Measurement and Telescope Data Sets and Application Objects

for Single-Dish Work and Interferometry

(AIPS++ Implementation Note #152)

Authors/Participants

Dave Shone, Brian Glendenning, Bob Hjellming, Gareth Hunt,
Bob Payne, Russell Redman, Ger van Diepen, Allen Farris,
Bob Garwood & Ron Maddalena

1 Introduction and Background

A series of meetings held in Charlottesville and Green Bank in November 1992 considered the definition of the contents of a Measurement Set and associated Telescope Model Data for radio single dish and interferometer data. This note is based on the conclusion of those meetings, but also contains a small amount of new material. Apart from recording this work, it is hoped that this note will be useful in stimulating feedback from those who create the "data sources", and as a guide to further protyping and implementation.

We have concentrated refining and extending the data modelling aspects of the principal objects involved in calibration, with particular interest in incorporating single dish analysis. The objects of interest are the Telescope Model and Measurement Set (formerly known as YegSet), both of which are described in the Project Book. We regard these as logically appearing as pure tables, and our objective has been to determine what columns are required for single dish and interferometric radio astronomy. Note that these are logical columns or coordinate-keys ("CoKeys"), and this does not necessarily reflect the way in which the data are actually stored, although it is likely that the actual implementation will be based on the extended FITS tables described in the Project Book.

The data models described are subject to further revision based on feeback from the detailed analysis of the requirements of particular intruments. Further revision of this material will appear in the project book, and/or cast in stone as a memo.

In order to set the direction for implementation and prototyping, some applications objects suitable for prototyping are proposed.

1.1 Minimalist Revisionism

Previous attempts to define the distinction between measurement data and Telescope model data have encountered the problem that some coordinate-keys in a Measurement Set are derived from data which clearly belong in the Telescope Model. An example is the baseline vector (u,v,w) in interferometry; this is dependent on the positions of antennas, and these may be adjusted, requiring a recalibration of (u,v,w). We have decided to adopt a "minimalist" approach to coordinate keys in the Measurement data set; only coordinates which are required to uniquely identify a measurement will be stored in the Measurement Set. Other relations may be produced by effectively joining the Measurement and Telescope Model data tables.

The distinction we propose leads to a clearer division of data between Telescope Model and Measurement Set, but many operations will be complicated by having to select data from two data tables. However, this is not likely to be a serious problem, since operations which often require such selections will usually be hidden in an application object. Interferometry again provides an example, where (u,v,w) will be implemented as an application object which performs the appropriate data selection, and calculates "on-the-fly", if necessary (in fact this might be a service of an interferometer instrument component of the Telescope Model). Alternatively, (u,v,w) columns might be be created temporarily (this is a good example of the the utility of being able to store columns separately).

In the interferometer case, applications will typically work with visibilities or other aggregates such as integrations or baselines; these will almost certainly have services to provide (u,v,w). In addition, generic table operations will always be possible on the Mesasurement and Telescope Model data, so a "joined" data set may be formed if necessary.

1.2 Implications for Tables

If we follow our original definition of a Measurement, in which it is atomic in the sense that it represents the smallest element of measurement which has a physical meaning, then the Measurement is an element of a spectrum rather than a complete spectrum. This appears somewhat alien (not to mention potentially inefficient, on the face of it), but is important in order to maintain a consistent mechanism for the selection and aggregation of data; a generalised spectrum might not alway consist of intensities regularly sampled in frequency, and thus it must be possible to treat frequency in the same way as any other coordinate.

In practice, a spectral Measurement Set will probably treat frequency as an implicit coordinate (a regular axis), and thus avoid the inefficiency associated with the general case, whilst retaining the same database interface.

In a similar fashion, some data which are traditionally considered as "header data" (e.g., the observer identity) are logically expanded in a Measurement Set, so that such data appear to be present in each Measurement. In practice, these will be implemented as virtual columns, thereby economising on storage requirements, whilst allowing the flexibility of storing these as real columns when necessary. Other traditional header data will appear in the Telescope Model, rather than being included with the measurement data in the Measurement Set.

2.0 Data Model Definitions

2.1 Measurement and CoKeys for Interferometer Data

Complex Visibility Measurement - the amplitude/phase or Real/Imaginary values for a single point in the observation.

Data Quality Measures - Essentially weights and flags, although application-specific measures are also possible.

Time

Polarisation - An enumerated type describing the polarisation of a measurement, e.g., one of I, Q, U, V or RR, LL, RL, LR.

Frequency channel - Used together with IF to index information frequency information in the Telescope data

IF - Used as index to receptor set in Telescope data, and together with Frequency channel, to index frequency information in the Telescope data.

(Antenna1,Antenna2) - Uniquely determines antennae and associated correlator channel in the Telescope data.

UserTags - Probably similar in function to Data Quality Measures; maybe we should lump these together, although the formal error in a measurement should probably be distinguished.

2.2 Measurement and CoKeys for Single Dish Data

Intensity Measurement

Data Quality Measures - Essentially weights and flags, although application-specific measures are also possible.

Time

Polarisation - An enumerated type describing the polarisation of a measurement, e.g., one of I, Q, U, V or RR, LL, RL, LR.

Frequency channel - Used together with IF to index information frequency information in the Telescope data

IF - Used as index to receptor set in Telescope data, and together with Frequency channel, to index frequency information in the Telescope data.

IntegrationPhase - An enumerated (and usually implicit) coordinate which describes the integration/calibration phase with values "on", "off", "dark" or " ".

UserTags - Probably similar in function to Data Quality Measures; maybe we should lump these together, although the formal error in a measurement should probably be distinguished.

Initially, it was unclear whether the Integration Phase should be implemented as a single coordinate with enumerated values (as described above) or as a different measurement values within a single Measurement. If we take the view that any one of these values is meaningless without the others, then the latter approach is required to make an appropriate atomic Measurement. However, in some circumstances it may be necessary to change groupings of Integration Phase (i.e. the astronomer may wish to use a different "OFF" phase for one or more "ON" phases), which supports the former approach. This case is analogous to calibration in interferometry, where different Integration Phases correspond to different sources, target and calibrator. For these reasons, we favour the approach where calibrator phase is a coordinate, particulary since it follows our general philosophy of maintaining a common means of selecting subsets of data. Additional calibrator phases could be introduced simply as new Integration Phase values, eliminating the need to change the Measurement definition, as would be necessary otherwise.

2.3 Radio Telescope Model Data

We envisage that radio telescopes will usually be sufficiently similar that a minimal core can be specified. Of course, this does not define the internal workings of a given Radio Telescope Model, but rather a minimal set of services which must be provided. A specific Telescope Model may also provide additional information, perhaps for a given site or application.

The Radio Telescope Model is likely to breakdown into a number of components as described in the project book, although this is not always necessary or desirable. The important point is that the interface presented to the outside world has the common core.

2.3.1 Receptor

Gain - complex for interferometer

Tsys

Tcal

Residual Delay

Residual Rate

2.3.2 Instrument

In the case of an interferometer, this will contain baseline/correlator information (e.g., gains which are factorised to individual receptors.

Correlator gain (Ant1, Ant2, IF, Time)

Delay

Rate

Frequency Channel Width

Frequency Channel Separation

Bandwidth

2.3.3 Telescope Element

Location - relative to platform

Projected Plane Coordinate

Sky Position - antenna pointing position

Mount Type

Axis Offsets

Gain(SkyPos) - generalised form of Gain versus Elevation

Primary Beam Gain - generally two-dimensional

SubArray identifier

2.3.4 Platform

Reference System - coordinates reference system descriptor/parameters

Time System - time system descriptor/paramerers

Pole Location

Earth rotation rate

Orbital Parameters - for oxbiting platforms

2.3.5 Environment

Zenith opacity

Weather - this is likely to be observatory specific.

2.4 Implementation Issues

2.4.1 Regular and irregular data arrays as implicit and explicit coordinates

We have already suggested that in the usual case of a regular spectrum, the frequency coordinate will be implemented as an implicit coordinate. In many cases, other coordinates may also be implemented in this way. Typically, calibration switching takes place in a cyclic fashion, and thus the integration phase may also be implemented as an implicit coordinate.

In the case of the UniPOPS single dish data format, the implicit data coordinates and the dimensions of the appropriate data array will be determined by the observing mode parameters; see the definition in the UniPOPS reference manual, in particular the observing parameters in class 3 and the phase block in class 11. (This is specific to Green Bank/Tucson; we should eventually generalise by reconciling this with Rick Fisher's note of 11 Nov 1992 and specifications from other instruments, e.g., the JCMT).

2.4.2 Time, Frequency, Location and Position classes and reference systems

Time, Frequency, Location and Position quantities should all be defined as classes, in order that the appropriate reference system and it's parameters may be encapsulated, and transformations from one system to another may be performed transparently. In the current data model, the Telescope Model Platform component provides a description of the reference systems used. This may be deemed unnecessary if such things are incorporated in the aforementioned quantity classes. However, it may prove desirable to allow these quantities to be calibrated or recalibrated, e.g., UT1-UTC might be adjusted, and therefore it would be useful for the parameters of a particular reference system for a particular observation to be accessible in one place, and the Telescope Model (probably the Platform component) seems the most sensible place. Of course, these parameters would be accessible through the coordinate classes, but in this case, these would refer back to an associated Telescope Model.

3.0 Some Application Objects for Prototyping

Most applications will not operate directly on a Measurement Set; rather, they will use application-oriented objects which manage the selection and aggregation of data from a Measurement Set. These objects embody data structures which are generally more "astrophysically meaningful" than implementation structures such as tables and arrays. In additions to the data structures, application objects have methods which operate on these data structure, with semantics appropriate to these types.

Prototyping is necessary to test several aspects of this system design:

- The basic data system architecture and interfaces - essentially a kind of object-oriented database sitting on top of something which more closely resembles a relational database.

- The low-level data model - the contents of the Measurement and Telescope Data tables and the division of data among these.

- Use by applications - definition of an appropriate set of application objects.

- Efficiency.

Probably the best way to test all of these is by designing, implementing and using some application objects to use the prototype low-level interface. We envisage several such objects which are suitable for early prototyping, and we describe them here in the order they should be attempted.

3.1 Spectrum - The principal application object for Spectral-line work

This was selected as a simple class for initial prototyping, and at the time of writing, work is underway to code this.

3.1.1 Construction

Two constructor methods are proposed for the prototype:

- A method to construct a spectrum from an underlying table of data. This will select data according to criteria specified to the constructor, and perform gridding or regridding where necessary.

- A method to construct a spectrum from a vector of intensity values, together with channel description information. This should eventually be extended to write data to a table with an implicit frequency Coordinate/Key, in order to provide a higher-level means of filling a spectrum measurement set in the case of regularly sampled data, or for a related copy constructor.

3.1.2 Attributes

- A service which returns a reference to a vector of intensities.

- A service which returns a vector of frequencies to support the general case of irregulary sample data.

- A service which returns a vector of velocities

- A spectrum may have an optional Integration Phase value, applicable to the entire spectrum - a spectrum in which the different elements have different Integration Phase values does not appear to be meaningful. However, if this is not present or is set to "none", then the spectrum may have been formed by averaging or some other mathematical combination of different Integration Phases, e.g., a calibrated spectrum could be formed by calculating (on-off/off-dark) in the constructor or "on-the-fly" when an element is accessed.

In addition, a service should also be available which provides a table with containing any of the attributes - essentially the subset of the original Measurement Set from which the Spectrum is formed.

3.1.3 Methods

In addition to the usual mathematical methods associated with arrays:

* Addition - e.g., to support the case of addition of irregulary sampled measurements.

3.1.4 Implementation Notes

It is not immediately obvious whether a spectrum should be implemented by derivation from a vector class, or as an object which has vector attributes. The former provides for a slightly neater way of using spectra in some cases (e.g., a spectrum, rather than one of its attributes could always be used where a vector might be use). However, a spectrum has so many additional properties (including a number of attributes which are vectors) that the second approach seems preferable. It is possible that we might ultimately find it useful to derive spectrum (and other application objects) from a purely mathematical series class.

3.2 The Visibility and other Interferometric Application Objects

Although we did not discuss these in any detail in the CV/GB meeting, some suggestions/guidelines are proposed here.

The minimalist revision of the low-level data model has placed additional responsibilities on some application objects. Our clearest example of this is in interferometry, where (u,v,w) no longer have an obvious place, since they are derived from the Telescope Model for each measurement. The Visibility will have to be able to provide a service to return a (u,v,w) attribute, and in the absence of any other object willing to take responsibility, this will have to be calculated by the visibility (this is still somewhat of a moot point; it might be sensibly be provided by the Telescope model). For the time being, we should assume the following:

* The visibility will always have a service which provides the (u,v,w) attribute;

* In the absence of a "(u,v,w) column" in the Telescope Model (which might be provided by some component thereof), the visibility will calculate (u,v,w) using the appropriate information from the Measurement and Telescope Data (this may or may not be "cached") according to an agreed "default prescription"

* Some Telescope Model builders may prefer to provide (u,v,w) calculated according to their own ideas and beliefs.

3.2.1 Construction

Construction from an existing database will be done by selecting an appropriate time, Antenna pair etc., This class (and the aggregates described later also offer a higher level filler interface than direct filling of the low-level table interface.

3.2.2 Other Visibility Attributes

In addition to (u,v,w), the visibility class will have service to provide:

* a complete polarised (I,Q,U,V,RR,LL,RL,LR) spectral visibility (we could provide calibrated and uncalibrated interfaces)

* antenna pair;

* IF.

3.3 Other Interferometric Application Objects

In order to "hide" the detail of data selection, we should probably provide some objects which are aggregates of Visibilities, in particular Integration (a selection of all visibilities in some narrow time range), Baseline data (all visibilities for a particular baseline) and Spoke (all visibilities in some narrow sector of the u,v plane). The implementation of these is an issue for further discussion and prototyping, but the following starting point is proposed:

* Each of these should have a service to return a Vector of Visibilities.

* Attributes of individual visibilities might also be usefully provided in Vector form, e.g., vectors of Amplitude, Phase, u, v, w.