The Structure of the WEC/ISDAT Data

WECdata Working Group

Contents

1 Introduction

It is trite, but nevertheless true, to say that the ease of processing and analysing data is directly related to the suitability of the way it is stored and accessed. The processing and analysis are determined by physical considerations, and so too should be the optimisation of the storage and access architecture.

Space physics data is characterised by being organised in long strings of objects in chronological order. For each value of the time, the corresponding data object consists of one or more values which are all determined simultaneously, and being either

- A single object, consisting of either:
  1. a value which is "stand-alone", like the density or temperature, or
  2. values which are "naturally associated", like the three components of a "true vector" or the nine components of a "true tensor", upon which standard operations can be performed, and in particular transformation to another coordinate system.
- A sets of several such objects, all of which have identical structure, all of which have the same spatial characteristics, but which nevertheless have some characteristic which distinguishes them from each other: examples are the different frequency channels of a power spectrum or a cross-spectral matrix, or the different energies of a particle spectrum.

All data needs its meta-data to be associated with it in a simple and logical way. The meta-data specifies units, the coordinate system, and other information which is essential to be able to process or scientifically interprete the data. It also includes attributes such as the processing software version and calibration set version used to derive the data. Generally the meta-data changes more slowly than the data itself, although this is not necessarily the case: for example, if the coordinate system is spinning.

Information about the nature of the data objects must be immediately available to applications (clients, filters, etc.), and is part of the meta-data.

Sometimes the data objects of type 2 above will not be a "true" vector or tensor, but may be "reduced", with only two components. They may even an inhomogeneous vector in which the different elements are of totally different nature, such as density, temperature and (scalar) pressure. (Inhomogeneous vectors are limited to being uni-directional.) As a general rule, however, it is far better to create a separate logical instrument for each set of data which is physically different, so as to benefit from the full power of ISDAT (see section **??** below).

The purpose of the data structuration which is proposed here is to separate:

- the time dimension, in which averaging and other more complicated processes like re-sampling or Fourier transformation can be performed;
- the space dimension, in which coordinate transformations may be applied;
- other (non-space, non-time) possible dimensionality of the data set.

2 The ISDAT Data Structure

The sequential data set consists of a series of pairs of structures, each pair consisting of

- one value of the time, and
- either
  1. a single structure, or
  2. a set of identical (and simultaneous) structures.

This association between the time-series, the data itself, and its associated meta-data, and can be obtained using the following architecture, in which

- the top level defines the general data object, and
- the second level is a typed data object. The type determines the nature of the operations which can be performed upon it: coordinate transformations, graphical displays, the method of taking differences (*i.e.*, the number of elements whose difference must be calculated) of two data sets are sampled simultaneously, but not operations in the time domain, such as averaging or joining.

2.1 Definition of the Data Structure

1. A list containing the meta-data. This can include pointers, although they need not all be filled (clients MUST test all meta data values and pointers which concern them).

2. The time array (time-series) pointer for time-tagged data: this will be null in the case of segmented data[1]. *(CCH question: Timing information [start and stop times, and time increment] is always available. Where, and where is this address specified ?)*

3. The data object type[2], which will be one of the following known types:

   - a tensor, which can be
     - a scalar (tensor of rank 0);
     - a vector (tensor of rank 1);
     - a tensor of rank 2, (*e.g.* pressure);
     - a tensor of rank 3, (*e.g.* heat flux).

     Arrays are stored in memory in the order of the C programming language.
   - a scalar function of a vector (*e.g.* a distribution function)
   - another standard known type:
     - a 2D vector
     - the diagonal elements of a tensor
     - a particle counts object (TBD)
     - an inhomogeneous vector object (this is a "catch-all" option which, however, is not recommended);
     - etc. (new standard types can be defined as and when necessary).

4. A pointer to the first element of the typed data object, which will contain either:

   A) Regular time-tagged data, with:
      - the number of identical objects per time tag, e.g., the number of channels in an energy spectrum or frequency channels of a power spectrum or a cross-spectral matrix.
      - the number of records being returned, *i.e.*, the quantity of data. Since both the time and the data structures are in arrays, this is the highest dimensionality of those arrays.
      - a pointer to the time array.
      - a pointer to the actual data array.

   B) Irregular time-tagged data. For example, a 2D vector in a measurement plane defined by its normal would have:
      - the number of identical objects per time tag (as for A, regular time-tagged data).
      - the number records being returned (as for A).
      - a pointer to the time array.
      - a pointer to the data array.
      - pointer to 3D vector which is normal to measurement plane. This is a 3D object with the following sub-pointers
        - number of records
        - pointer to time array
        - pointer to vector array
      - a pointer to the origin of phase in the measurement plane. This is also a time series
        - number of records
        - pointer to time array
        - pointer to vector array

[1] Note that the structure of the data segment is unaffected by this choice of timing.
[2] This information is needed at a high level in the hierarchy because it is essential to determine which filters and/or clients can be used.

2.2 What cannot be handled

The proposed data structure does not allow a single logical instrument to handle mixed objects lying on the same time-line. The first bullet of sections 4A and 4B of the preceding section indicate the number of *identical* sub-structures contained in the data structure. For example, it is not possible to mix scalars and tensors at this level. This is different from CDF; but the latter is primarily a storage tool, not an analysis tool.

This is not an ISDAT design deficiency: the mixing of different data objects is totally compatible with the concept of the logical instrument. Data objects with different structure are necessarily from physical observables of different nature. Therefore they cannot share the same clients, and should be kept separate, in different logical instruments. Note that there is nothing to prevent different logical instruments from sharing the same time-line.

3 Meta Data and Status Data

Meta data is information which is intimately attached to the physical data (*e.g.*, units, coordinate system, possibly limit values). It must be joined when the physical data is joined.

Status data is information which is separated from the instrument data to which it refers, for use independently of the instrument physical data. In particular, it may be used by other experiments, to determine the conditions for their own observations. It has an inherent time granularity, which is somewhat arbitrary. Depending upon the precise definition of the status, it may be more or less degraded when joined. It is preferable not to join status files when they are changing rapidly (which is also when the are most interesting).