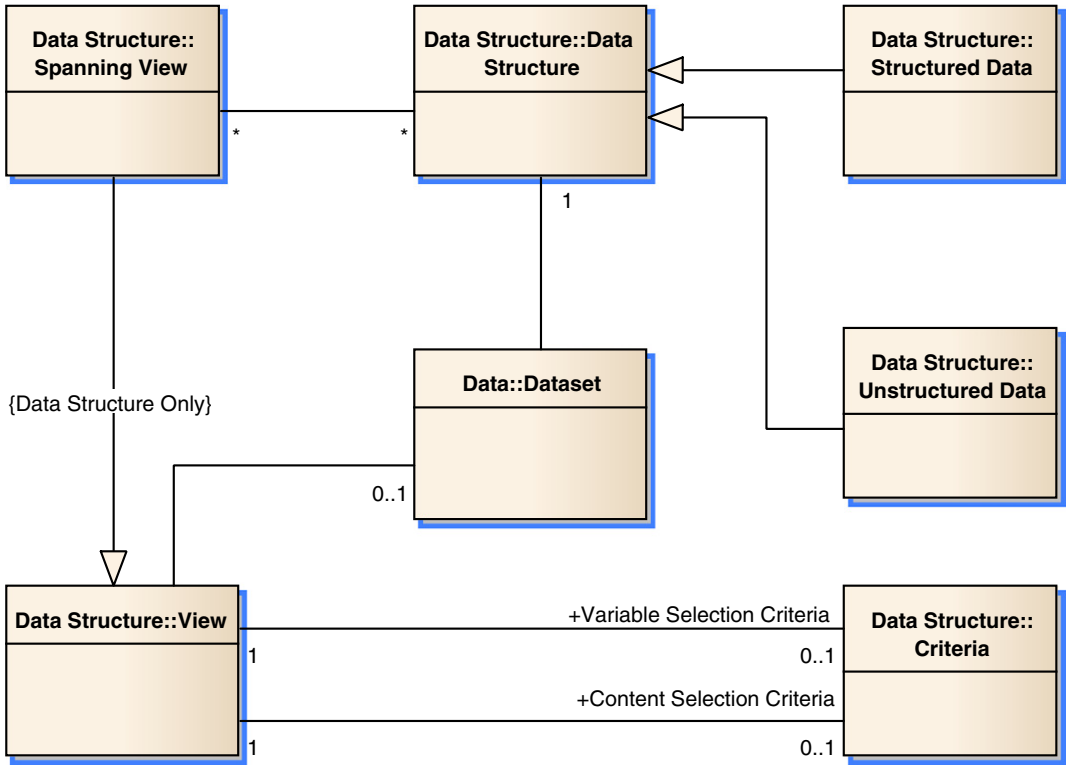

Fig. 6. Views with their content and variable filtering criteria.

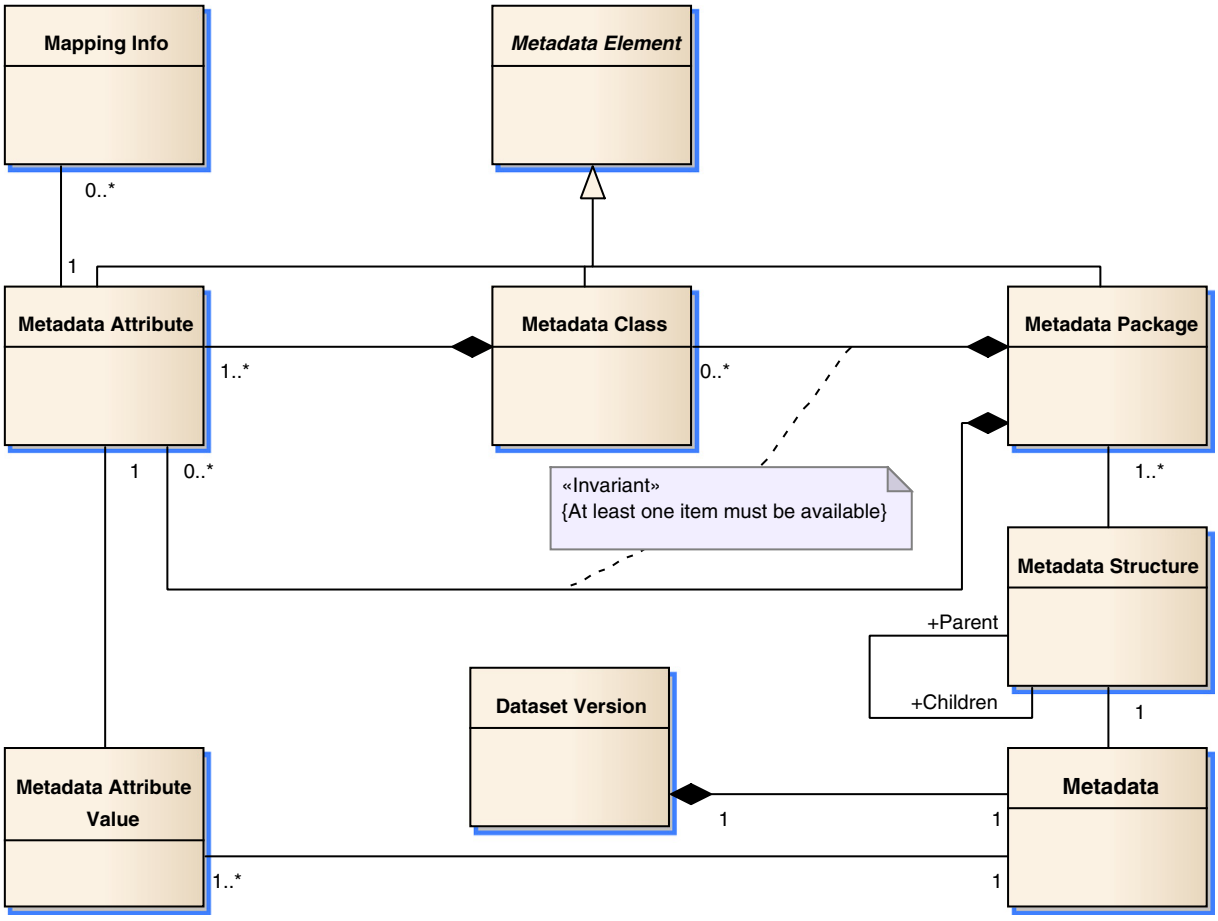

Fig. 7. Metadata structure and its relationship with metadata and datasets.

that is (has to be) done according to the research goals. *Administration unit* class encapsulates the information about institutions, organizations, agencies and any other entities having a role, e.g., administrative, managerial or controlling, in the research.

As shown in Fig. 9, every *dataset* can be associated with any number of *execution units* which enable datasets to have their full context described. The association class on the execution unit's self-relationship allows having interrelationships between administration

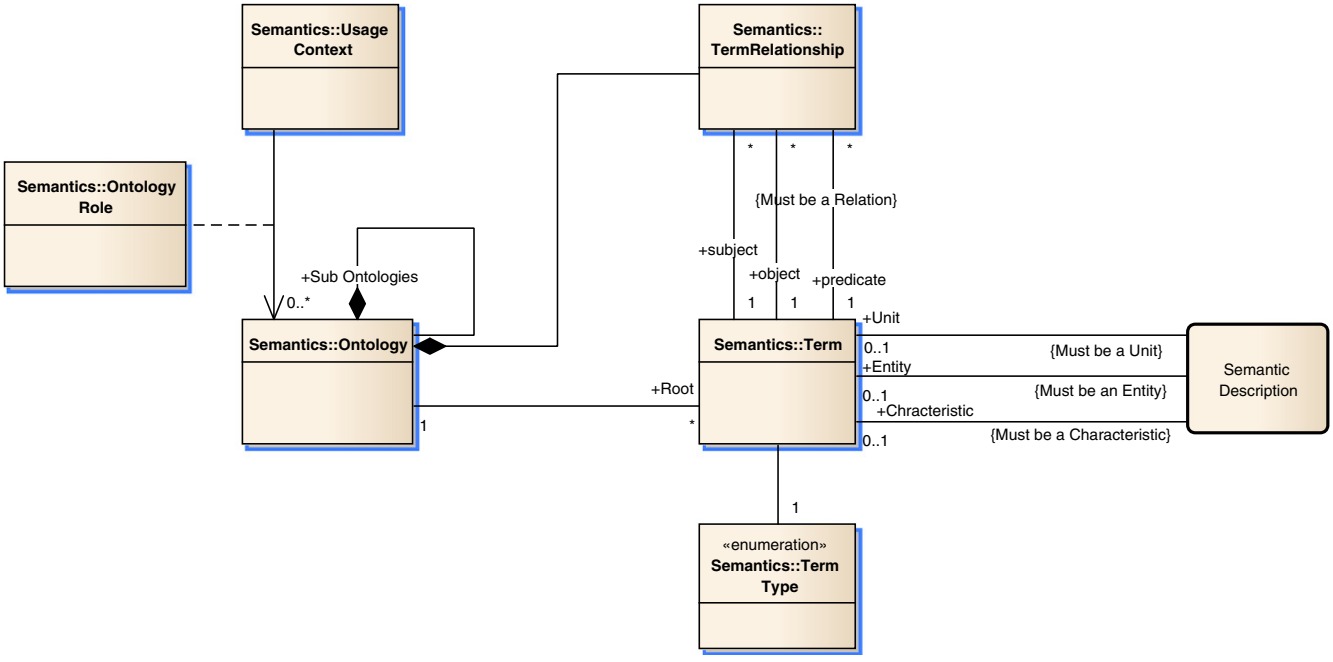

Fig. 8. Semantic package which provides a base for compiling taxonomies, controlled vocabularies, and thesauri.

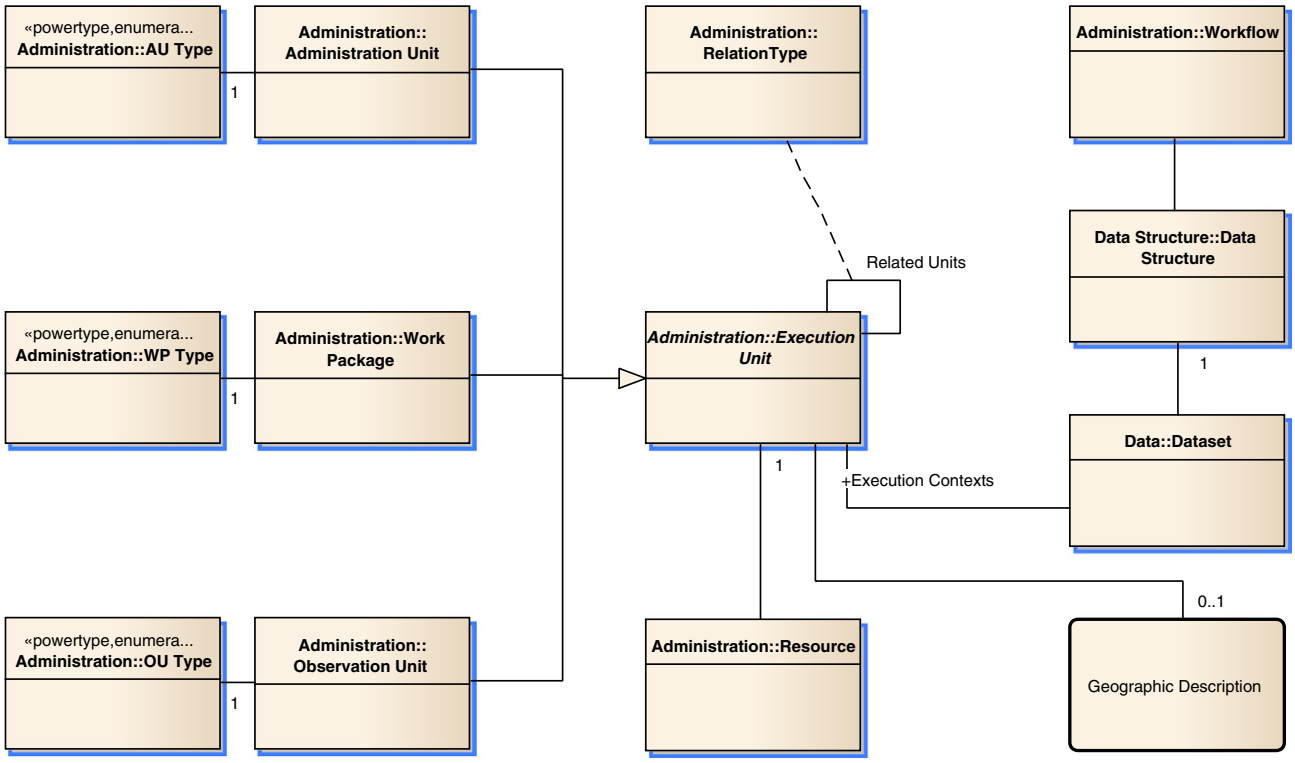

Fig. 9. Administration package contains elements to model projects, plots, organizations, and their relationships.

units, work packages, and observation units. This way it is possible to have a hierarchy of single type units, e.g., projects/sub projects or to create a graph of different unit types, e.g., a connection between Project 1 which belongs to Institutes 2 and 3 and works on Sites A, B, and C.

The *Research Plan* class is designed to specify and apply copyright, publishing, maintenance, and other types of policies on datasets belonging to their designated execution unit.

1.8. Versioning

The diagram shown in Fig. 10 illustrates the application of versioning and staging to datasets. Versioning means changes to the primary data do not overwrite each other; instead they create new versions and put previous ones aside. Versioning follows the check-out, change and check-in pattern. When the dataset is checked out, all the tuple additions, editings, deletions and re-orderings go to a newly created version. Untouched tuples of the previous versions are also re-referenced to the checked out version. Finally the check-in operation commits all the changes and makes the checked-out version visible to the consumers. Users are able to apply multiple changes for multiple times between any check-out/check-in sequence. The history of the changes is maintained in addition to the timestamp and actions applied to each tuple. This way comparing different versions of a dataset, going back and forth among the versions, or running processes against previous versions are easier to perform.

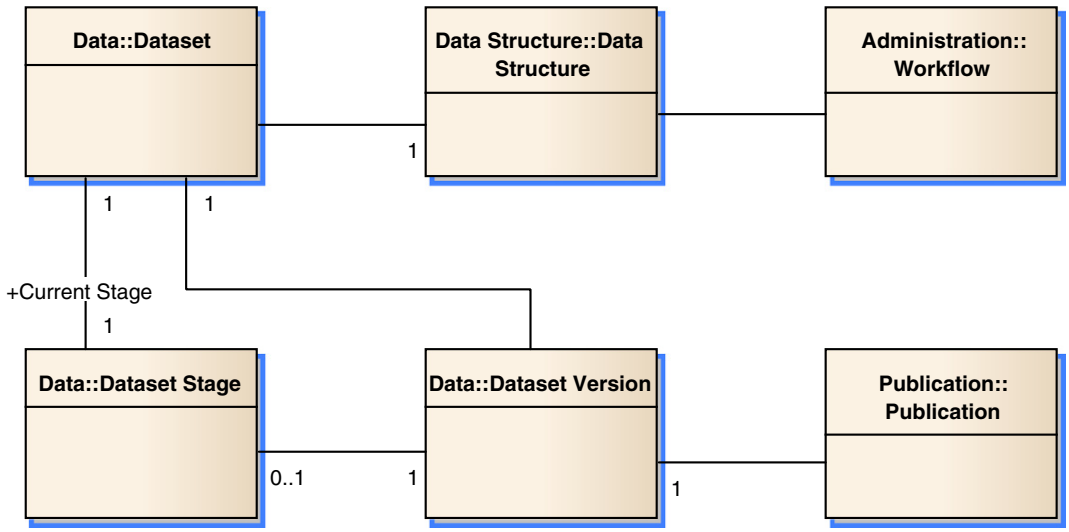

Fig. 10. Classes to manage versioning, staging and publication of datasets.

Staging is more focused on the quality of data. If data undergoes a review process for example, the quality of the resulting data is supposed to be higher. If data is processed using some algorithms, the resulting data should be marked in some way. Staging helps scientists to create, manage and query distinguishable versions of datasets. Also processes can be aware of these stages and consume specific/desired ones. Indeed having a new stage means a change has happened in the data, at least in the quality, therefore any new stage creates a new version, too. This fact is shown in Fig. 10 by drawing a 0.1 to 1 association between *dataset stage* and *dataset version*. In addition, the relationship between *dataset* class and *dataset stage* shows the current stage of the dataset, which in fact is the stage of its latest version.

Introduction of workflows is a proper way to manage staging in a more controlled and systematic manner. A workflow can control and facilitate the steps that a dataset should follow. After any successful completion of the workflow steps, the staging information can be (semi) automatically created or being inferred from the workflow execution history. *Workflows* are associated with *data structure* class forcing belonging datasets to follow the flow of activities determined by them. In case of existence of more than one workflow for a data structure, scientists can select the proper one based on their needs and requirements.

1.9. Publication and citation

Datasets can be published to external repositories and/or catalogs but they may change over time. In order for the datasets to be permanently and uniquely identified, instead of the dataset itself, one of its versions is published and this is the reason why in Fig. 10, the *publication* class is linked to the *dataset version* class. Linking to a specific version makes it possible to cite the published version so that the citation info and the published version remain unchanged, regardless of the later changes applied to the dataset. By the model, dataset versions can be published multiple times, probably in different repository and/or catalog systems. When published, information about the repository, publishing date, published content (metadata and/or primary data), publishing format, DOIs and etc. are stored in the attributes of the *publication* class instances. Consumers, e.g., papers, and derived works, can cite a *publication* instance directly or through repositories or catalogs by referring to its identifier. Having a *publication* cited, it is guaranteed that the exact same data in the same stage is accessed during the lifetime of the dataset. Basic legal effects of publication are managed by *research plan* and *policy* classes, but we have not deeply investigated this area yet.

1.10. Security

The security package is designed to manage authentication and authorization. The authentication is out of scope of this modeling effort as it is more an application level concern. Authorization relies on "Subject → Permission → Object" predicates, so that the subject can be a user, a role or a group, the object is "data" or "action". The data in the security context can be datasets' primary data, metadata or views and actions can be features of the application. Permissions define the access type to the object in a specific time interval in a specific scope. The scope determines contextual entities like projects, organizational units, geographical areas and so on. Scopes can be put together by means of logical operators to create fine grained scopes. As an example, User u1 can have write access to all datasets belonging to Project P1, delete access to all datasets belonging to Plot PT1 which belongs to Project P1 from March 1, 2013 to February 28, 2014 and read access to other datasets without restriction.

2. Conclusion

In this paper, we have presented a generic data model for almost all types of primary data that adhere to O&M. A complete HTML version of the model is available at: http://fusion.cs.uni-jena.de/bppCM/index.htm. We believe this model to efficiently support ecological research endeavors.

Since the model supports and even encourages sharing of variables and parameters among data structures, in contrast to existing systems performing cross dataset syntheses and searches becomes easier. Having a set of shared variables in multiple datasets, helps data analysts to compare and merge them. Variable sharing can be done before, during or even after dataset submission, considering data consolidation is taken into account. It is a useful tool for small to medium size teams that are able to define a set of data attributes and reuse them in different combinations. In addition to this first level data design, by incorporating the semantic package, it is possible to annotate variables, parameters and metadata attributes by means of ontologies, taxonomies or thesauri. These annotations are useful for semantic search, standardization, localization and also for managing the variety of meanings of data containers.

Although we have incorporated versioning and workflows into the model, they need more investigation. In the case of workflow, we are able to rely on well-known workflow management systems, but a detailed customized versioning mechanism would be an integral part of the model; which we are currently working on.

It is worth to mention that the model does not imply its possible implementations must consider all the packages. In particular, the publication and citation for long term repositories can be out of the scope of a project based data management software.

We are currently implementing the model as part of the new version of BExIS. There, by mixing the relational logic with XML at the implementation level we obtain a high degree of flexibility at the database level and improve the stability of the physical design against the variety of data and metadata structures. Once the implementation is completed and the system is being used by a number of projects, it will become possible to thoroughly evaluate the contribution. Given that the model was developed with significant input by a number of very different projects, we are confident, that this evaluation will be very positive.

Acknowledgments

The work described in this paper is done in the context of BExIS++ (BExIS++, 2011) project. BExIS++ is funded by DFG (German Research Foundation) within the LIS (Scientific Library Services and Information Systems) program.

References

BEFData, 2012. The data model of BEFData. (Available at: https://github.com/befdata/befdata/tree/master/app/models).
Berkley, C., et al., 2001. Metacat: a schema-independent XML database system. Scientific and Statistical Database Management, 2001. SSDBM 2001. Proceedings. Thirteenth International Conference, pp. 171–179.
BExIS++, 2011. BExIS++, Biodiversity Exploratory Information System. (Available at: http://fusion.cs.uni-jena.de/bexis).
Biocomplexity (KNB), T.K.N, 2013a. Metacat: metadata and data management server. (Available at: http://knb.ecoinformatics.org/knb/docs/).
Biocomplexity (KNB), T.K.N, 2013b. Morpho. Available at: http://knb.ecoinformatics.org/software/morpho/MorphoUserGuide.pdf.
Birkhofer, K., et al., 2012. General relationships between abiotic soil properties and soil biota across spatial scales and different land-use types. PLoS ONE 7, e43292 Available at: http://dx.doi.org/10.1371%2Fjournal.pone.0043292.
Carpenter, S.R., et al., 2009. Accelerate synthesis in ecology and environmental sciences. Bioscience 59, 699–701.
Chavan, V.S., Ingwersen, P., 2009. Towards a data publishing framework for primary biodiversity data: challenges and potentials for the biodiversity informatics community. BMC Bioinforma. 10 (Suppl. 14), S2 (Available at: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2775148).
Cushing, J.B., et al., 2002. Template-driven end-user ecological database design (SCI2002).
Fowler, M., 1997. Analysis patterns: reusable objects models. Addison Wesley.
Hernández Ernst, V., et al., 2010. Data & Modelling Tool Structures—Status Report on Infrastructures for Biodiversity Research.

INSPIRE Cross Thematic Working Group on Observations & Measurements, 2011. Guidelines for the use of Observations & Measurements and Sensor Web Enablement.
Kattge, J., et al., 2011a. A generic structure for plant trait databases. Methods Ecol. Evol. 2, 202–213.
Kattge, J., et al., 2011b. TRY—a global database of plant traits. Glob. Chang. Biol. 17, 2905–2935.
Kepler, 2012. Kepler Scientific Workflow Management System. (Available at: https://kepler-project.org/).
Lachat, T., et al., 2012. Saproxylic beetles as indicator species for dead-wood amount and temperature in European beech forests. Ecol. Indic. 23, 323–331.
Lotz, T., et al., 2012. Diverse or uniform?—intercomparison of two major German project databases for interdisciplinary collaborative functional biodiversity research. Ecol. Inform. 8, 10–19.
Madin, J., et al., 2007. An ontology for describing and synthesizing ecological observation data. Ecol. inform. 2, 279–296.
Michener, W.K., Jones, M.B., 2012. Ecoinformatics: supporting ecology as a data-intensive science. Trends Ecol. Evol. 27, 85–93 (Available at: http://www.sciencedirect.com/science/article/pii/S0169534711003399).
Missier, P., et al., 2010. In: McGuinness, D.L., Michaelis, J., Moreau, L. (Eds.), anus: From Workflows to Semantic Provenance and Linked Open Data. 6378, pp. 129–141 (Available at: http://dx.doi.org/10.1007/978-3-642-17819-1).
Nadrowski, K., et al., 2012. Identifiers in e-Science platforms for the ecological sciences. Virtual Enterprises, Research Communities & Social Media Networks.
OPM, 2012. The Open Provenance Model (OPM). Available at: http://openprovenance.org/.
Pfaff, C.-T., Nadrowski, K., Wirth, C., 2012. Using Kepler workflows. Ecology, 3. F1000 Posters, p. 1356.
Reichman, O., Jones, M.B., Schildhauer, M.P., 2011. Challenges and opportunities of open data in ecology. Science (Washington) 331, 703–705.
Taverna, 2012. Taverna Workflow Management System. (Available at: http://www.taverna.org.uk).
Valle, M., 2012. Scientific data management. Available at: http://mariovalle.name/sdm/scientific-data-management.html.
VisTrails, 2012. VisTrails Open-Source Scientific Workflow and Provenance Management System. (Available at: http://www.vistrails.org/index.php/Main_Page).
Weiss, M., Hagedorn, G., Triebel, D., 2012. DiversityCollection information model. Available at: http://www.diversityworkbench.net/Portal/CollectionModel_v2.05.17.