Noumea: a Model Driven Framework for NetCDF-CF Data Extraction and Analysis

Jean-Philippe Babau, Lab-STICC/UBO (France), babau@univ-brest.fr
Jannaï Tokotoko, Lab-STICC/UBO (France), tokotokojannai@yahoo.fr
Oumar Kande, Lab-STICC/UBO (France), oumarkabde15@yahoo.fr
Abdallahi Bilal, Lab-STICC/UBO (France), abdallahy.bilal@gmail.com

Introduction

To improve interoperability, scientific dataset modeling follows abstract standards like the Unidata's Common Data Model (CDM) [1]. To implement CDM, NetCDF [2] proposes a set of software libraries and self-describing, machine-independent data formats. NetCDF supports the creation, access, and sharing of array-oriented scientific. Because CDM and NetCDF are generic purpose model and libraries, scientists use specific standards like Climate and Forecast (CF) [3] or OceanSITES [4]. The standards propose constraints on the metadata declarations to facilitate file exchange and their manipulation.
To check if NetCDF data conforms to a standard, a code-oriented checker is used classically. Thus, the constraints are not formalized and a modification in the standard results in a manual modification of the code. Moreover, when a file does not conform to the standard, the file correction requires manual modifications or another specific tool. Another aspect is that, if NetCDF appears as a standard for storing scientific data, other formats are used for specific applications (sensor acquisition, specific analyzing tools). And for a huge volume of data, solutions based on Big Data technologies have to be investigated. To finish, for data following standards and stored in an efficient way, data visualization and analysis require other dedicated tools.
Face to the implementation complexity, to reduce implementation cost, some geospatial content management systems (CMS) appear (see GeoCMS [5]). In the same idea, this paper proposes to investigate a model-based approach. Modelling is a new challenging approach of software engineering to master the complexity of platforms and to express domain concepts effectively [6]. The idea is to propose adapted model for each data management phase and then to replace the coding activity by a configuration activity, as in a CMS. The project is ambitious considering the following aspects:

- Format verification
- Standard application
- Data storage
- Data visualization and analysis

In this paper, we present first experiments showing the interest of the proposed approach.

Format verification

OceanSITES is a system for gathering and measuring scientific data especially for time series sites. The OceanSITES User Manual holds around 30 pages of constraints expressed in natural language. These constraints are of different forms: naming conventions, possible attribute values, constraints on dimension length and many others. As an example, a CDM *DataSet* should hold instances of *Dimension* called *TIME*, *LATITUDE*, *LONGITUDE* and an instance of *Variable* called *TIME*. The *Variable TIME* should be of *double* datatype. To check the data conformity to OceanSITES, a Java tool already exists [7]. Since constraints are not formalized and the tool is hand-coded, there is no guaranty on checking. Furthermore, for each standard, a particular tool should be developed. To avoid constraint edition ambiguity and to reduce conformity tool development, we proposed a dedicated rule-based *CdmCL* language [8].

Format correction

The conventions for CF metadata are designed to promote the processing and sharing of files created with NetCDF. CF proposes constraints in a less restrictive way than OceanSITES. CF is based on guidelines proposing different recommended approaches to express metadata. For example, an attribute axis may be attached to a coordinate variable and given one of the values X, Y, Z or T which stand for a longitude, latitude, vertical, or time axis respectively. Alternatively, the standard_name attribute may be used for direct identification. To consider variants on constraints, we first extend the *CdmCL* language. Then we propose a tool able to extract from a given NetCDF file, the potential coordinate variables (depending on axis, standard_name and unit). The User Interface proposes helpers to choose and to correct, if necessary, coordinates attributes, following the standard recommendations.

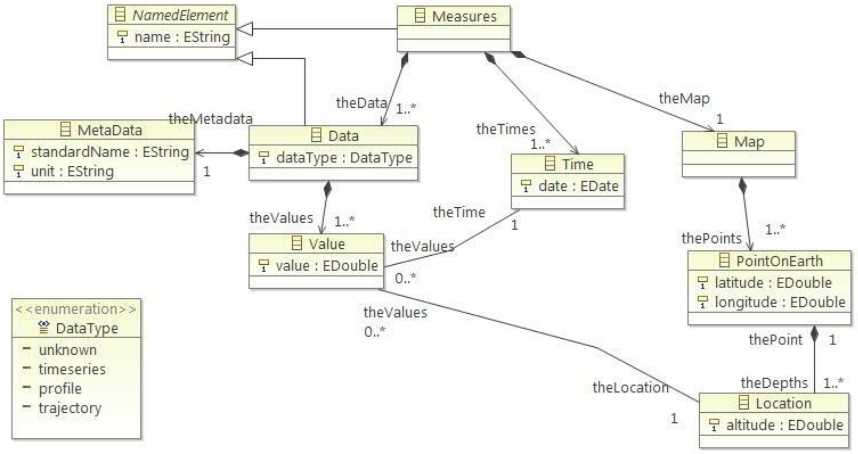

Figure 1: the proposed generic spatiotemporal data model

Generic format and data storage

After correction, we consider data following the proposed unified object-oriented model of spatiotemporal data (see figure 1). Data are then stored locally or using a Big Data technology. For this purpose, we reuse the Kevoree project technology [9]. From object-oriented models, Kevoree offers an interface for object persistence and an API to store and to load objects from a data store. Different driver implementations allow to choose the most appropriate data backend for the application (local, database or big data technology) in a transparent way.

Tools

For data analysis tools, we propose to develop specific tool-oriented data model, considering the relevant part of the generic spatiotemporal data model, and tool-oriented parameters. For example, for a time series visualization tool, the *Map* information is reduced to one specific *Location*, and the model is enriched with User Interface parameters, as classically with a CMS. The model is then instantiated for each application. From a model, it is then possible to generate a dedicated visualization tool. For our experiments, we reuse classical visualization JavaScript libraries.

Conclusion

The paper proposes a model-based approach to improve tools development. We target tools dedicated to checking, correction, analyzing, storage and visualization of scientific data. The different proposed models are standard-independent and technology-independent. We experiment the approach on two different standards, using different implementation technologies. We are now working on extending the approach to provide new analyzing tools and to consider new applications.

References

[1] Unidata, Common Data Model www.unidata.ucar.edu/software/thredds/current/netcdf-java/CDM/
[2] NetworkCommon Data Form *http://www.unidata.ucar.edu/software/netcdf/*
[3] The Climate and Forecast Conventions and Metadata http://cfconventions.org/
[4] OceanSITES User's Manual, Version 1.2. http://www.oceansites.org/
[5] GeoCMS https://github.com/dotgee/geocms
[6] G. Mussbacher and all "The Relevance of Model-Driven Engineering Thirty Years from Now" Chapter of Model-Driven Engineering Languages and Systems LNCS Vol 8767, pp 183-200, Oct 2014
[6] OceanSITES file format checker. http://www.coriolis.eu.org/Observing-theocean/ Observing-system-networks/OceanSITES/Access-to-data.
[7] A. Ahmed, P. Vallejo, M. Kerboeuf, J.-P. Babau "CdmCL, a Specific Textual Constraint Language for Common Data Model » 14th International Workshop on *OCL and Textual Modelling*Valencia, Spain. 2014
[8] The Kevoree Modeling Framework "https://github.com/kevoree-modeling"