IS-ENES3 Milestone M10.2

CMIP Data Request Schema 2.0

Reporting period: 01/07/2020 – 30/12/2021

Authors: Martin Juckes (UKRI STFC), martin.juckes@stfc.ac.uk

Release date: 01/11/2020

Reviewers: Michel Rixen (WCRP), Matthew Mizielinski (UKMO),
Bryan Lawrence (NCAS, Uni. Reading)

ABSTRACT

The CMIP Data Request schema defines the database structure and for the CMIP Data Request.

The aim of the upgrade to the data request schema is to provide continuity but also resolve some issues in the current schema. The Data Request 1.0 schema (currently updated to 1.00.33) was released in a bundle with supporting python libraries and embedded scientific content. A primary high-level objective of the 2.0 schema is to bring into effect a clean separation between the different components.

A second high level objective will be to provide greater clarity in the interactions between the Data Request and other stakeholders. The stakeholder group includes the MIP co-chairs who are trying to translate their scientific requirements into the DR and the peer group of infrastructure providers who are developing related services which need to interact seamlessly with the DR and have consistent content.

Finally, there is a need to make it easier for groups to review the content of the request. The inclusion of features which, when linked together, create complexity which forms a barrier to reviews of the content should be avoided.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 824084

Table of Contents

EXECUTIVE SUMMARY

The aim of the upgrade to the data request schema is to provide continuity but also resolve some issues in the current schema.

The Data Request 1.0 schema (currently updated to 1.00.33) was released in a bundle with supporting python libraries and embedded scientific content. A primary high-level objective of the 2.0 schema is to bring into effect a clean separation between the different components.

A second high level objective will be to provide greater clarity in the interactions between the Data Request and other stakeholders. The stakeholder group includes the MIP co-chairs who are trying to translate their scientific requirements into the DR and the peer group of infrastructure providers who are developing related services which need to interact seamlessly with the DR and have consistent content.

Finally, there is a need to make it easier for groups to review the content of the request. The inclusion of features which, when linked together, create complexity which forms a barrier to reviews of the content should be avoided.

Results

Establishing sustainable operation able to support a wider variety of use cases may be supported by exploiting well developed concepts from relevant ISO standards governing metadata registries. These standards give insight into best practices both in terms of procedures and metadata structures.

Perspectives

It is expected that the CMIP7 request will contain only moderate extensions to the content of the CMIP6 request, but a larger expansion is expected in CMIP8. The changes implemented here are designed to be scalable to be able to accommodate the major expansion of CMIP8. The rationalisation of the data request structure also provides cleaner access to reusable elements that can be used for model intercomparison efforts outside CMIP.

1 Background and Objectives

1.1 The history and status of version 1 of the schema

The status of version 1 is described by Juckes et al. (2020) (https://doi.org/10.5194/gmd-13-201-2020).

The data request for the Coupled Model Intercomparison Project 6 (CMIP6, Eyring et al. 2016, Balaji et al. 2018) compiled data requirements from all the participating Model Intercomparison Projects (MIPs) as part of the endorsement process run by the CMIP Panel on behalf of the World Climate Research Programme (WCRP) Working Group on Coupled Models (WGCM). The aggregated data requirements were then provided to participating modelling groups as a single integrated and machine interpretable ("workflow ready") document.

There is substantial continuity between the technical content of the CMIP6 data request and the equivalent documents compiled and used in earlier phases of CMIP. Two substantial organisational changes are worth highlighting, as they affect the communication framework through which the content of the data request is agreed:

1. responsibility for the request was moved from PCMDI to the Centre for Environmental Data Analysis (CEDA), separating it from the core organisational role which PCMDI continues to deliver, as it has done from the inception of CMIP;
2. there was a substantial increase in the number of participating MIPs and a more formal approach to coordinating them through the endorsement process.

The 1.0 schema was developed as an evolutionary step from structures in Excel spreadsheets used to compile the CMIP5 data request, with extensions to cover new features requested in CMIP6 such as a statement of preferred output grids, and was provided as an XSD schema document with accompanying documentation.

It became clear during the evolution of the CMIP6 Data Request that design decisions around the database schema could not be considered in isolation from the complex network of stakeholder requirements.

A characteristic conflict occurs between downstream users of the database seeking stability and content providers seeking flexibility. The downstream users place a high value on stability of semantic structures so that design of software consuming the database can be based on secure foundations. The content providers, on the other hand, prefer flexibility to accommodate emerging requirements inspired by scientific advances. The information modelling effort will provide a framework within which to balance these requirements.

1.2 Moving to version 2 of the schema

The Data Request Support Group (DRSG) has been established, reporting to the WGCM Infrastructure Panel (WIP). The DRSG is meeting regularly to discuss priorities and the approach to implementation. A roadmap document is being prepared to provide an overview of status and progress.

The interaction between stakeholder engagement and technical requirements can be illustrated by considering the prioritisation of variables within the data request. The prioritisation is intended to guide contributing modelling centres when they tailor their contribution to match the resources they have available. The intention is that all modelling centres should provide the highest priority variables so that data users can benefit from a uniform selection of variables.

The converse, which is unfortunately common, is that a user wanting to look at a single variable may find a large ensemble of models, but as soon as they want to look at a collection of variables they find that many model only provide part of their requirements and they must either proceed with a much smaller ensemble or find ways of dealing with the omissions of variables in some cases (ie. working with a sparse data matrix).

One contributing factor here is the grade inflation which has devalued the prioritisation process to the extent that there is a significant mis-alignment between the stated intent of the variable priorities (that all relevant priority 1 variables should be provided) and the implementation (a subset of priority 1 variables).

The interactions of the different stakeholders and governing bodies associated with the data request are complex. In this document a four stage approach to describing and managing the interactions is set out, inspired by an ISO standard for representing a technical design process [2] (ISO 19508 described in more detail below). The first stage sets out the scope and domain of the activity as a whole: the details are derived from decisions of various WCRP bodies, but also influenced by the scope of the commitment from the host institution. The second stage, which, in addition to the formulation of the four-stage description, is the primary goal of this milestone, sets out the technical framework, requirements and assumptions which will form the basis of the version 2 implementation. The third stage will include the application schema, a machine interpretable technical document defining all the metadata attributes for the data request. Such a document exists for the CMIP6 data request schema, but the provenance and significance of some of the terms is unclear. For instance, some terms are tightly bound to the algorithms for generating data files, so that associated content needs to conform to expectations of software libraries, such as the CMOR library at PCDMI, which are maintained to support the implementation of these algorithms.

1.3 What will the new request look like?

With the revised structure the request from each MIP will be made up of one or more packets, with each packet specifying a list of tiered experiments and a list of prioritised variables which are requested from all of those experiments. When, as may be the case for larger MIPs, there is a need to specify different variables for different experiments, this should be done by requesting multiple packets. Different packets are expected to share many variables and may also share experiments (such as control experiments).

This revised approach is intended to avoid the complications that arose in the CMIP6 request from allowing an a la carte approach of specifying a different set of variables for each experiment. The same flexibility is technically still available as it would be possible for MIPs to request a different packet for each experiment. Although this is possible, it is not advised. For most MIPs one or a handful of packets should be enough.

1.4 Scope and layout of this document

This document provides "an information model to be used for the CMIP7 Data Request (STFC)". This information model will take the form of diagrams setting out the objectives and scope of the application schema.

The next two sections set out the context, including a new mission statement, and requirements which guide the development of the data request. Section 4 sets out terminology and is followed by section 5 setting out the approach to implementation and section 6 defining the major sections of the data request. Section 7 provides a final summary.

2 Context of the Data Request

The Data Request does not, of course, evolve independently of other aspects of the climate modelling infrastructure.

It was clear in the early stage of CMIP6 that a complex clash of expectations caused disruption to initial plans, with conflicting expectations coming from distinct groups: the infrastructure providers, the science teams in the intercomparison projects, the overall scientific coordination and the modelling groups. This clash occurred in spite of significant and detailed communication at all stages of the process.

One outcome of this clash was a data request timetable which involved putting together the initial schema in a matter of weeks. This short and rushed period of design was, in part, due to mis-communication resulting from confusion about the distinction between the definition of physical parameters and the requirement to specify precisely which variables are needed for each experiment. These two aspects of the data request raise distinctive issues. The approach to reaching agreement on physical variables is complex, but the approach used is an evolution of that taken in CMIP5 and earlier CMIP phases. The approach for defining requirements specific to each experiment was, on the other hand, a significant organisational innovation. Lack of understanding, and conflicting presumptions, about the implications of this change disrupted initial work on the data request.

The architectural design set out below aims to facilitate clearer communication about some of these issues.

When considered as part of a global enterprise, it is important to consider that simplifications in one part of the enterprise can have unexpected consequences elsewhere. The mission statement set out in section 2.1 below aims to address some of these problems by setting out, in a concise form, the main areas of activity which the Data Request Service aims to cover and the objectives they serve.

Stating these activities does not imply that there is long-term funding: the current support comes from EU H2020 through the ISENES3[1] project and continues to the end of 2022. The mission statement does, however, reflect the ideas that will be used in trying to secure continued funding.

At the start of CMIP6, some “decisions” about implementation were taken by the CMIP Panel without consultation. Supporting this kind of behaviour is not part of the mission. As noted below

1 Infrastructure Support for the European Network for Earth System Modeling for Climate (www.enes.org).

in the elaboration of the mission statement, the impact of decisions is far from clear when dealing with interactions of multiple complex infrastructure components.

2.1 Mission Statement

- Develop and maintain a registry of defined physical parameters appropriate for the analysis of climate simulations and exchange of climate model output;
- Develop and maintain a registry of file metadata specifications (including templates and parametric templates) to facilitate interoperable exchange of climate model output;
- Support WCRP-endorsed model intercomparison projects by facilitating the specification of output requirements for climate model intercomparison and evaluation efforts.

Elaboration of the Mission Statement

The approach to defining physical parameters is heavily dependent on the CF Convention, particularly the CF Standard Names, but is not completely covered by the CF Convention. The usage in CMIP requires specification of a variable short name (suitable for use as a variable name in software applications), a title (suitable for use in publications) and specific units, constraints which are not supported by the CF Convention. There are also a substantial number of physical parameters which can not be defined by a standard name alone, such as fractional land cover variables which are specified by a combination of a CF Standard Name and a CF Area Type. The registry of physical parameters covers these additional requirements and provides the interface to the CF Convention.[2]

The Data Request 1.0 contained specification of structure records which defined, explicitly or implicitly (through directives), the structure of metadata in NetCDF files. Where it is explicit, as in the specification of a cell methods string which should be inserted in the `cell_methods` attribute of a variable in a NetCDF file, it may be considered as part of a template. In other cases there are directives which can trigger a range of implementations depending on options selected by modelling groups. For instance, a spatial grid might be formulated as a regular latitude-longitude grid, or it might be a tri-polar grid. The details of implementation for the different options are currently embedded in CMOR, which implements an approach consistent with the CF Convention. In moving to Data Request 2.0, we will try to document these CMOR options explicitly to create more explicit templates which clarify the expected structure of model output. The starting point will be the structure and grid records defined in Data Request 1.0 with an evolutionary change adding more detail.

Supporting the mapping of variables and experiments onto scientific objectives absorbed a considerable amount of effort in CMIP6, not least because of the new nature of the approach to endorsing MIPs which created new complexities. One clear emerging requirement is for better communication about what the request means when many different MIPs are submitting related but

2There could be a link back to the CF Conventions through the “common concepts” idea which has been discussed but not adopted by the CF community.

distinct requests to cover related but distinct objectives. In simpler projects, for which a single list of variables are required from all experiments, this element of the request is trivial. The problem arises when many different MIPs are combined. The strength of this approach is that it allows simulations to support multiple objectives, increasing efficiency. It does however, introduce complexities in planning which are still being worked through.[3] One aspect of the CMIP6 approach was that different science projects had, as might be expected, different priorities, expressed in the Data Request as an integer priority number for each request for a variable. The fact that a variable might have different priorities in different experiments was part of the initial requirements, but caused considerable downstream problems because of the difficulty in implementing decisions in workflows at modelling centres. Once an experiment has been chosen, and a decision about MIPs to be supported is taken, it is easy enough to extract the relevant priority … but working out the implications of this approach in the context of implementing a modeling program appears to be complex.

The concept of a registry is intended to include both contents and interfaces, including web, command line and programmable interfaces.

3 Requirements

The requirements are split into two sections labelled as “strategic” for requirements which address broader issues of integration with other parts of the infrastructure and “technical” for requirements which address specific technical problems associated with the CMIP6 data request.

Juckes et al. (2020) identified four baseline requirements:

- provide feedback to MIPs on feasibility of data requests,especially regarding estimated data volumes;
- provide precise definitions and fully specified technical metadata for each parameter requested;
- provide a programmable interface that supports auto-mated processing of the DREQ;
- support synergies between MIPs, maximizing the reuse of specifications and of data.

3.1 Strategic Requirements

Table 1: Strategic requirements

ID	Label	Description
R1.1	Continuity	Avoid disruptive changes: continued support for existing APIs for CMIP6 and smooth transition to CMIP7;
R1.2	Technical	Clear up linkages with ES-DOC and CMIPX CVs;

3 The fact that a MIP might request data from an experiment designed by another MIP means that the objectives of a modelling group performing a simulation might not be driven by the MIP which designed the experiment. This distinction means that, with the metadata specifications for CMIP6, ths objectives of a simulation are not cleanly captured in the file metadata. This in turn causes some problems in creating clear documentation of model simulations in some of the automated documentation systems.

	Integration	
R1.3	Database	Database Normalisation: linkage between tables should be SQL compatible (one attribute targets at most one table);
R1.4	Coherency	Structural Rationalisation: use same semantics for linking and grouping in experiments and variables, so clients can use simpler and clearer logic; Clarify approach to groups, collections etc.
R1.5	User groups	Clarify External Dependencies: some parts of the request need broad consultation with the scientific community (mainly descriptive content), others need to be checked for compatibility with other services (especially CMOR, also other groups IPSL XIOS already integral in NEMO and being integrated in EC-Earth4 atmosphere; GFDL working on similar activity; Need to understand dependencies of other software tools).
R1.6	Complexity	Strip out excessive complexity -- e.g. data volume estimation -- which can be handled by downstream software.
R1.7	Transparency	Clarify definitions of attributes and external dependencies: hundreds of attributes used .. some have unclear meaning through lack of clarity in their purpose .. others have unclear documentation.
R1.8	Content	Over time the priority values of variables have become devalued. There are too many “top priority” variables. There needs to be a re-assessment of the priorities .. and some means of giving authority to “priority = 1” statements. Discussed at WGCM 2019 in Barcelona -- intention to reduce number of variables at priority = 1 from c. 50% to a significantly smaller number, perhaps starting with those prioritized by AG6 WG1. Schema requirement is to support documentation of priority setting.
R1.9	Community Resource	Develop and maintain a community resource supporting wider usage than just the CMIP projects.
R1.10	Interoperability	Different versions of the request should be compatible, so that the same software version can read multiple request versions.
R1.11	Traceability	Understanding changes in the request and the reasons behind them.

3.2 Technical Requirements

This list is derived from the issues in the “CMIP7 Forward Look” Github repository:

github.com/cmip6dr/cmip7_forward_look

Issues raised during the CMIP6 process which could not be resolved (for lack of time, or because any reasonable solution would be too disruptive). There are 56 issues, many related to CF Conventions implementation decisions and other aspects of content. The following 10 items relate to the schema:

1. Multiple experiment start dates [#56]
2. Clarify approach on ancillary variables [#53]

3. variable names for latitude and longitude of ocean variables are not standard -- would be easier if they were standardised. Users should use the metadata, but scientists want to rely on variable names.

4. Behaviour of CMOR is sometimes taken to be part of the standard

5. Many of the issues which have been noted with ocean grid variables/coordinates are due to data being written without using CMOR (even though a `cmor_version` is identified in the file written, e.g. CMIP5 files).

6. Clarify rules on file names to avoid clashes [#50] and approach to requests with partial overlaps. See also #1

7. Defining methods attached to some classes [#45]

8. Separating what is wanted (e.g. percentage area covered) from implementation (e.g. [#33])

9. Objectives:experiment objectives vs. data selection objectives [#31]

10. Dealing with options in cell methods strings (e.g. [#27])

11. Specifying the time of day required which sub-daily data is requested [#21]

12. Changing the vocabulary used for variable types from FORTRAN [#18]

13. Specify intervals for calculations of time mean [#9]

14. Tracking changes to variables and other reusable components [#60]

3.3 More on variable priority

In addition to the issue of inflation of request priorities, with far too many variables being listed as top priority, there was considerable confusion in CMIP6 about the interpretation of request priorities.

In CMIP5, each requested parameter was assigned a priority from 1 (high) to 3 (low), and this priority applied to requests for that variable from all CMIP5 experiments. In CMIP6 it was felt necessary to provide more flexibility, because, in CMIP6, modelling groups have the choice of which component MIPs to support. Consequently, modelling groups need to know how important the variables requested are for the MIPs that they have elected to support.

The CMIP6 data request also met a stated requirement that it should be possible for different variables to be specified for different experiments. For instance, analysis of one experiment may require detailed data on sea-ice processes while another focusses on convective storms. This requirement was met by allowing each MIP to specify both the required variables for each experiment individually, and to give a specific priority for each experiment. Expressed symbolically, this can be written as:

```
producer.priority =: get_priority(variable, mip(s), experiment),
```

where `producer.priority` is the priority that is relevant to the data producer, and get_priority is a function which evaluates this priority based on the specification of a variable, the MIPs being supported, and the experiment being performed.

This allows great flexibility and ensures that modelling groups are not confronted with requests for, for instance, huge volumes of data on atmospheric convection from experiments which are designed to analyse sea ice. However, there was a serious down-side in that many groups struggled to understand how to implement this CMIP6 approach. Part of the problem appears to be in the high level of flexibility which resulted in very fine-grain information … making it hard to manage processes and workflows. The approach proposed for Data Request 2.0 will reduce the level of granularity:

```
producer.priority =: get_priority(variable, mip(s) [, analysis_objective(s)]).
```

This makes the priority dependent on the MIPs supported and, potentially, on a subset of the analysis objectives specified by each MIP. The option of specifying multiple analysis objectives with distinctive data requirements was not used widely in CMIP5, but was important for some of the larger and more complex MIPs (e.g. HighResMIP).

The modelling groups are required to provide information to the CMIP panel about which MIPs they support, so it should not be problematic to deal with priorities which vary between MIPs. If this information, together with optional specification of analysis objectives, is provided by modelling groups through the Controlled Vocabularies, as done in CMIP6 (though not foreseen at the stage when the Data Request for CMIP6 was designed), the above function could be replaced by:

```
priority =: drq:view.priority(variable, model).
```

This approach would give the modelling groups the information they need to work on the configuration of their models in the long period of preparation that precedes the actual execution of CMIP simulations. The model specific information from the CVs would not be embedded in the request, but should be imported for use by the "view" methods. This will allow modelling groups to combine the request with the latest CVs or draft versions of updates.

Box 1: Assumptions on cv:source_id vocabulary.
In order for the data request to provide accurate and specific information about the level of participation of each model. The CMIP6 cv:source_id vocabulary contains a list of MIPs supported by each model, but does not contain additional information that was requested by the CMIP panel regarding the specific tiers, priorities and analysis objectives. We can split the decision process here into two stages: (1) Strategic decisions taken by the modeling centres should be reflected in the CVs so that the data request, along with other infrastructure, can exploit that information; (2) Options related to operational decisions should be supported by the data request user API. The split between strategic and operational is, to a degree, arbitrary, but needs to be agreed

before software components are designed.

The following assumptions guide the current design:

The CMIP CV for source_id **MUST** provide a list of MIPs supported by each model (as done for CMIP6 in the “activity_participation” list);

The CMIP CV **MAY** provide additional information about analysis objectives, tiers and variable priorities (e.g. by providing a dictionary instead of a simple for “activity_participation”).

4 Implementation: Setting out the modelling approach

4.1 Namespaces

The MDR standard supports the use of namespaces. Within this document we will use namespace notation to remove ambiguity where needed: for instance, “mdr:object” refers to the MDR class “Object”.

For terms introduced within this document, we will use the namespace tag “drq”.

4.2 The core concepts to be modelled: examples

The approach taken has been informed by a review of a range of relevant ISO standards dealing with metadata, modelling of metadata and methodologies for modelling metadata. The ISO standards reflect a substantial trans-disciplinary body of experience of considerable value, but can easily become restrictive if applied without fully understanding the context and consequences of the approach recommended or required.

The distinction between metadata, modelling of metadata and methodologies for modelling metadata is worth dwelling on. For many stakeholders and users of the data request, the metadata itself is already somewhat detached from the critical reality of the data they are interested in: what would they make of metadata and methodologies?

The metadata entities that we will be interested in can be illustrated by these examples from the CMIP6 Data Request:

- **drq:var**: A physical parameter.
- **drq:var.sn**: The CF Standard Name is part of an extensive vocabulary maintained within the CF Convention.
- **drq:CMORvar.modeling_realm**: A string that indicates the high level modeling realm which is particularly relevant. Note that sometimes a variable will be equally (or almost equally relevant) to two or more realms, in which case a primary realm is assigned as the first listed and other relevant realms follow in a space separated list.
- **drq:requestItem.tslice**: Optional link to a time slice specifier which will define a subset of

the years from an experiment;

- **drq:requestItem.tslice.title:** A short description of the requestItem.tslice concept;

Some metadata entities contain string content which should be transferred directly into model output data files, while others specify directives for the writing of files, guidance for the selection of variables which should be produced or guidance for the preparation of the data itself. Other metadata entities within the data request are associated with cross-linkages between tables. Descriptions of these different metadata roles will be given formalised through metadata modelling.

The task of describing what metadata is somewhat specialised, but there is a considerable body of literature and experience to draw on. The CF Convention, for instance, relies on a narrative document to describe terms, backed up by a more structured (and less readable) conformance document. Within the ISO family of standards there are standards both for the layout of documents and for the detailed technical specification of metadata registries. Much of the ISO machinery is far too detailed and resource intensive for the scale of operation considered here, but there is nevertheless a considerable resource of information on scalable and re-usable approaches.

4.2.1 Imported concepts

Concepts imported from the ISO 11179 standard for metadata registries will be referenced using an “mdr” prefix, e.g. “mdr:reference_document.title” for the title of a reference document, described in section 6.3.7.2.5 of ISO 11179-3.

4.3 Registers and Registries

In the ISO framework, the standards 19135 and 11179 describe the operation of metadata registries for geographic information and for general purpose metadata registries respectively. These two standards are linked to a wide range of ISO standards listed below.

4.3.1 ISO Standards relevant to the Data Request

The main pillars of the work are the standards listed here:

- ISO80000: QU Quantities and Units: Parts 1: General, 3: Space and Time, 4 : Mechanics, 5: Thermodynamics, 7: Light and Radiation, and 9: Physical Chemistry and Molecular Physics. [referenced as ISO80000-1, etc]
- ISO19115: GIM Geographic Information — Metadata;
- ISO19135: GIP Geographic Information — Procedures;
- ISO11179: MDR Metadata Registries [sections 1-7, referenced as ISO11179-1 etc; see S5.5 below];
- ISO38500: ITG Governance of Information Technology;

- ISO19439: EIF Enterprise integration Framework for enterprise modelling;
- ISO19440: EIC Enterprise integration Constructs for enterprise modelling;
- ISO19508: MOF OMG Meta Object Facility;
- ISO11404: GPD General Purpose Data Types.

The main focus here is on the OMG Meta Object Facility (MOF) which provides a layered framework for talking about complex systems, allowing detail to be built up systematically, and the Metadata Registries (MDR) standard which sets out systems for describing metadata.

The Geographic Information standards contain detailed information about how to record geographic information: this is highly relevant to the CMIP data, but deals with a level of detail which is not covered in this document. The Geographic Information standards could be considered as an implementation of MDR, though they evolved in parallel rather than sequentially. MDR allows us to build a system for describing the metadata which is outside the scope of Geographic Information standards.

The enterprise framework standards (ISO19439, ISO19440) and the Governance on Information Technology (ISO38500) standard do not have anything to say about the technical design issues of the data request, but they do provide a useful framework for describing the decision making process around the many standards, protocols and conventions. The approach appears to be compatible with parts of the MDR standard which refer to harmonisation of registry content: the idea that information in the registry should reflect a community consensus, not just the views of the individuals submitting the data. The work done on harmonisation of content in the data request is split across many different groups, e.g. the science teams behind the MIPs working on variable definitions and the technology experts assembled by the WIP working on technical implementation. These groups are not managed by the data request, but, in the interests of transparency, they do need to be described. The enterprise framework standards provide a means of describing such a network with distributed and heterogeneous decision making.

The GDP standard on data types is not exploited explicitly here, but provides background on what the concept of “data type” means within the ISO framework.

4.4 Metamodel Layers

The idea of metamodel layers is introduced in MOF, originally with reference to a 4 layer approach, though arbitrary layering is considered valid from a MOF perspective.

MOF2.0 suggests a 3-layer approach for databases, with layers corresponding to specification of database table schema, database record schema and contents. Here we add a fourth layer to enable a clear representation of not only the database, but also the procedures for maintaining the database and the software tools supporting its use.

This paper will not develop all layers in full: the focus is on the database itself. However, the

interactions with the other elements are crucial interfaces and will be defined here.

Table 2: MOF metamodel layers	
Label	Description
M3:Domain	Sets out the modeling approach and describes the implementation of MOF2. The standards that will be exploited, and the intended level of compliance will also be described here. This layer also defines the overall objectives of the undertaking: why the data request exists and what it hopes to achieve. The mission statement in section 3 is a key component of this layer.
M2:Definition	The metamodel layer. This layer defines the major divisions of the data request activity, separating out the organisational and service elements (including interactions with users) from the design of the database. Interfaces between these elements will also be defined. This layer includes a schema defining tables which instantiated in the specifications of the tables in the next layer down[4]. Information about the approach to modeling the software elements is also given. This layer includes the specification of the schema for the database tables. Base classes for the software element are also described here.
M1:Design	The model layer, including the schema for the database records. Also included are classes for software resources which are used in services or provided as ancillary tools.
M0:Delivery	This layer includes the database itself, software and technical artefacts, specifications of services, etc.

This section (#4: Implementation) provides the M3 layer specifying the modelling approach.

4.5 Exploiting the Metadata Registry Framework

The ISO 11179 standard on metadata Registries (MDR) provides an extensive range of tools for describing the contents and the operation of metadata registries. It is primarily concerned with systems for talking about metadata, rather than making specific recommendations for metadata describing user data.

This should not impose any restrictions on the way in which the schema is implemented in the application schema, but it may help in providing a clear description of the structures being created.

4 The tables could be relational database tables, but a here the will be implemented as registry tables which have more detailed semantics around the table specifications. See also Lawrence et al 2012: https://gmd.copernicus.org/articles/5/1493/2012/gmd-5-1493-2012.pdf for discussion of different approaches.

For the specification of concepts, MDR requires the specification of a concept system and a notation. For instance, the concept system could be SKOS-CORE and the notation SKOS/Turtle. There is no explicit restriction on the complexity of concept systems which can be constructed – but the requirement to express them in a reusable form will certainly put a brake on uncontrolled growth of complexity in the modelling of concepts.

This approach has the benefit, from our perspective, of separating the notation used for specification of concepts from the administrative metadata of the metadata registry/database.

Within MDR there are seven sections:

- ISO/IEC 11179-1:2015 Framework (referred to as ISO/IEC 11179-1)
- ISO/IEC 11179-2:2019 Classification
- ISO/IEC 11179-3:2013+A1-2020 Registry metamodel and basic attributes
- ISO/IEC 11179-4:2004 Formulation of data definitions
- ISO/IEC 11179-5:2015 Naming and identification principles
- ISO/IEC 11179-6:2015 Registration
- ISO/IEC 11179-7:2019 Metamodel for data set registration

Here, we will exploit ideas from 11179-3, “Registry metamodel and basic attributes”.

The concept of a “Data Element” lies at the heart of the MDR metamodel. The Data Element should contain the full description of the data expected, and a datatype specifying the type of digital data expected. The Data Element can be seen as a system of defining concepts in terms of triples of the form:

Object – Property – Range,

where the “Range” includes a data type with a description and/or a list of acceptable values.

“Object” here is “anything perceivable or conceivable” which needs to be represented. There is no intended link with the grammatical role of object versus subject.

For example,

Earth System Model Simulation – Monthly Mean Near Surface Air Temperature – Global Spatial Array.

When a value is assigned, this becomes a triple of the form:

Object – Property – Value.

The Object and Property should be represented using an MDR Concept System. The standard provides mappings from SKOS and OWL concepts onto the MDR Concept System. For the

development of the DREQ 2.0 Schema we will rely on representing Objects and Properties as SKOS concepts, and take the MDR representation as following implicitly from the known mapping.

4.5.1 Serialisation

The ISO11179 “Object-Property-Value” triples map naturally, albeit with a naming clash, onto RDF “Subject-Property-Object” triples, and this will be used in serialisation of data request concepts (discussed further in 4.6 and 6.2.6 below).

4.6 Export versus Recursion

This section looks at the difference between metadata and data. In the triple at the end of the last subsection it would be reasonable to consider the value as the data and the details in the property as metadata.

However, one person's metadata is another person's data. Here we consider the process through which a property such as “mdr:Variable.title”, which might have a value such as “Monthly Mean Near Surface Temperature”, can be considered as an object in its own right, to be described by a collection of properties.

In the Data Request schema 1.0 a recursive approach was used, such that the semantic structure used to define the property “title” was the same that used to describe variables which use “title” as a property. This recursive approach is a familiar feature of RDF. One way of looking at it is that the properties are defined in the same namespace and schema as the object they are describing.

The MDR approach is different: properties should be defined by an externally referenced schema, preferably a well known standard such as SKOS. The “Export” approach refers to the fact that, in order to comply with this aspect of the MDR, we need to create a SKOS representation of properties defined in the request in order to exploit them in registers defining variables et cetera.

The recursive, RDF-like, approach, gives an attractive sense of unity and consistency in the logical structure of the full system comprising definitions of variables and definitions of the properties used to define variables, but, at the same time, it conceals import distinctions and dependencies. The MDR approach, on the other hand, creates a clear boundary and makes it easier to describe the different governance and harmonisation processes which need to be considered when refining the definitions of properties which are themselves used to define other records.

Thus, a full registry record defining the variable title property will contain information about the change history and governance. When it comes to implementing this definition in creating variable records with the “title” attribute, an exported version can be used with a reference back to the full registry record.

The possibility of when it comes to serialising the structure by exporting it into a single namespace for downstream use remains, but the MDR approach provides a clearer mechanism to manage the construction phase and the associated traceability issues.

5 Major divisions of the data request activity [MOF:M2]

The MOF M2 layer provides the outline of the database structure in terms of the key functional components.

1.1 Data Request Functional Components

The 2.0 schema will segregate different functions in order to improve transparency in communication with stakeholders. The functional components will map onto packages in the application schema.

Table 3: Functional Components		
Package	**Description**	**Key Stakeholders and Process**
Utility Packages		
Data Types	Data Type classes used in the definition of registry classes, often with restrictions based on imported vocabularies.	Each data type will be associated with some external documentation and, in many cases, a technical document specifying terms. New terms here need careful review by developers supporting data preparation software, especially CMOR.
Aggregations	Classes representing collections of other classes which can be used to structure requests and aid reproducibility.	
Views	Classes which are constructed automatically by methods attached to other data request classes in order to provide composite structures to support backward compatibility and consistency checks.	Many views will be constructed down-stream of the request, but some mission critical structures will be included automatically as part of the distributed database.
Physical Parameters		
Physical Parameters	Clearly defined physical parameters with a CF Standard Name and a small amount of additional metadata.	The terms are intended to be used across the community, as a shared resource. New terms need to be reviewed for consistency.
File Metadata		
Structures	Combined specification of the information needed about the grid dimensions and the data on the grid.	
Configured Parameters (CMOR Variables)	Physical parameters with added information detailing how the parameter is to be stored, including the temporal frequency and processing,	New terms may be added for specific projects, but some consultation is needed to avoid duplication. Resource

	and masking. A pack of parameters represents a selection of parameters to be used for a specific purpose.	considerations are important for high frequency of high rank variables.
Grids & Coordinates	Specification of NetCDF coordinate variables, both directly and via directives.	
Aggregations	Classes representing collections of other classes which can be used to structure requests and aid reproducibility.	
Analysis Objective		
Analysis Objectives	Classes which link the Activity with Aggregations of data variables and experiments, including a specification of the objectives that the request will address.	

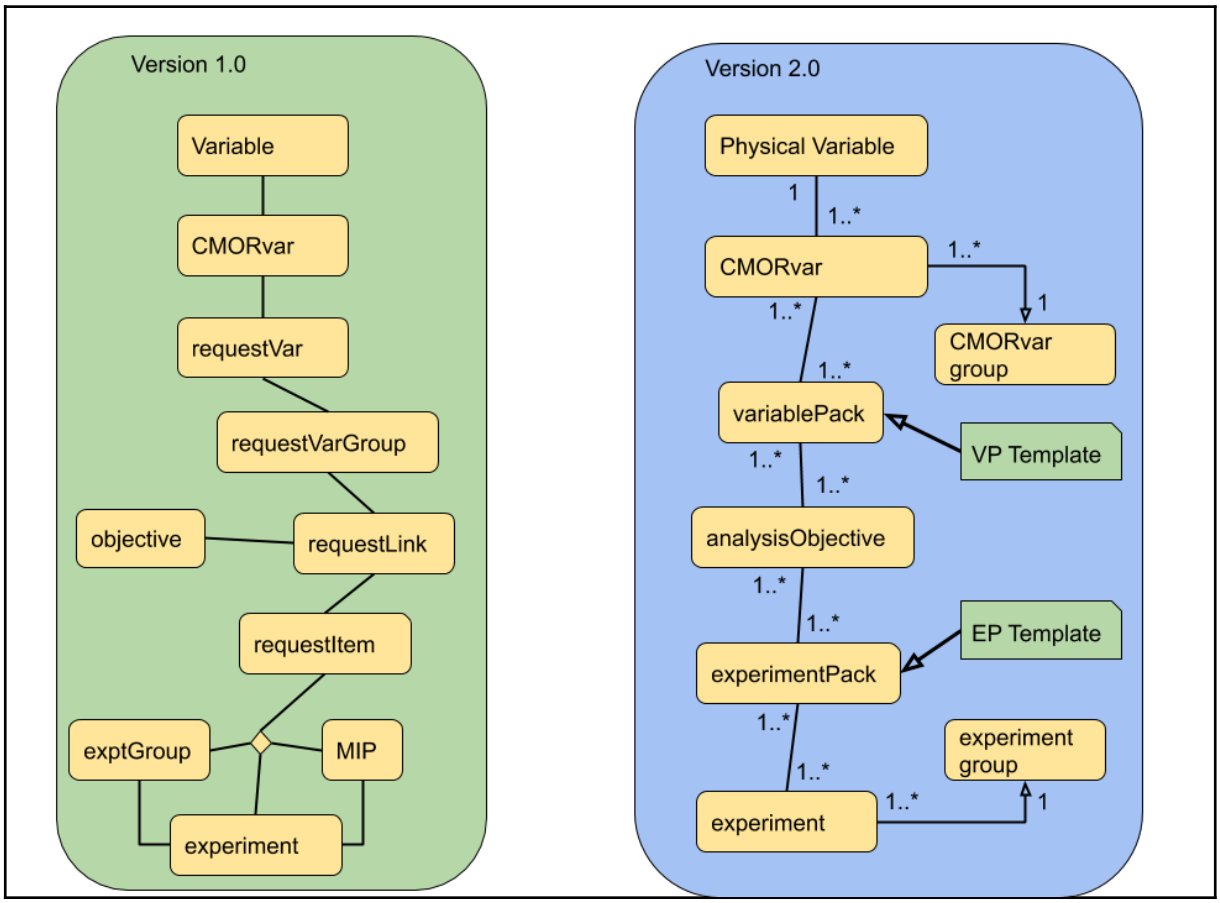
1.2 The Spine of the Data Request

The sequence of records which connect variables to experiments will be referred to as the “spine” of the request. The remaining records specify properties, provenance, intent and options.

In the CMIP6 request there are 387 requestLink records. The objective in the version 2.0 schema is to enable a sharp reduction in this number in order to create an easier route to reviewing the request.

Comparison between version 1.0 and 2.0, showing the links joining a variable to an experiment. The variablePack and experimentPack objects will have analogous structures. Both will be generated from a more general Aggregation class, as discussed in Section 7.1.2 below. Each “Pack” will have a “content()” method to return a full list of variables or experiments respectively, so that applications do not need to navigate a tree of relationships.

Figure 1: Comparison between version 1.0 and 2.0, showing the links joining a variable to an experiment. The variablePack and experimentPack objects will have analogous structures, and each will have a “content()” method to return a full list of variables or experiments respectively. Version 2.0 will include explicit representation of templates used to deliver bulk input into the data request. The “variablePack” and “analysisPack” will link directly to the analysis objective.



1.3 Classes in the Data Request Schema

Properties of classes may be specified through attributes or associations. Since an attribute may take a value defined as a class instance and an association will point to a class instance, the distinction between these two can be unclear, leading to redundancy and ambiguity in the modeling. We adopt the convention that attributes will be used with values defined by Data Types, and associations for links to other classes.

A Data Type is defined in UML to be a class whose instances are identified by their values. For example, “integer” is a Data Type and “4” is an instance which is identified by the value “4” rather than by a separate identifier. There is, however, no general restriction on the complexity that can be described by a Data Type. ISO 11404 gives examples of an address record Data Type consisting of 5 strings for name, address, city etc.

The distinction between attribute and association can also be considered in terms of the intended behaviour of the objects. If objects are intended to have a degree of independent existence they should be defined as normal class elements. The Data Type, on the other hand, is appropriate for use when a quantity is intended for use solely in combination with a specific attribute.

For example, we will consider Comment as a Data Type superclass defined with the general objective of providing information which complements a class description. Sub-classes may then be defined with a range of different attributes. For example, a MIP Variable comment will have attributes for usage and preparation.

2 Sections of the Data Request

2.1 Utility Packages

2.1.1 Data Types

ISO 11404 defines a datatype as “set of distinct values, characterized by properties of those values, and by operations on those values”.

Data Types in the will conform to the MDR standard and will be defined through 4 attributes: name, description, reference_document and annotation (optional). The RDT Data Type is defined, with attributes type_name, type_description, type_reference_document, type_annotation to avoid namespace clashes (and/or confusion in documentation) in instances of the data type which use “name”, “description” etc for instance specific information.

The Reference Document data type as defined in MDR, with some simplifying specialisations, is shown in Figure 2 below. The MDR profile requires that the Data Type be associated with a reference document which describes the type being implemented.

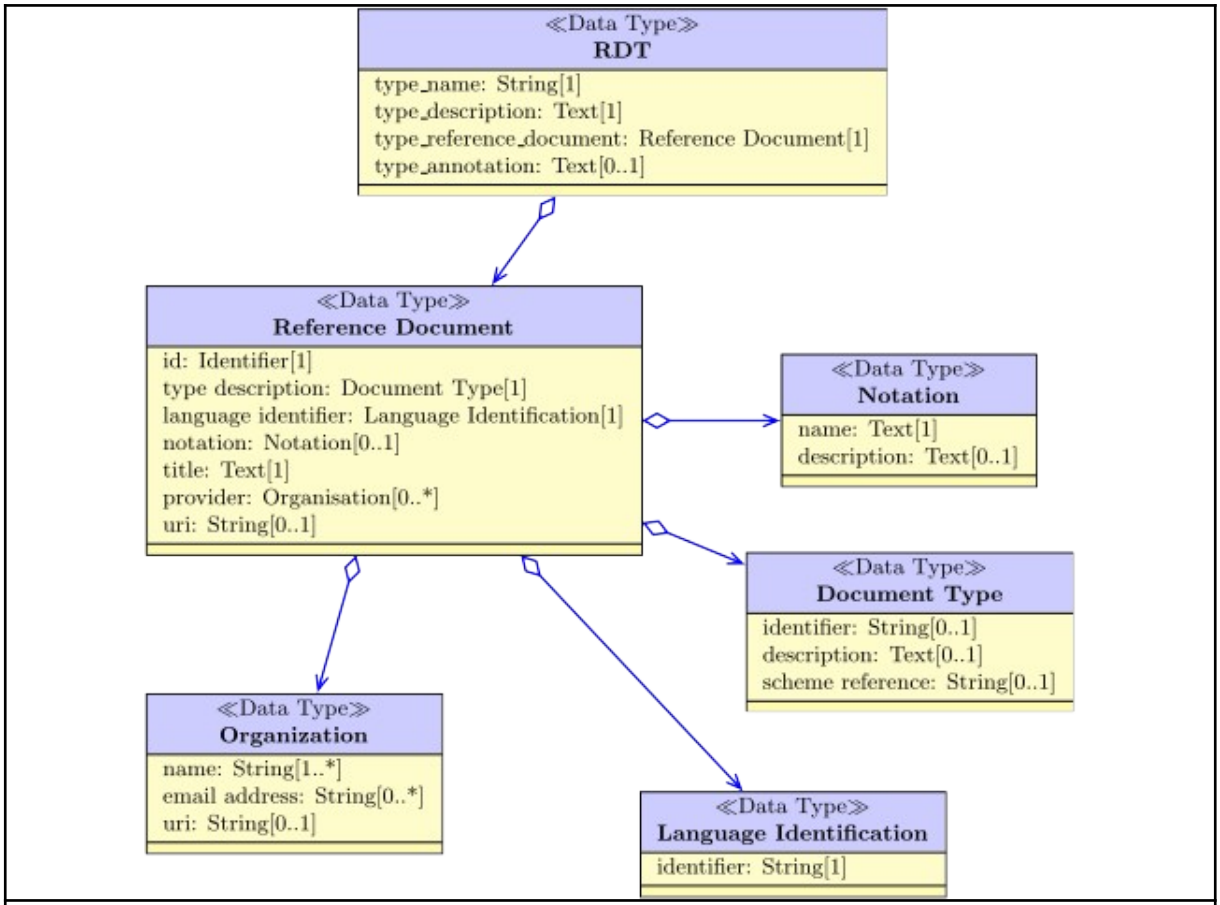


Figure 2: Reference document Data Type implemented from MDR. The MDR Sign data type, which permits strings as well as images and other formats, has been specialised as a simple string, and the Language Identification has been specialised by omitting optional

attributes (e.g. omitting the postal address from the organization class).

Example Data Types

Once a data type has been defined, it should be possible to express the data type and associated constraints in multiple application schema languages.

Example 1: the Variable Name

The MIP variable names, such as “tas” (Near Surface Air Temperature) are familiar to all involved in CMIP. The variable names are intended for use in code, and are also intended to have a mnemonic character. Hence, atmospheric variables ending in “s” are often[5] surface or near-surface variables. In order to make the terms suitable for use in code, the constraint that the first character is alphabetical and subsequent characters are alpha-numerical is imposed.

In order to express this description in a Data Type, we need to distinguish between rules and guidance. The character constraint is clearly a rule to be enforced, and can be expressed as an XSD constraint or via a python regular expression matching requirement (see Box 2). The mnemonic element of the variable name requirement is, however, clearly subjective and needs to be dealt with through a harmonisation process. Within this harmonisation process conflicts arise between the objectives of firstly having a consistent set of names within each MIP which facilitate communication in the community supported by that MIP and secondly wanting consistency between MIPs. An example of this can be seen in the contrast between “prsn” (Snowfall Flux) and “prsnIs” (Icesheet Snowfall Flux) versus “sndmasssnf” (Snow Mass Change Through Snow Fall). The last of these follows a pattern used in SIMIP to establish consistent naming for all the sea-ice variables requested in CMIP6, while the first two reflect the process of harmonisation across the whole of CMIP.

Box 2: XSD Constraint for Variable Name

```
<xs:restriction base="xs:string">
<xs:pattern value="([a-zA-Z])([a-zA-Z0-9])+"/>
</xs:restriction>
```

Python Class for Value of a Data Variable

```
class SimpleCodeWord(object):
    pat = '[A-z][A-z0-9]*$'
    re1 = re.compile( '[A-z][A-z0-9]*$' )
    def __init__(self, x, strict=True):
        """SimpleCodeWord
           ==============
           For use in code or documentation.
```

5 Some exceptions are, for example, “co2mass” (Total Atmospheric Mass of CO2) and “rlutcs” (TOA Outgoing Clear-Sky Longwave Radiation), in which the final “s” is part of a longer component such as “cs” for “clear sky”.

```
            Must only contain alpha-numeric characters, and start with an
            alphabetical character.


            Example:
              variable_name = SimpleCodeWord(‘tas’)
            """
        if strict:
            assert self.re1.match( x ), 'Value "%s" does not match required
pattern: %s' % (x,self.pat)
        self.value = x


    def __repr__(self):
        return self.value
```

With this approach, the mnemonic nature of the name will be expressed as guidance, and the associated review process captured in the provenance of the record containing the value. That is, the data type is associated with the intrinsic nature of the object, the guidance is associated with the appropriateness of the object in the context of the variable description.

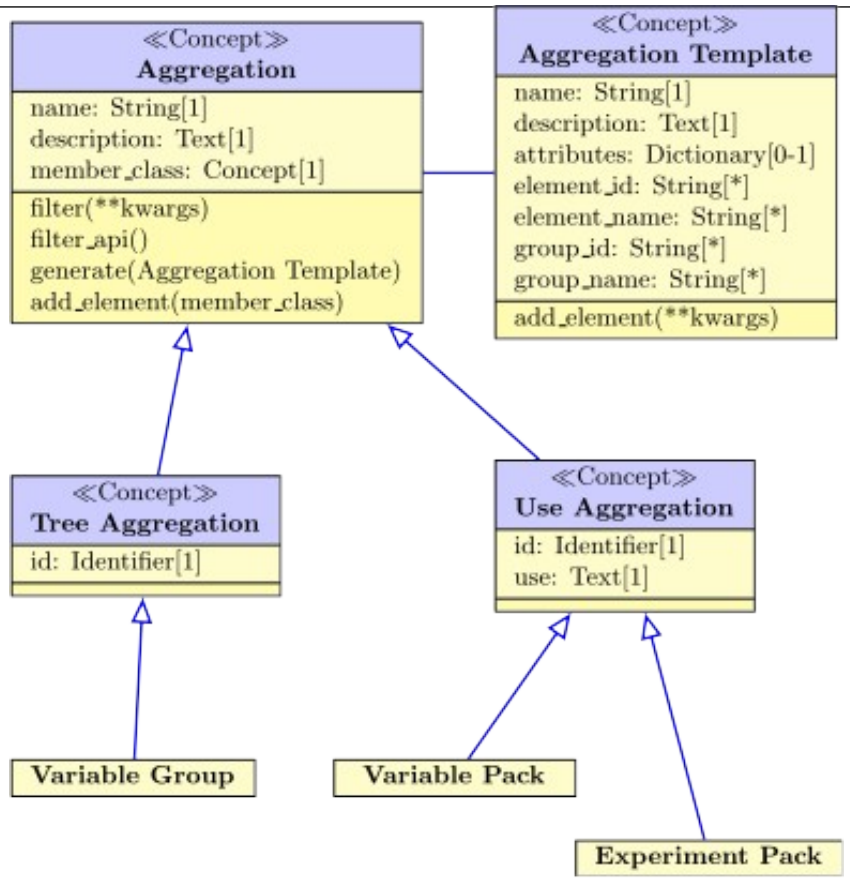
2.1.2 Aggregations

Aggregations of concepts are needed both for the management of concepts and for their use. To manage concepts we provide a many-to-one aggregation, while for use of concepts there is a many-to-many aggregation which allows a concept to be re-used in many aggregations.

These aggregations should have a common “filter” method to allow structured inspection of the aggregation, and a “filter_api” method to provide information on facets etc.

The Aggregations are not MDR concepts as such, but make use of MDR conforming data types. The departure from MDR here is in the use of associations and inheritance. This is not necessarily inconsistent with MDR, but it is more natural to follow a more open modelling approach.

Figure 3: Class structure for aggregations.



The Variable Pack will directly aggregate CMOR Variables (unlike the Request Variable Group in DREQ1.0) but provide an optional priority attribute to override the default priority if required.[6] This means that variables with multiple default priorities may be included in a single pack, but if variable priorities are being modified, a new pack for each new priority will be needed. This change removes a confusing layer in the object hierarchy and will make the motivation of any change in priority clearer (the large number of Request Variable records in DREQ1.0 makes it hard to extract an overview of information).

The Experiment Pack and the Objective Pack resemble the Variable Pack, but apply to aggregations of experiments and objectives. For instance, if a project wishes to request a selection of variables from selections of experiments, they need to define one Variable Pack and one Experiment Pack. In some cases they may be able to re-use existing Packs, especially if they want to follow specifications set by someone else (e.g. a network of projects) rather than set their own directly. Multiple Variable Packs and Experiment Packs can be attached to a single Analysis Objective.

6 The priority of a variable is an indication of the importance of that output for that variable from a simulation and for a specific objective. A single variable may have different levels of importance for different simulations and different objectives.

The Groups are used to organise concepts in trees which reflect a form of ownership. For instance, a Variable Group might contain variables associated with stable isotopes of oxygen and hydrogen which are used in paleoclimatology to relate proxy-climate records to climate processes.

The Request Group will contain all the Request records specified by a given project.

The explicit inclusion of a template to facilitate bulk import of content is a novelty here. Templates were used in Data Request 1.0, but they were not part of the schema. This led to some confusion about the relationship between the two. The template allows for some redundancy in the form of multiple ways of specifying the intended content. This redundancy provides flexibility, allowing the scientists who are providing content to choose an approach which fits their own workflows (bearing in mind that content generally comes from scientific teams or communities, rather than from individuals). This redundancy is removed when the content is imported into the aggregation records.

For instance, we may decide to define a Core Dynamics group of variables consisting of 5 variables: 3-dimensional fields of wind speed, temperature and humidity, and surface pressure. An analysis planned by an intercomparison project may require these variables plus one more: cloud cover. To specify this they will be able to use a template to record that they want “the Core Dynamics group plus the Cloud Cover variable”. The Variable pack created will then have 6 variables listed. This separation between the template and the variable pack and the template will be particularly important when a project is interested in re-using a long list of variables, e.g. ocean tracers, which is still being revised by another group. The separation of the template from the variable pack allows the variable pack to have a simple structure which will keep the interface simple for downstream applications. Having the specification of the template within the schema will ensure that the mapping from template to variable pack works smoothly.

2.1.3 Views

The drq:views package will describe the methods used to export content associated with data request records, including content aggregated from linked records. This might include, for instance, all the variables requested for an experiment, filtered according to priorities and MIPs, or all the experiments for which a variable is requested.

2.2 Physical Parameters

The physical parameters package will have a relatively simple semantic structure, but a critical role in specifying a reusable list of parameters to be used for model intercomparison across multiple MIPs and CMIP phases.

The definition of parameters relies heavily on the standard name from the CF Convention, but there are other components which are introduced below.

2.2.1 Standard Names

Much of the work is already done by the CF Standard Names, but there are many cases where the CF Standard Name alone does not convey enough information to identify what is considered by the CMIP community as a unique variable. In the CMIP6 request 929 standard names are used, and the list of physical parameters includes 1272 terms. For instance, the standard name “snowfall_flux” is

used in CMIP6 for variables dealing with snowfall on land, sea ice and land ice. The difference between these variables is made clear in the file metadata through inclusion of, for instance “where sea_ice” in the specification of the method associated with “area” in the “cell methods” string.

2.2.2 Labels (“short name”)

A crucial factor constraining the approach taken to specifying physical parameters in CMIP is the need for a short name for each parameter, which can be used in software and documentation, and which has a mnemonic value, such as “tas” for “Near-surface Air Temperature”. The short name is used in many circumstances for which an opaque identifier or the full name cannot provide a reasonable alternative.

The choice of the short name is constrained by a non-trivial namespace specification which is discussed in more detail in 7.3.1 below.

2.2.3 Title (“long name”)

The physical parameter package will also define the titles associated with variables. In the course of CMIP6 a style guide, with a style checking software package, was introduced to bring consistent usage to the full range of variables (Juckes 2018). For example:

- Carbon Mass in Vegetation on Shrub Tiles,
- Net Primary Production on Land as Carbon Mass Flux [kgC m-2 s-1],
- Upward Component of Land-Ice Surface Velocity.

The inclusion of the “[kgC m-2 s-1]” string in the mass flux title is a compromise. The land surface community commonly use “kgC”, meaning “kilograms of carbon”, as a unit of measure. This gives them a consistent way of avoiding confusion between fluxes expressed as “kilograms of carbon dioxide” and those expressed as “kilograms of carbon”. Considerable effort has been expended to arrive at this convention, but it unfortunately clashes with the approach to units in the CF Convention, discussed below.

2.2.4 Units

The CMIP variables are given specific required units, which is a more restrictive constraint than the canonical name specified by a CF Standard Name. For instance, “air_temperature” has canonical units “K” and this means that the standard name can be used with any units which are equivalent to Kelvin, such as degrees Celsius or micro-Kelvins. In order to make things easier for analysts, the CMIP data requests specify a specific unit string which must be used precisely in the form given for each variable.

As noted above, there is a clash between usage in the land surface community and the CF convention on the expression of “kilograms of carbon”. It is not just the land surface community that uses “kgC” as a convenient shorthand for “kilograms of carbon”. CF reserves the “units” string for units which conform to the SI concept of units and can be handled using the Unidata UDUNITS package. This aspect of the CF Convention rules out use of “kgC”. The CF Convention approach is

mirrored in the ISO 80000 standard, which states “Any attachment to a unit symbol as a means of giving information about the special nature of the quantity or context of measurement under consideration is not permitted.”

The ISO standard is in fact stricter than CF in this respect because it rules out use of formulations such as “kg kg-1” which is used in CMIP to distinguish mass fractions from other dimensionless quantities. This CF usage is consistent with the SI recommendations (SI2019). The CF Convention approach is constrained by the requirement for a units string that can be reliably and robustly interpreted by software and used to convert between units. For implementation, CF relies on the Unidata UDUNITS package.

UCUM2017 recognises the same problem in a different community: “because people want annotations and deeply believe that they need annotations [within the units string]”. Their solution is to allow annotations to be embedded in comment strings using braces, e.g. “kg{C}”.

There is some potential overlap with an attribute supporting alternative representations of units in the CMIP5 request. In CMIP5, the “print_units” attribute provided the option of a 2nd units string containing a unicode UTF-8 string (e.g. “m^{-2}”) rather than the ascii (e.g. “m-2”) required for use in the CF units attribute. NetDCF 4 supports use of UTF-8 in string attributes, so such strings could be included in the file metadata, perhaps as “print_units”, and might be allowed to follow the UCUM convention of supporting annotations. Such an attribute would be outside the CF Convention.

A preferable solution would be some form of convergence between IPCC community and the ISO standards, but that is clearly beyond the scope of the data request (it could potentially be addressed through related work at the IPCC Data Distribution Centre).

2.2.5 Constraints

The full specification of physical parameters in NetCDF files often requires additional CF Convention metadata, such as elements of the cell methods string or coordinate variables. The way in which this is implemented within the CMIP Data Request runs into further complexity because the coordinate variable and cell methods constructs are also used for quantities which vary independently of the physical parameter. This means that we need to express some cell method and coordinate information in the specification of the physical variable and some in the specification of the sampling structure.

In CMIP5 the constraints implied were not expressed within the semantics of the schema and needed to be handled independently through subjective checks. This approach does not scale well and undermines the potential for using the physical parameter list as a self-standing resource.

In order to resolve, or at least mitigate this problem, the 2.0 schema will introduce constraints which do not seek to specify the file metadata but do indicate what it should achieve: e.g. “Near Surface Field” or “Defined on a Grid Masked by Land Area”. The same key phrases will be included in the metadata of compatible structure records to enable automated consistency checks.

2.2.6 Serialisation

In ISO 11179 concepts are defined as members of a concept scheme, where the scheme should be a recognised standard (e.g. SKOS). For CMIP Variables, SKOS is a good choice. SKOS is already

used to hold and share CF standard names. Projecting the definition of properties into SKOS gives a high degree of re-usability.

The semantics of SKOS allow for specification of a concept through a range of properties. Values may be either SKOS concepts or literals, optionally qualified by a data type. We can use this structure to express the intention of our metadata. For instance, each variable has a title string, such as “Near Surface Air Temperature” which is designed to be compatible with usage as a figure caption.

Box 3: Illustrative view of a SKOS serialisation of the Near Surface Air Temperature physical parameter, showing use of data types.

```
var:tas rdfs:type skos:concept ;
   skos:prefLabel “Near Surface Air Temperature” ;
   skos:altLabel “tas” ;
   skos:definition “near-surface (usually, 2 meter) air temperature”;
   skos:notation “air_temperature”^^drq:type.standard_name ;
   skos:notation “K”^^drq:type.units ;
   skos:broader: drq:tag.Atmosphere, drq:tag.NearSurface .
```

```
drq:type.standard_name rdfs:type skos:concept ;
   skos:definiton “CF Standard Name: the value must be a valid CF Standard Name.” ;
   ……

drq:type.units rdfs:type skos:concept ;
   skos:definition “A CF compliant units string”;
   …

drq:tag.Atmosphere;
  skos:definition: “A property of the atmosphere, or a component of the atmosphere”
drq:tag.NearSurface;
  skos:definition “A property of the atmosphere (or ocean) in or on a layer which is close to the surface (i.e. the lower boundary of the atmosphere),
```

```
usually specified by a height coordinate” .
```

The use of skos:notation with a data type avoids the need to explicitly record terms which are imported from external vocabularies. The data type specification will contain the information needed to link to the full specifications in the external vocabularies (including version information). Some of this information may be imported for the “view” functions discussed below, but we avoid duplication by not explicitly representing the information in the data model.

The skos semantic relation broader is used to link concepts to key-word concepts which are defined within the data request.

2.3 File Metadata

The File Metadata package combines two main components: “Structures” and “CMOR Variables”.

The Structures specify combinations of dimensions, coordinates and attributes which can be re-used with multiple physical parameters. For instance, the “Temporal mean, Global field (single level) [XY-na] [amnla-tmn]” structure is used by 202 variables in the CMIP6 request. The code words at the end of the title indicate that the structure specifies horizontal coordinates and no vertical coordinate[7] and use of the cell methods string “area: mean where land time: mean”[8] indicating an area-mean masked by land.

The CMOR Variable records bring together physical parameters, structures, a frequency attribute (which takes a value from the CMIP6 CV for frequencies) and a range of other attributes. Some of these, inherited from CMIP5, are now redundant and can be removed at the next update. For instance the deflate, deflate_level, and shuffle options, which are compression options. If needed these should be specified outside the data request, as a CMIP infrastructure policy decision, and perhaps imported into the data request to be included in some of the output documents). They do not need to be specified independently for each CMOR Variable.

2.3.1 Harmonization of Filenames

There will be more significant changes following from the proposal on “Harmonizing Metadata and Filenames Across CMIP Eras” which is being developed by the WIP (WIP2020). This proposal affects the namespace rules for variables.

Within CMIP6 and earlier CMIP phases, each variable has a name which is unique within a “MIP Table”. Different MIP tables typically have different frequencies and different “super realms”[9], though there are exceptions to this general rule.

7 In CMIP6, as in CMIP5, variables without a vertical coordinate are assumed to be evaluated at the lower boundary of the atmosphere. It has been suggested that this should be changed in the future, so that the vertical coordinate is explicitly stated for all variables.

8 In most cases, the cell methods strings in CMIP6 included comments giving the name of the corresponding area fraction variable which is often needed for analysis of masked variables. This is done through a comment as there is currently no CF Convention option for expressing the connection.

9 The “Ocean” tables “Omon”, “Oday” etc, for instance, cover both “Ocean” and “Ocean Biogeochemistry” realms.

During CMIP6, the nature of the “frequency” label was changed to specify both the time interval between data points and the category of time processing, with four categories: point, mean (which includes maxima and minima over a sampling period as well as time mean), annual climatologies and monthly mean diurnal cycles.

In CMIP5, different tables could have variables with the same name frequency and realm, but with different masking specifications. During CMIP6 it was decided to remove this anomaly, and distinguish variables with different masking options by distinct variable names.

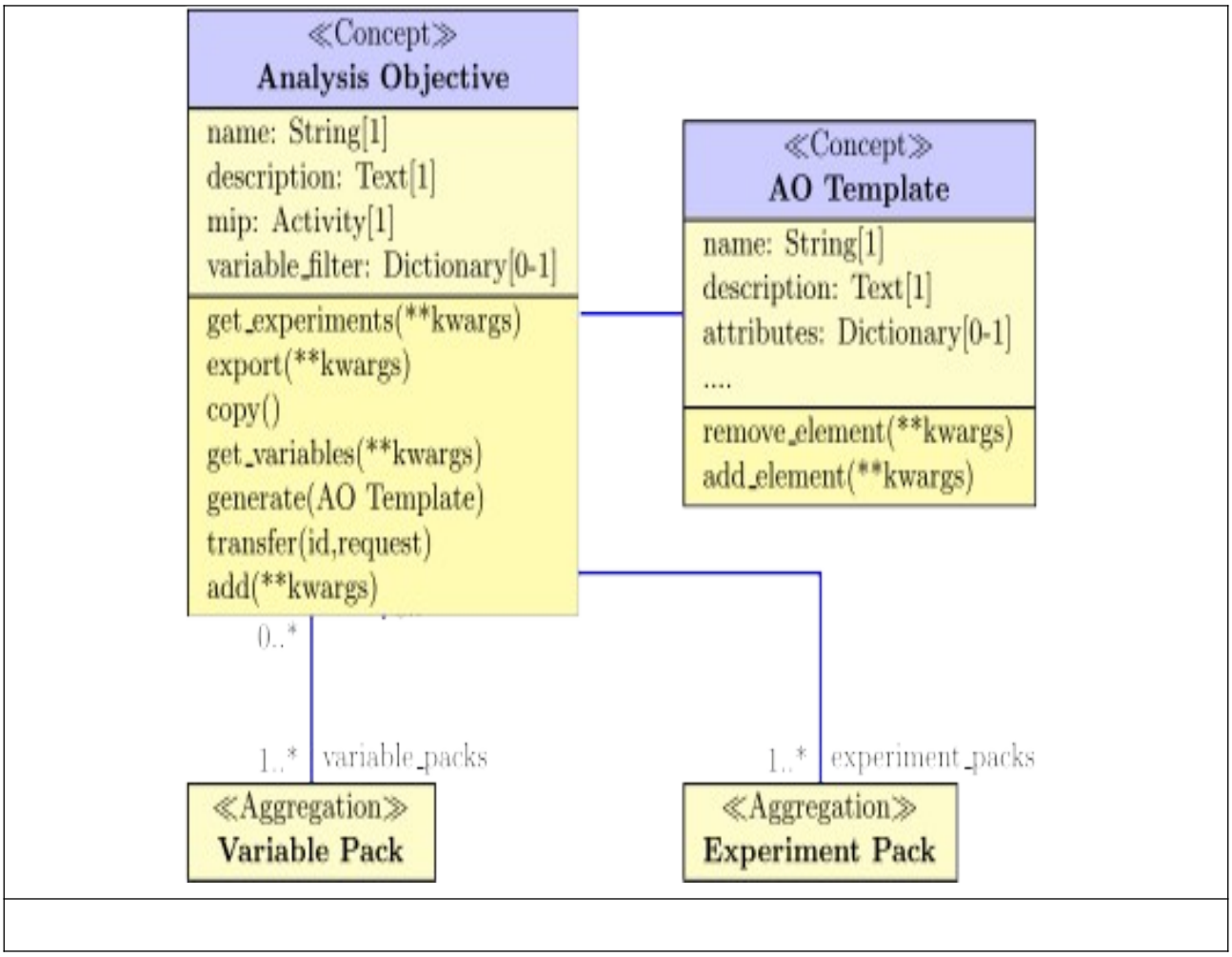
Under the WIP2020 harmonization proposal, the MIP tables will cease to be part of the file names and the namespace rule for variables will change from requiring uniqueness within each MIP table to uniqueness within each frequency/realm combination. For example, “pr.Amon” and “prCrop.Emon” would change to “pr.atmos.mon” and “prCrop.atmos.mon”, “co3abio.Omon” will change to “co2abio.ocnBgchem.mon”. Initial analysis suggests that this change will require a small number of variable changes. Removing the MIP table from the file names will bring greater clarity and flexibility. MIP tables may still be used in the data request views, for displaying lists, and templates, for ingesting them.

2.4 Analysis Objectives

The Analysis Objective package will set out the data requirements associated with specific aims of the MIP. For instance, the HighResMIP has an objective described as “Improved understanding of biases in the simulated diurnal cycle, and potential consequences for surface fluxes, energy cycle, and extremes” and another for “Improvement in the simulation of ocean and sea-ice dynamics and the exchange with the overlying atmosphere in hot spots such as the Gulf Stream”. These objectives have different data requirements.

In CMIP6, it was possible to specify multiple “requestLink” records for a single analysis objective. This allows, for instance, the freedom to link a request for 25 variables from experiment1 and 100 variables from experiment2 to a single objective. The proposed structure for Data Request 2.0 would remove this complexity. Instead, this example would result in two different data request analysis objective statements, perhaps designated as part 1 and part 2 of an overall science objective. This change will result in a simpler structure for the request schema.

Figure 4: Principal classes in the Analysis Objective package



3 Conclusions

The experience of CMIP6 has generated a clear set of requirements guiding the development of an enhanced structure for the data request in future CMIP exercises.

The design proposed here makes provision for re-use of variable names and metadata combinations across multiple MIPs, formalising a process of re-use which already takes place informally. In the informal approach people may have a variety of different interpretations of what should be preserved when a concept is re-used, and this diminishes the value of the concepts being shared. The formalisation will make it possible to re-use concepts in a more consistent manner.

The aim is to have a schema which allows flexibility around the organisation of content, so that the schema and associated software can be re-used across multiple programmes.

Data Request 2.0 will have a simpler and more transparent structure, making it easier for MIPs to specify simple requests with simple technical input.

4 References

Balaji, V., Taylor, K. E., Juckes, M., Lawrence, B. N., Durack, P.J., Lautenschlager, M., Blanton, C., Cinquini, L., Denvil, S., Elkington, M., Guglielmo, F., Guilyardi, E., Hassell, D., Kharin,

S.,Kindermann, S., Nikonov, S., Radhakrishnan, A., Stockhause, M., Weigel, T., and Williams, D.: Requirements for a global data infrastructure in support of CMIP6, Geosci. Model Dev., 11,3659–3680, https://doi.org/10.5194/gmd-11-3659-2018, 2018

Eyring, V., Bony, S., Meehl, G. A., Senior, C. A., Stevens, B.,Stouffer, R. J., and Taylor, K. E.: Overview of the CoupledModel Intercomparison Project Phase 6 (CMIP6) experimental design and organization, Geosci. Model Dev., 9, 1937–1958,https://doi.org/10.5194/gmd-9-1937-2016, 2016.

M. Juckes, K. E. Taylor, P. Durack, B. Lawrence, M. Mizielinski, A. Pamment, J.-Y. Peterschmitt, M. Rixen, S. Sénési, The cmip6 data request (version 01.00.31), Geoscientific Model Development 2020 (2020) 201–2024. doi:10.5194/gmd-2019-219. URL https://www.geosci-model-dev.net/gmd-2019-219/

Juckes 2018: Style Guide for Variable Titles in CMIP6. https://zenodo.org/record/2480853

SI2019: https://www.bipm.org/utils/common/pdf/si-brochure/SI-Brochure-9.pdf

UCUM2017: The Unified Code for Units of Measure, https://ucum.org/ucum.html#section-Character-Set-and-Lexical-Rules.

ISO19508: Information technology — object management group meta object facility (MOF) core, Standard ISO/IEC 19508:2014, ISO/IEC, Geneva, CH (2014). URL https://www.iso.org/standard/61844.html

ISO80000-1: Quantities and units — part 1: General, Standard EN ISO 80000-1:2013(E), ISO, Geneva, CH (2011). URL https://www.iso.org/standard/30669.html;

ISO80000-3: Quantities and units — part 3: Space and time, Standard EN ISO 80000-3:2006, ISO, Geneva, CH (2006). URL https://www.iso.org/standard/31888.html

ISO80000-4: Quantities and units - part 4: Mechanics, Standard EN ISO 80000-4:2019, ISO, Geneva, CH (2019). URL https://www.iso.org/standard/64975.html

ISO80000-5: Quantities and units - part 5: Thermodynamics, Standard EN ISO 80000-5:2019, ISO, Geneva, CH (2019). URL https://www.iso.org/standard/64976.html

ISO80000-7: Quantities and units - part 7: Light and radiation, Standard EN ISO 80000-7:2019, ISO, Geneva, CH (2019). URL https://www.iso.org/standard/64977.html

ISO80000-9: Quantities and units - part 9: Physical chemistry and molecular physics, Standard EN ISO 80000-9:2019, ISO, Geneva, CH (2019). URL https://www.iso.org/standard/64979.html

ISO19115: Geographic information — metadata part 1: Fundamentals, Standard EN ISO 19115-1:2014+A1:2018(E), ISO, Geneva, CH (2014). URL https://www.iso.org/standard/53798.html

ISO19135: Geographic information procedures for item registration part 1: Fundamentals, Standard ISO 19135-1:2015, ISO, Geneva, CH (2015). URL https://www.iso.org/standard/54721.html

ISO38500: Information technology — governance of IT for the organization, Standard ISO/IEC 38500:2015, ISO/IEC, Geneva, CH (2015). URL https://www.iso.org/standard/62816.html

ISO19439: Enterprise integration framework for enterprise modelling, Standard EN ISO 19439:2006, ISO, Geneva, CH (2006). URL https://www.iso.org/standard/33833.html

ISO19440: Enterprise integration constructs for enterprise modelling, Standard EN ISO

19440:2007, ISO, Geneva, CH (2007). URL https://www.iso.org/standard/33834.html

ISO11404: Information technology general-purpose datatypes (gpd), Standard ISO/IEC 11404:2007, ISO/IEC, Geneva, CH (2007). https://www.iso.org/standard/39479.html

ISO11179-1: Information technology -- metadata registries (MDR): Part 1: Framework, Standard ISO/IEC 11179-1:2015(E), ISO/IEC, Geneva, CH (2015). https://www.iso.org/standard/61932.html

WIP2020: Taylor et al. Harmonizing Metadata and Filenames Across CMIP Eras. https://docs.google.com/document/d/1WLxaxQmGuAf757lqgX1bVp7T1U08VO8IuR2esD11jvw/edit

Additional Reading

RDA Working Group on Data Type Registries: https://www.rd-alliance.org/node/145/wiki

Deep Carbon Observatory: https://deepcarbon.net/dco_datasets

RDF datatyping: http://infolab.stanford.edu/~melnik/rdf/datatyping/#s-a

Expressing Dublin Core™ metadata using the Resource Description Framework (RDF) : https://www.dublincore.org/specifications/dublin-core/dc-rdf/2007-06-04/

Semantic Specification of Data Types for a World of Open Data: ISPRS Int. J. Geo-Inf.2016,5, 38; doi:10.3390/ijgi5030038
https://ui.adsabs.harvard.edu/link_gateway/2016IJGI....5...38M/doi:10.3390/ijgi5030038