Definitions for Single Dish Calibration

The Overall Structure of Yegs and TelescopeModels

1992 August 25

R. O. Redman and R. Payne

1 Basic Definitions

AssociatedInformation

— information contained in the header of a Measurement or IntensityModel which is not directly used to calibrate the data. The AssociatedInformation may be used in a selection criterion during later processing. Beyond compatability with the chosen file format, no constraints can be laid on the format or content of AssociatedInformation.

Atomic Yeg

— the smallest astronomically meaningful unit of data produced by an instrument. As examples, the atomic yeg for a CCD camera would be the pixel, for a spectrometer would be a single frequency channel, and for an interferometer would be a single visibility on a particular baseline. An atomic yeg may be represented as a list representing the measured value together with sufficient information to distiguish this atomic yeg from all others. If an instrument samples an n-dimensional coordinate space (including signal strength), its atomic yeg will usually be an identifier together with a simple n-tuple. The atomic yeg is primarily intended as a unit of data selection. There are philosophical and practicle problems involved in the definition of atomic yegs for some kinds of Measurements and IntensityModels (see the discussion in the next section).

Calibrated Yeg

— an IntensityModel containing measured data. Note that there are some predefined IntensityModels which do not describe measured (or even fake) data, and which are not usually considered to be yegs. Examples might be a table of known rest wavelengths for a set of spectral lines, or the temperature of a cold load. On the disk, a calibrated yeg will usually look just like the raw data except that the numbers will be properly scaled and descriptions of the non-measured and implicit coordinates may be sequestered in a "header". Note in particular that a calibrated yeg will often inherit the discreteness and dimensionality of the original raw measurement. A yeg is said to be partially calibrated if some of the radiation coordinates implicit in the raw measurements are not described in the IntensityModel. In this case the relevant unprocessed measurements may be included by applying another TelescopeModel to the Measurements, using the same IntensityModel for output. Notice that subsequent steps in data reduction, such as averaging scans and fitting Gaussian components, are actually operations on IntensityModels or on their MathComponent parts.

Calibration Group

— an aggregate of Measurements which naturally group together during calibration. Each kind of calibration group will have an associated TelescopeModel describing the "role" of the component Measurements; it would be desirable for each Measurement to carry a label in its header describing its intended role plus optional selection criteria to locate the other members of its group. For example, in an ON/OFF switching mode, each valid observation would be labelled ON or OFF, and the ON might specify that the nearest preceding OFF should be used for calibration.

Component

— an abstract class serving as an umbrella for all of the data and telescope related classes. Its only real requirement is that everything have a name which may be used to refer to it.

Identifier — a selection criterion for a particular yegset which does not depend upon the contents of the yegset. It must be possible to specify every valid atomic yeg individually using identifiers. Note that some of the contents of a yegset may be used quite legitimately to identify a subset of the yegset. For example, an instrument yeg containing an array must have an identifier which does not refer to the array; that identifier in conjunction with an index into the array could be used as an identifier for a particular yeg within the array. Note that there are many other selection criteria which can be used than just identifiers. Identifiers merely allow us to refer to the data in a non-prejudicial way.

IntensityModel

— a kind of RadiationComponent usually representing a piece of calibrated data. An IntensityModel will use MathComponents (as discussed in Note 142) to describe some significant aspect of the intensity

$$I(\nu, t, l, b, P).$$

Note especially that an IntensityModel is NOT required to describe all aspects of the intensity. Many spectroscopic observations will report only the apparent wavelength of a chosen set of spectral features, for example, ignoring all other aspects of the radiation. An IntensityModel is permitted to include a physical description of the source of the radiation, which may include some additional physical coordinates in addition to the radiation coordinates. For most purposes these additional coordinates will be restricted to the radial velocity of a part of the source in the chosen reference frame. For each measured coordinate, an IntensityModel should also provide an (optional) internal error estimate. To represent the significance of the data an IntensityModel should provide an (optional) weight to be used in combining different IntensityModels. The primary differences between Measurements and IntensityModels are in their use of instrumental coordinates vs. radiation coordinates respectively, and in the existance of methods to combine IntensityModels as part of the IntensityModel objects. In principle, Measurements should only be combined after first using a TelescopeModel to convert them into IntensityModels, although a knowlegable user will have no conceptual problems with averaging the components of a set of raw Measurements.

MathComponent

— an object defining a collection of arrays and functions which may be used to represent data. These are essentially the mathematical classes defined in Note 142, Figure 2, perhaps supplemented with the FuncOfChans for frequency and wavelength defined in Note 142, Figure 4. Although these classes were considered to TelescopeComponents in Note 142, these classes by themselves are purely descriptive; they do not have obvious operations "setup", "solve", or "apply" in relation to a Measurement. In Figure 1 (see below), they have therefore been moved out of the classes of TelescopeComponents and under the less restrictive umbrella class "Component". See also the discussion for RadiationComponent. Note that a general MathComponent in Figure 1 is explicitly allowed to consist of a whole collection of simpler MathComponents.

Measurement

— a block of data output from an instrument at a particular time. This is the basic unit of input data for most direct calibration schemes. A Measurement will consist of n-tuples of instrumental coordinates and arrays of measured values, making up a MathComponent, together with sufficient identifiers to distinguish it from all other instrumental yegs. The indices into the arrays form a set of implicit instrumental coordinates. To form an atomic yeg from one of the measured values it is necessary to provide the measured value itself plus the Measurement identifier and either the identity of the n-tuple or the identity of the array plus the indices into the array. Note that a Measurement need not consist of a aggregation of atomic yegs (which might be very expensive in storage), but that it must always be possible to refer to a data item within the Measurement as part of an independent atomic yeg. A Measurement always has an associated InstrumentModel which defines the meaning of its component parts. A Measurement may optionally have an associated IntensityModel defining known values for some of the radiation coordinates which will be used during calibration.

Radiation Coordinates

— the six quantities

$$I,\ \nu,\ t,\ l,\ b, \text{ and } P,$$

where I is a measure of the intensity, \nu represents the frequency or wavelength as appropriate, t is time, (l,b) are the longitude and latitude in an appropriate spherical coordinate system on the sky, usually right ascension and declination, and P is a polarization state.

RadiationComponent

— an abstract Component which is directly related to the sky intensity. There are two concrete subclasses, which are the TelescopeComponent and IntensityModel classes. The primary duty of a RadiationComponent is to describe how it interacts with each of the radiation coordinates. Since many components are not sensitive to particular radiation coordinates, the inclusion of any particular coordinate in a RadiationComponent is optional. Although not a formal requirement, each coordinate in a RadiationComponent may be expected to have several common properties:

- a mean value,

- an optional error estimate for the mean,
- a sensitivity function, eg. beamshape, sensitivity profile, etc.
- a width parameter for the sensitivity function (beamsize, channel width, etc.)

Note that the width and the error estimate might be the same quantity for a measured coordinate, in which case the "sensitivity function" could describe the expected statistics (Gaussian, Poissonian, etc.).

TelescopeComponent

— one of the classes describing how a telescope and its environment interact with the intensity of radiation from the sky. Specifically, they describe the processes which convert incoming radiation into data in a Measurement, and provide the methods needed to invert these proceeses, at least approximately. They are broken on physical grounds into five broad subclasses describing the atmosphere, the individual telescope antennae, the instrument package, the motion of the platform (the Earth or a spacecraft), and the interactions which tie the set of TelescopeComponents into a whole telescope. Each of these components may interact with several, and perhaps all, of the radiation coordinates and are therefore subclasses of the abstract RadiationComponent class. For example, the atmosphere does not usually affect frequency, but does shift wavelengths; timing delays in the atmosphere and ionosphere are an important consideration in interferometry; refractive corrections may significantly disturb the apparent location of an object and differential refraction is a serious nuisance in some kinds of photometry; correcting for atmospheric opacity is a major problem at most wavelengths shorter than 1 cm. A similar analysis shows that each of the other kinds of TelescopeComponent must also be considered to be a RadiationComponent.

Yeg — a general term for a piece of astronomical data which will normally be qualified to indicate the nature of the unit involved. The qualification may be given explicitly, or implicitly from the context of the discussion. If the qualification is not clear from the context and is not given explicitly, **the word will refer by default to either a Measurement or an IntensityModel** (see the discussion in the next section), which are the normal forms for raw data and calibrated data respectively. A yeg consists of one or more measured values together with sufficient identifiers to distinguish it from all other yegs, and optionally a block of associated information. The identifiers and optional associated information will be referred to collectively as the header of the yeg, and the measured values will be referred to as the data. Note that some quantities are ambiguous, and may be considered either as data or as part of the header depending upon the context; the traqditional distinction between header and data is sometimes quite blurry in the proposed models, and should not be taken is more than a convenient grouping of the information in a yeg for purposes of storage.

Yegset — an aggregate of yegs. An atomic yeg by itself, a calibration group and an entire dataset all constitute valid yegsets. If otherwise unqualified, the term will usually refer to the entire dataset under consideration. A yegset may be qualified by any desired selection criteria.

2 Relationships to TelescopeModels

The three most important quantities in a data reduction system are the raw measurements, the TelescopeModel used to interpret the measurements, and the IntensityModel which represents the calibrated output of a TelescopeModel applied to a Measurement. This section will consider the relationships which define these three objects, as illustrated in Figure 1.

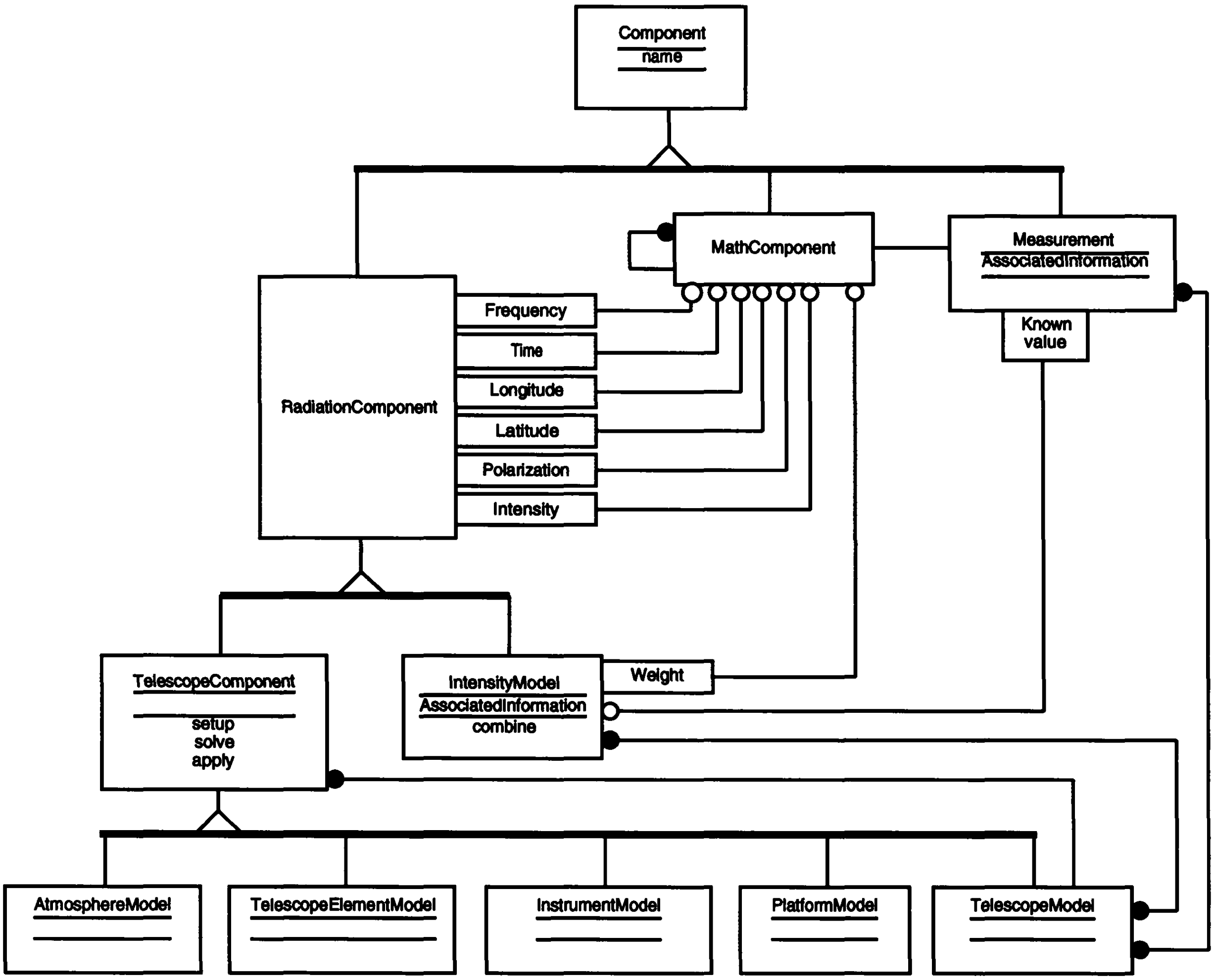

Figure 1 — Relations Among TelescopeModels, IntensityModels and Measurements

The Measurement is conceptually the simplest object in the system. It is simply a collection of numbers, which we may organize into a MathComponent for the purpose of discussion. Generally speaking it contains the raw output of an instrument package, often with some additional information appended by the telescope control system. At this level of discussion there are no constraints whatever on the format of either the data or the AssociatedInformation. It would, of course, simplify many peoples lives if the data could be packaged in some standard format, such as the extended FITS being discussed for AIPS++. Some of the data in a Measurement will represent actual measured values of some instrumemtal coordinate; these may or may not have an associated estimate of their internal error. Other instrumental coordinates will be present implicitly as offsets into arrays of measurements.

Directly related to each Measurement is a set of TelescopeComponents which may be used to calibrate the Measurement. Although it should be possible to use each TelescopeComponent individually, they will normally be coordinated through a suite of TelescopeModels. It is the responsibility of these TelescopeModels to be able to read and interpret the Measurement into radiation coordinates. In essence, a Measurement is meaningless until it has been associated with the correct suite of TelescopeModels. Each TelescopeModel has three methods which define its interaction with the measurements and IntensityModels:

- setup() — defines constant and default parameters for the TelescopeModel,
- solve(Measurement&) — solves for internal parameters of the TelescopeModel,
- apply(Measurement&,IntensityModel&) — calibrates a Measurement, returning an IntensityModel.

An IntensityModel provides a description of the relevant properties of the intensity. An IntensityModel may have internal coordinates such as an index into an array, but must define all of these internal coordinates in terms of physically meaningful units such as the radiation coordinates. IntensityModels are not restricted to just the radiation coordinates and may, in principle, include an arbitrarily complex description of a source. Construction of complex source models should, however, be left to individual users. For calibration purposes to only physical variable likely to be of interest outside the radiation coordinates would be the radial velocity of the source in the chosen reference frame.

During calibration, an IntensityModel is associated with a calibration group of Measurements via a suite of TelescopeModels. An IntensityModel is refered to as "fully calibrated" when all of the information in the calibration group of Measurements which is accessible through its associated suite of TelescopeModels has been used to define the corresponding quantities in the IntensityModel. If only some of this information has been transferred, the IntensityModel is "partially calibrated". Once calibration is completed, however, the link to the suite of TelescopeModels is broken. A fully calibrated IntensityModel is an independent object which carries its own description inside itself.

An IntensityModel necessarily implies a parameterization of the sky intensity. The actual sky cannot be parameterized in any meaningful way, since it contains arbitrarily fine details (the shadow of a pebble on the outermost moon of a gas giant planet orbitting a G-type star in the disk of a galaxy containing a quasar seen at a redshift of 3.81 . . .). However, our instrumentation smooths the sky intensity on a scale defined by the TelescopeModel (beam-size, channel width, sample time interval, etc.) and our measurements can easily be interpreted as sampling this smoothed representation of the sky intensity, which we might refer to as the PerfectTelescopeSky parameterization. It is important to realize that the PerfectTelescopeSky is only one of many possible representations of the calibrated data, and an end-user of this package may want to define their own TelescopeModels producing IntensityModels in some other system incorporating their own, physically meaningful parameterization of the sky intensity. However, the PerfectTelescopeSky is the parameterization which will yield the closest correspondance between the raw data and the calibrated data, is extremely general, and is easy to understand. Looking ahead to the implimentation phase, the immediate goal of direct calibration (`apply`'ing the TelescopeModel to a Measurement) should be to produce data parameterized for a PerfectTelescopeSky. For most single dish work, this calibration will be sufficient in itself. For the more complicated, iterative calibrations which can occur in interferometery the data can be processed in a loop, with the final image I(l,b) feeding back into an InitialModel used by the TelescopeModel so that the initial calibration of the Measurements produce the correct IntensityModel for a PerfectTelescopeSky in the dual space I(U,V). (Interferometrists, read that last sentence again slowly and tell me, IS THIS RIGHT, in principle if not in detail?)

Considered from another angle, a calibration group of Measurements in association with their TelescopeModels very nearly constitutes an IntensityModel by itself (it only lacks formal methods to combine with other IntensityModels). In this case the "internal physical coordinates" have to do with the telescope and its surroundings, which may be of considerable interest to an engineer or an atmospheric scientist, but will probably not have a lasting interest for most astronomers.

Generally speaking, an IntensityModel, its associated TelescopeModel, and the Measurements to which they both refer, will all share a common structure and this structure may be used to simplify the construction of all three objects through the use of a common MathComponent. For example, a 2048x2048 CCD will be represented by an array of numbers representing the flux entering each pixel, with the axes of the array representing angular offsets on the sky. This same structure will be present in the raw data, in the final image, and in the flat field and dark current images used to calibrate the image.

For calibration purposes, a Measurement will often carry an associated IntensityModel, whose relevant coordinates are known in advance. As examples, a DARK observation is known to have a signal strength of 0 in all its channels, a hot load calibration will have a signal strength of T_HOT, and a spectral line whose velocity is being measured will have a known rest wavelength. Generally speaking these "known" IntensityModels will be extremely simple and will not occupy much storage.

It will be recognised that Figure 1 includes a major elaboration of the definition of a TelescopeComponent. Each TelescopeComponent now explicitly carries an option to define the behaviour of the output IntensityModel on each of the radiation coordinates. In the figure this is indicated by an optional association with a MathComponent, but it should be understood that each stub also implies a function which transforms the data in a Measurement and appends it to the output IntensityModel. The associated MathComponent simply carries the parameters of the transformation. That this definition is both necessary and sufficient may be recognized by noting that the radiation coordinates completely define the electromagnetic radiation coming from the sky; the behaviour of a TelescopeComponent will thus be completely specified by its action upon the radiative coordinates. In many cases, the net effect on the incoming radiation may be computed by applying each TelescopeComponent in turn. In other cases a coordinating TelescopeModel must be used to connect several TelescopeComponents, such as when analysing a skydip which involves an AtmosphereModel (zenith optical depth), a TelescopeElementModel (telescope efficiency as a function of elevation), and an InstrumentModel (receiver temperature).

As an important aside, it will be recognized in this that the Longitude and Latitude stubs are tightly bound together, representing an arbitrary choice of the many possible coordinate pairs which may be imposed upon the celestial sphere, globally or in patches. There will surely be other objects defining coordinate systems and their transformations which will be attached to these stubs as appropriate. Definition of these coordinate systems lies outside the scope of this note.

In Figure 1 the natural units of data are the Measurement and the IntensityModel. Since these two objects are so intimately related, representing raw and calibrated data respectively, we propose that they be taken as the definition of a Yeg. This is consistent with the definition of Yeg given in the glossary of the Project Book (as of 1992 August 24), but not with the definition given under Yeg Set which states that a yeg is always an atomic yeg. It is perhaps not surprising that a discussion oriented around calibration should favour a definition of yeg which makes calibration look simple, but this problem is sufficiently important that it merits some careful thought.

Consider first a Measurement from a complex device. Most instruments are designed to be conceptually simple, with internal coordinates which map directly onto the radiation coordinates and the data represented as simple arrays. Supposing the measurements are represented by arrays (x[], y[], z[]), if x, y, and z have the same dimensions it is usually possible to associate the array entries into atomic yegs as (x[0],y[0],z[0]), (x[1],y[1],z[1]), and so forth. Sometimes the dimensions of the arrays are different, but the associations are still simple, say (x[i,j],y[i],z[j]). In this case too the atomic yegs are well defined, although their storage is more complex. A more complex device might require complicated associations between items, as in Figure 2. If these remain stable from one Measurement to the next then each Measurement may be viewed as an aggregate of distinct atomic yegs and the whole yegset will simply be heterogeneous. This in fact happens all the time when different observing modes are used with the same backend, or when different backends are harnessed together. If, however, the associations vary unpredicatably from one Measurement to the next, giving a free-format data stream, then even the definition of atomic yeg becomes difficult. This final pathological case is fortunately quite rare, so that the vast bulk of raw astronomical data can be broken into atomic yegs without serious difficulty.

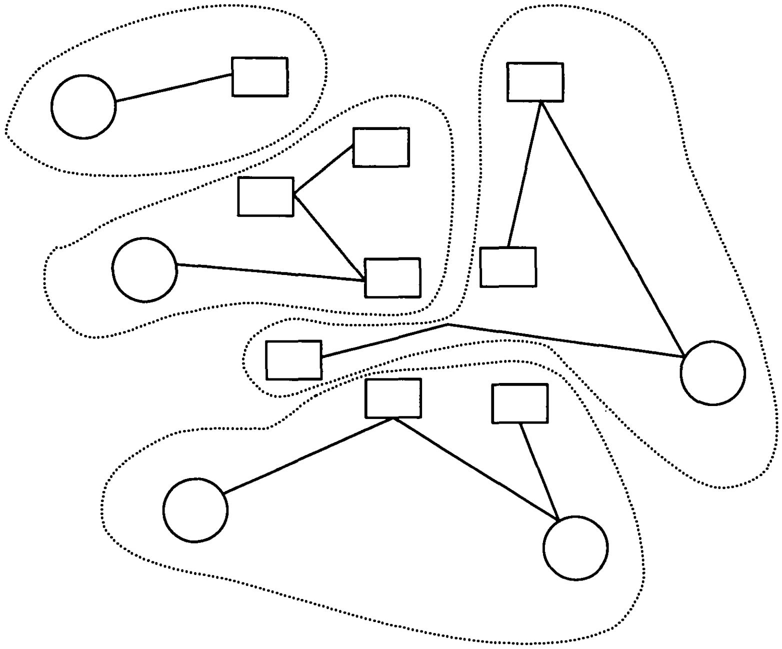

Figure 2 — A Measurement with a Complicated Atomic Yeg Structure

Things are not always so clear for the IntensityModels representing calibrated data. To the extent that they mirror the original raw data, the same considerations should allow IntensityModels to be broken into atomic yegs with their components drawn from the radiation coordinates. Further processing of the data, however, may yield IntensityModels with complicated functional representations, such as a sum of elliptical Gaussians on a polynomial background. Nothing in the specification of a MathComponent requires that it break easily into linearly independent pieces with simple, identical parameters, nor that the resulting parameters have simple physical interpretations. It would be perverse to insist that an object as complex and extended as an elliptical Gaussian should be treated as an atomic yeg on the same basis as a pixel in the image to which it was fitted. It is clear that the IntensityModel still constitutes a yegset, and it should be possible to extract the same kind of information by applying a selection criterion. Intuitively, we would like an atomic yeg to represent the finest granularity in the radiation coordinates which we believe to be physically meaningful, regardless of the internal representation of the IntensityModel. How this might be represented in general seems to be a difficult problem, and not one which can be swept under the rug, considering the ubiquity of parameterized image models in data processing.

Because of the difficulties encountered with atomic yegs in IntensityModels, it seems preferable to reserve the useful term "yeg" for Measurements and IntensityModels, where its meaning is unambiguous, and to limit the term "atomic yeg" to those (very common) circumstances where the atomic units are easily identified and have simple interpretations in the radiation coordinates.

It is useful to remember in this discussion that an atomic yeg is truely a unit of data seen from the user interface. It does not need or imply that the atomic yegs are easy to identify in the storage or the internal representation of a yeg. It is sufficient if each Measurement and IntensityModel knows how to present an atomic yeg to the user interface upon request. In the terminology of Rumbaugh et. al., an atomic yeg is a derived object, determined from either a Measurement or an IntensityModel.

3 The Structure of Single Dish Data Reduction

Figure 3 shows the general flow of information during a single dish data reduction session. This figure is conceptually very similar to Figure DLSCALIMGFM in Section 2.4.1.1 of the Project Book except that the inversion step, which is rarely necessary in single dish work, would be buried in the Combine process, and there is almost never any need for the complex feedback loop from the Image (a form of IntensityModel) to the TelescopeModel.

Two other minor changes are the explicit inclusion of a manual facility to edit the data stores and the ability to setup a TelescopeModel from an observatory database. It is expected that the editor will be capable of examining the data either graphically or, like a spreadsheet, as a binary table whose entries can be modified in blocks. The setup function of the observatory database should be easy to include since it could use exactly the same kinds of selection criteria, file formats and access routines as would be needed to read and write IntensityModels to the regular yegset datastore.

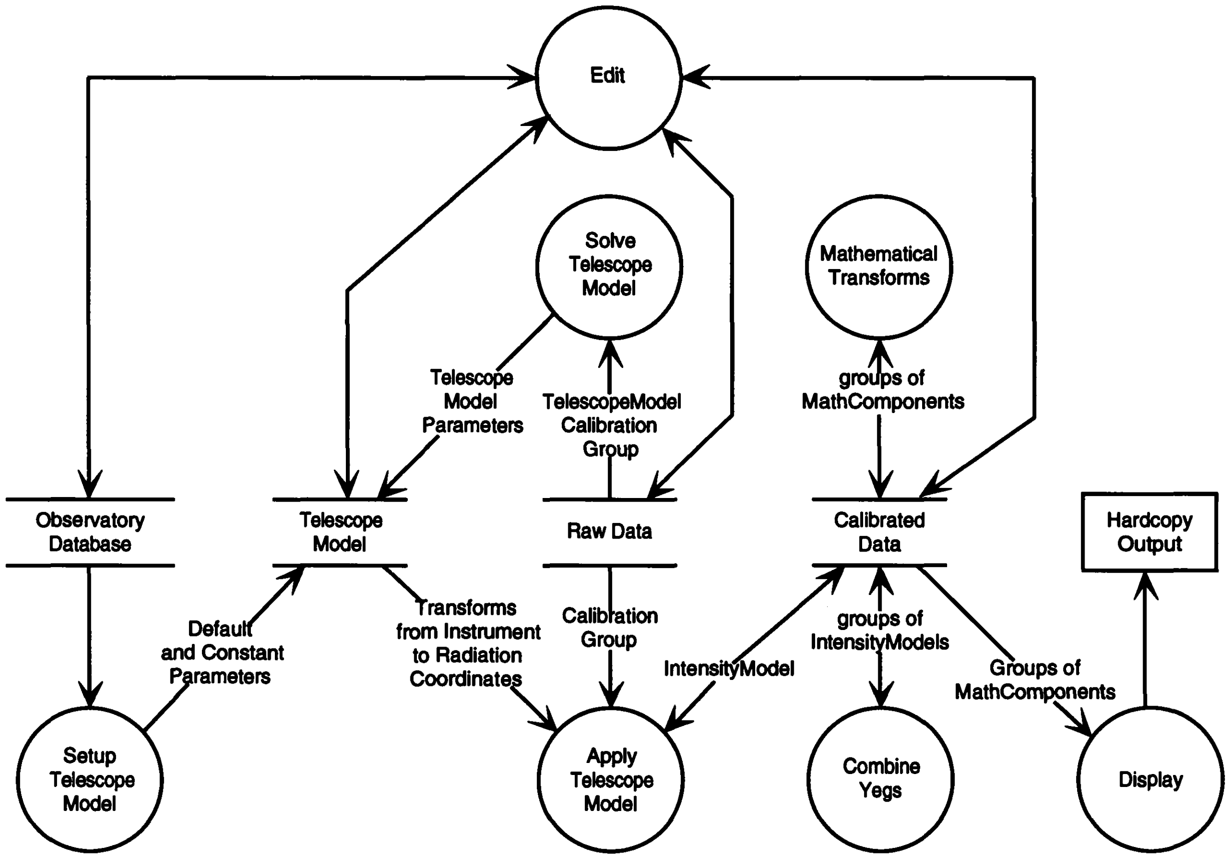

Figure 3 — A Normal Single-Dish Data Reduction Session

Two conceptually different kinds of operation may be used to manipulate a calibrated yeg. The most important routines combine or modify the data in ways which affect the formal significance of the data, either by modifying the sample points of the independent radiation coordinates, or by changing the weights assigned to the data points. The most obvious of these processes cobines two different yegs in a weighted average, so the whole set of processes is labelled in the figure as "Combine Yegs". A second set of transforms acts directly upon the arrays of data without using its formal significance. These include baseline removal, unweighted scan arithmetic such as dividing one spectral line by another, and even the CLEAN procedure used in VLBI. Most often these methods are used to remove undesireable artifacts left over from an inadequate calibration procedure — perfectly calibrated data, after all, would not have baseline problems. From a programmers viewpoint, the two classes of methods are distinguished by their arguements; a Combine operation requires an IntensityModel (possibly several) for input, whereas a Mathematical operation requires only a MathComponent (or set of MathComponents).