SIAP Extension RFC and Draft Specification, Part 1: Quantities and Coordinates

Steve Lowe, SAO/CfA
Version 1.0
April 22, 2003

This document is a draft Specification and also a "Request for Comments" on extending SIAP, presenting some issues and suggestions based on discussions by the NVO group at SAO/CfA over the past weeks. Since these issues are complex, this document will be distributed in several parts and at different times.

The present installment considers the addition of new quantities to SIAP queries.

Discussion is presented to explain our view and to answer some objections that have been made both in our internal discussions and also on the NVO mailing lists. *One of our objectives in distributing this document is to stimulate discussion on this subject.* We look forward to a vigorous response.

While it may take some time before agreement is reached on a final Specification, our group at SAO/CfA will be proceeding with an implementation based on the current draft. That implementation will be used in the SAO/CfA science prototypes.

SIAP is only a temporary standard, so we should not become too embroiled in creating a protocol that is "exactly" right. Nevertheless, some of the issues that arose in our internal discussions bear on the general philosophy of querying archive services, so they may be worth thinking about now in the SIAP context to learn some things that can be applied when a permanent successor protocol is defined.

One of our guiding ideas is that when we add to SIAP, we will do so in a general way. For example, to add spectral properties, we devise a generic scheme to add new properties to SIAP, and then apply that scheme to spectral data.

In the proposal we are making, *compatibility with SIAP 1.0 is preserved as far as possible*. In addition, the features that are being proposed here are nearly all optional for compliant services, so very little change will be required for existing image services.

1. Principal Ideas

The elements of the proposed changes to the SIAP 1.0 query are as follows:

a) Physical quantities or properties other than spatial position, such as time, wavelength, flux density and polarization state can be used. *An archive service need only support quantities that apply to its data.* Syntax will be provided for the query to "declare" a quantity that will be used in the query.

b) In general, these quantities will appear in a manner analogous to spatial position, for example as arguments to the POS and SIZE keywords. For example, a space-time-wavelength position in with units of degrees, degrees, MJD, and Angstroms might be indicated by POS=148.9,69.1,44000.0,6600.

c) The syntax for declaring a quantity will permit the specification of the coordinate system and units that apply to the numerical values in query input parameters.

d) The structure of the data product sought can be indicated. The return product may, in general, be thought of a sampling of values of a one or more functions of several variables (for example, a data cube). The data product structure is defined by indicating the physical properties sought as the arguments and value(s) of these functions.

e) Very importantly, the archive service response to the Metadata Query (FORMAT=METADATA) will allow the service to list the physical properties that it can understand as input parameters, and the structure of the data products it has available.

f) Analogs to Image Generation Parameters (IGP) for pointed or atlas (non-mosaicking) archives will be defined. For these services it is natural to specify a range limitation on data product characteristics instead of prescribing values as in the IGP. We have in mind particularly restrictions on resolution and images size.

g) Some issues regarding projections will be addressed. Strictly speaking, in SIAP 1.0, the region of interest is specified in terms of a projection of the sky. This seems to be a special instance of a more general issue.

h) A list of proposed labels for declaring quantities, coordinate systems and units will be proposed.

The present documents describes our thinking on items a)–e), which form the major conceptual framework of our proposal. f)–h) will be addressed in subsequent documents.

It should be clear from the above list of items that this proposal is influenced to a considerable extent by our orientation toward Data Models.

2. Why are we extending SIAP?

The immediate reason we want to extend SIAP is to enable queries to specify a request for spectra rather than for conventional images, to support the several spectral archives at CfA. This at least requires a way to indicate that the client expects a data product that has a spectral axis. It would also be useful to be able to specify the waveband of interest. In addition, some data products are more complicated than simple images, such as data cubes, and in some cases the observed quantity is something other than flux density, such as radial (Doppler) velocity.

3. What does the query ask for?

Based on the discussions that our group has had and on the SIAP 1.0 specification, the following seems to be the general expectation of what a query is trying to ask.

First, it defines a Region of Interest. In SIAP 1.0 this is just a patch of sky, but we are proposing to expand it to cover time and wavelength, as well as other dimensions.

Second, it specifies the type of data we're looking for, such as an image (brightness at an array of positions on the sky), or a spectrum, or a time series. The axes of the data could differ from those of the Region of Interest. Included in this point is the specification of the physical quantities observed or measured.

Third, it can indicate our preference or requirements for content-related attributes of the data, such as size and orientation of an image, or resolution along one of the axes, or the projection used for an image.

Fourth, it can indicate our preference for format of the returned data, such as filled arrays vs. tables (the latter possibly representing a sparsely filled array or pixel list), FITS vs. JPEG, etc.

We should keep in mind that some of these elements could be eliminated from query capabilities and moved to the metadata contained in the candidate list returned by the query service. Then the inquiring client would have to filter the list.

4. Does anything actually need to be done?

To a certain extent, SIAP already enables a spectral data archive to be queried. For example, to query the CfA's Z-Machine archive, which contains 1-D spectra, the existing interface expects a sky position and a search radius—essentially, a cone search. All that is needed is for the service to define a new parameter that the client can use to indicate that the data product it is looking for is a spectrum rather than a 2-D image. (A client that does not send the new parameter in its SIAP query will receive an empty list from the service. This will protect "spectrally-unaware" clients from mistakenly retrieving data products that are not images.) Additonal metadata fields could also be defined for the query response VOTable to describe spectrum-specific characteristics of the datasets. Also, the set of allowed input parameters could be expanded to permit the query to indicate constraints on spectral aspects of the data requested.

A service can make these definition changes unilaterally within SIAP 1.0. However, adding them to the standard would have the advantage that any client looking for spectral data would know what parameters to use to do so. Furthermore, although SIAP 1.0 provides a way for the *existence* of new parameters to be described to a client in the Service Metadata query (FORMAT=METADATA), there is no way to describe the *semantics* in a machine-interpretable way. This means that a software client looking for a spectrum service doesn't know how to recognize parameter names that correspond to spectral information.

Furthermore, adding other physical properties to the query can be useful even for users seeking images. Consider that, at present, there isn't a way to constrain the waveband of returned images.

For these reasons, we think that there are some modifications that are worth making to the standard. Specifically, we will propose a framework in which other physical properties, such as wavelength, time, velocity, polarization and so on may be referred to in the query language in a manner similar to the current spatial parameters. For example, the region of interest can be specified as a space-time-wavelength box. Also, the quantities of interest in the data product sought can be identified.

5. Are spatial coordinates special?

SIAP 1.0 focuses on spatial coordinates in the query, and since we are going to propose a treatment in which sky position is just one of several "quantities" involved in a query, we should address the question of whether this is a sensible approach. Is there something special about spatial coordinates that would argue against combining them with other quantities?

There are several features of sky coordinates that distinguish them from other quantities:

- For many existing archives, sky position is the primary or only search criterion.
- The sky is 2-dimensional, so regions are more complicated to describe than for 1-dimensional quantities such as time or wavelength. In particular, in one dimension a connected region is just an interval described by its two endpoints (plus 2 bits to indicate whether the endpoints are included), or a union of such intervals
- The sky is not flat. This creates annoyances such as the fact that the size and shape of a coordinate "box" depend on position. It also means that we tend to use flat projections when possible, and that means software has to be prepared to deal with the transformations between these projections and the "real" sky.
- There are several reference coordinate systems in use that differ by more than just a multiplicative constant (i.e., more than just different units).

The fact that sky position is the predominant search criterion may just reflect the way data is currently organized. Since an archive is often associated with a single instrument or collection of similar instruments, wavelength is already somewhat narrowed down just by the choice of archive, so sky position becomes the most prominent constraint that has to be explicitly stated for a search *within* an archive.

If a VO objective is to start thinking of the union of all archives as an entity to which queries can be addressed, even indirectly, then there is reason to define the way in which other quantities can be used in queries. Is it a goal of the VO that an astronomer can pose an information request to the union without first identifying particular archives to search? If so, then it is worth studying

- How that request is posed, and
- How an archive can respond, allowing that the archive may have limited “intelligence” outside of its own specialty.

This issue is related to the development of registries. A registry could serve either as a means for astronomers to locate relevant archives to search, or as a means for client software to locate and interrogate archives to satisfy a user request. In the latter case, machine-interpretable semantics are needed for the query and the service description.

Concerning coordinate systems and units, it is true that the calculations needed to convert between the various sky coordinate systems are complex. However, in concept it is not so different from unit conversion: in each case there is a coordinate-independent quantity (e.g., a location on the sky or wavelength) whose specification requires numerical values augmented with a coordinate system to give meaning to those values. If in some cases the only coordinate systems that are in common use are related by multiplicative constants (i.e., units like Angstroms vs. meters) then that is just a happy convenience. In other cases, the coordinate system definition may be traditionally separated into elements such as a units (degrees vs. arcseconds) plus a more complicated reference system (FK4 vs. FK5), but one could still consider the composition of these elements to comprise the “coordinate system” that links numbers to physical properties.

Also, whether unit conversions are the only transformations required may depend on the level at which concepts are described. For example, in vacuum the properties of wavelength, frequency and photon energy are interchangeable, but are not all related by unit conversions. So, one might define a property which is “the spectral property of electromagnetic radiation in vacuum,” EM_SPEC, which can be expressed using any of the three “reference systems” WAVELENGTH, FREQUENCY and PHOTON_ENERGY, plus units appropriate to each system.

This last point *might* be stretching these ideas a bit, but maybe not. Why shouldn't our query engines be expected to be smart enough to convert between frequency and wavelength, if they can convert between galactic and ICRS coordinates?

As to the complexities due to the dimension and curvature of the sky, these might be just a matter of degree rather than a fundamental difference from other quantities. Compared with a one-dimensional quantity, it may be more complicated to specify a location or a region on the sky's geometry, but conceptually it is the same. Furthermore, there are other multidimensional quantities that we may want to consider, such as polarization state.

6. Summary of Proposal

6.1. New capabilities

1) Most importantly, allow specification of what types of physical quantities are involved in the query.
2) Allow the Region of Interest to be constrained to ranges of other quantities besides spatial position, most immediately wavelength and time.
3) Allow the client to indicate the physical quantities that are to appear as axes of the requested data product.
4) Allow client to specify limits on resolution.
5) Add elements to the definition of the Service Metadata self-description response (to the FORMAT=METADATA query) so that the service can identify which quantities can be used.

6.2. Coordinates for input values

A fundamental capability that is needed in a query is to be able to specify values for physical quantities. To do this three things must be identified: what the *quantity* is; the numerical *value* for the amount of the quantity; and a *coordinate system,* including *units*, that translates the numerical value into a definite physical system. Here, the coordinate system includes everything required to give the number unambiguous meaning, such as units and reference frames.

From the point of view of the protocol, the combination of quantity and coordinate system (including units) will be indicated simply by a single *string* of characters. For example,

```
SKY:DIST:EQUAT:ICRS:(DEG,DEG)
EM_SPEC:WAVELENGTH:ANGSTROM
```

Now, before we get embroiled in a debate over names and delimiters, let us admit that the detailed labels and syntax shown here may not be quite the best. At this point we wish to concentrate on the framework and present some examples as a focus for discussion.

The labels would by standardized much like the CDS UCD list so that issues of order and ambiguity in the construction would be avoided. Although the labels do not, strictly speaking, have a structure, there is a rationale for the definition of the strings: the more abstract properties appear to the left and the more concrete to right.

The interpretation of the above strings is as follows. **SKY** indicates that positions will be referred to some 2-dimensional projection of thesky; **SKY:DIST** denotes an idealization of the sky involving distant objects for which the precise observing position within the solar system is unimportant. We introduce **SKY:DIST** anticipating that other projections may need to be introduced later, for planetary astronomy for example. These projections of the sky, which would be denoted as something like **SKY:XXXX**, will be treated in a subsequent document. **SKY:DIST** is the sky implied by SIAP 1.0.

EQUAT indicates some version of an equatorial coordinate system, **ICRS** means the ICRS reference system; other equatorial systems might be **EQUAT:J2000:FK5** or **EQUAT:B1950:FK4**. Finally, **(DEG,DEG)** implies that the units for the two axes are each decimal degrees. Similarly, in the second line **EM_SPEC** denotes the "spectral property of EM radiation" as discussed in section 5 above. Then **WAVELENGTH** refines that concept to wavelength as opposed to frequency, and finally **ANGSTROM** supplies the units.

These strings will be called *coordinate system specifiers*.

The service will indicate what quantity/coordinates/unit combinations it can accept by a simple list in response to the FORMAT=METADATA query. The client can specify only quantity/coordinate/unit strings that appear in the list.

For example, the **ROI_COORDS** input parameter indicates the quantities that the query will specify the Region of Interest:

```
ROI_COORDS=SKY:DIST:EQUAT:ICRS:(DEG,DEG),
           EM_SPEC:WAVELENGTH:ANGSTROM
```

Then **POS** and **SIZE** each provide three numerical values:

```
POS=148.89,69.06,6200
SIZE=2,2,3800
```

This query is expressing an interest in *optical* data (a 3800 A wide window centered at 6200 A) in a certain 2x2 degree patch of sky.

6.3 Coordinates for data products

In general, a data product represents a scalar- or vector-valued function of one or more variables. This function is typically measured or sampled at a discrete set of points. The quantity or quantities represented by the values of this function will be called the *observable(s)*. The independent variables (or parameters) of the function will be called the *axes*, similarly to the FITS pattern of storing such data along axes of a data cube, hypercube or pixel array.

In this proposal, the query can indicate the structure of the data product by designating the quantities that form the axes and observable(s). For example, a traditional image might be requested by

```
AXIS_COORDS=SKY:DIST:EQUAT:ICRS:(DEG,DEG)
OBS_COORDS=EM_INTENSITY:FLUX:ERG/CM2/SEC
```

where the observable is the intensity of the electromagnetic radiation, specifically the quantity flux, measured in erg/cm^2/sec. Similarly, a 1-D spectrum might be requested as

```
AXIS_COORDS=EM_SPEC:WAVELENGTH:ANGSTROM
OBS_COORDS=EM_INTENSITY:FLUX:ERG/CM2/SEC
```

However, the requester need not be completely specific. It can allow the archive the option of returning an available data product in whatever units are convenient by giving a partial coordinate specification, as for example:

```
AXIS_COORDS=SKY:DIST:EQUAT
OBS_COORDS=EM_INTENSITY
```

to request an image using equatorial coordinates and any flavor of brightness.

To request a spectral data cube (3-D spectrum) giving flux as a function of spatial position and wavelength:

```
AXIS_COORDS=SKY:DIST:EQUAT,EM_SPEC:WAVELENGTH
OBS_COORDS=EM_INTENSITY:FLUX
```

To request a recession velocity (Doppler velocity) map on the sky in galactic coordinates:

```
AXIS_COORDS=SKY:DIST:GAL
OBS_COORDS=VELOCITY:DOPPLER:KM/SEC
```

The list of proposed labels will be the subject of a subsequent Part of this document series. This may be closely allied with the UCD development work being conducted in the Data Model group. We solicit suggestions. Note, however, that *services are not required to recognize all of the defined labels.* See the section on Metadata Query below.

One last note: it may be that these labels should be case-sensitive. This is especially relevant for the components specifying units, where mW and MW must be considered distinct for example.

6.4. Coordinates for image generation and other input values

Application of coordinate specifiers to image generation parameters (more generally, *data product* generation parameters) and other input parameters (*resolution* in particular) will be deferred to the next article in this series, *Part 2*. Two general points are worth pointing out here:

- Whenever a numerical value is transmitted in the query, the full coordinate specification including units is needed.
- Data product generation parameters applied to archives that do not construct custom datasets (e.g., not mosaicking services) will generally need to be ranges of values. This is because one is unlikely to find an exact match in image size, pixel spacing, orientation, etc.

6.5. Metadata Query

SIAP 1.0 recommends that services provide a self-description in response to a query requesting FORMAT=METADATA. It is proposed that one component of this information be a list of coordinate system specifiers. In addition, for each specifier it will be indicated what role(s) it is permitted to play in the query. Some of the roles are

- ROI : Region of Interest.
- AXIS : Data product axis. The query is looking for data products in which the indicated quantities appear as independent variables, e.g., as the axes of a data cube or pixel array.
- OBS : Observed quantity. The quantity that is a function of the axis coordinates, e.g., the values of the elements of the data cube. More than one observed quantity can be requested.

A specifier may be permitted to take more than one role. Note that for ROI the coordinate system descriptions must be complete including units so that the numerical values in POS and SIZE can be interpreted. For AXIS and OBS, which merely request the type of data product, less detail is required.

The METADATA response will contain a component (i.e., a VOTable RESOURCE) containing a two-column table in which each row is of the form

```
role     specifier
```

This table will be called the *service coordinate system list*.

As an example, consider the service coordinate system list that would specify the functionality of an SIAP 1.0 service:

```
ROI      SKY:DIST:EQUAT:ICRS:(DEG,DEG)
AXIS     SKY
OBS      EM_INTENSITY
```

What does this mean? The Region of Interest may be specified as a region of the SKY, and the client must specify that region using equatorial coordinates in the ICRS frame using decimal degrees for positions on the two axes (RA, Dec by convention). The client may request a data product that expresses the intensity of the incident radiation (EM_INTENSITY) as a function of SKY position. (The use of **`EM_INTENSITY`** without any qualifiers such as **`EM_INTENSITY:FLUX`** means that the definition does not commit to a specific measure of intensity, such as flux or flux density.)

If this service wishes to accept either equatorial or galactic coordinates, and also allow the search to restrict the time (in MJD) and wavelength ranges, the definition can be expanded as follows:

```
ROI      SKY:DIST:EQUAT:ICRS:(DEG,DEG)
ROI      SKY:DIST:GAL:(DEG,DEG)
```

```
ROI      EM_SPEC:PHOTON_ENERGY:EV
ROI      EM_SPEC:WAVELENGTH:ANGSTROM
ROI      TIME:MJD
AXIS     SKY
OBS      EM_INTENSITY
```

Note that the only available return product is still an image, but a request for a hard X-ray (4000 eV wide band centered at 7000 eV) image taken during a particular 1000-day interval could be phrased:

```
ROI_COORDS=SKY:DIST:EQUAT:ICRS:(DEG,DEG),
           EM_SPEC:PHOTON_ENERGY:EV,TIME:MJD
POS=148.89,69.06,7000,45000
SIZE=2,2,4000,1000
ROI_AXIS=SKY
OBS_COORDS=EM_INTENSITY
```

6.6 Does the Metadata Query Response Need More Structure?

A question that has been raised in our group regarding this simple string definition is whether the Metadata Query response needs more structure, rather than consisting of unstructured string labels. The string idea may be adequate depending on the actions that need to be performed by the client and the archive service. Actions that are needed to negotiate a query include:

a) Determine whether a service supports a certain property such as **`TIME`** or **`SKY:DIST`**. This is done by the client looking for the desired property string as a prefix of a coordinate specifier in the service's list.

b) Pick a coordinate system to use to transmit query constraint values. The client must select among the full strings found in a).

c) Convert between coordinate systems. Coordinate conversion should probably be encapsulated in a module, and the interface/API for this module could provide a generic function to perform conversion based on the coordinate specifier labels. Thus, the client and service applications could treat the specifier as just an unstructured label, except for the prefix analysis described in a).

It seems that the nearly unstructured labels (perhaps they should be called prefix-structured) are simple and will serve these needs. In addition, the design anticipates that different coordinate issues will arise, say in considering projections, so avoiding building too much structure into the coordinate specifiers may avoid trouble later. In other words, the lack of s<tructure was intentional to keep the design simple and simultaneously completely general.

On the other hand, it is possible that this idea is too cute by half. The issue has been raised of whether unstructured labels will be too ambiguous to parse by any application that needs to identify the quantity, coordinate system and units separately. For example, in the label **SKY:DIST:EQUAT:ICRS**, the first component **SKY** must refer to the "property," because the label must begin with a property; but, is **DIST** part of the property designation or part of the coordinate system designation? It is not apparent from the syntax; software can only know by consulting a master list that designates **SKY:DIST** as a quantity and **EQUAT:ICRS** as a system. For the operations a–c above it may not matter. But if a client application wants to read the achive's coordinate system list, to extract from it a list the available ROI quantities (**SKY:DIST**, **EM_SPEC**) and to present that list to the user, then easy and unambiguous lexical analysis without consulting a master list may be important.

Since it is expected that the coordinate list information will be returned in a VOTable in the Service Metadata response, there would be little additional cost in splitting the specifier string into quantity, coordinate system and unit fields. For example, the service coordinate system list could be a four-column table (VOTable):

role	quantity	coord	unit
ROI	SKY:DIST	EQUAT:ICRS	(DEG,DEG)
ROI	SKY:DIST	GAL	(DEG,DEG)
ROI	EM_SPEC	WAVELENGTH	ANGSTROM
AXIS	SKY		
OBS	EM_INTENSITY	WAVELENGTH	

If this structure is used in the Service Metadata response, a separate question is whether this structure is preserved in the query string. That is, should the query concatenate the components back together like this:

```
ROI_COORDS=SKY:DIST:EQUAT:ICRS:(DEG,DEG),
           EM_SPEC:WAVELENGTH:ANGSTROM
```

or should separate keywords be introduced for the three components, perhaps as follows?

```
ROI_QUANTITIES=SKY:DIST,EM_SPEC
ROI_COORDSYS=EQUAT:ICRS,WAVELENGTH
ROI_UNITS=(DEG,DEG),ANGSTROM
```

A final matter that arises if a decomposition like this is used is deciding on the appropriate components for this factorization. Are these three concepts, quantity/coordinate system/units, sufficiently general? These issues may need to be addressed by consideration of use cases.

7. Conclusion to Part 1

In this Part of the series the basic strategy for expanding the set of quantities that can be processed using SIAP has been described. The focus has been on the quantities and the coordinate systems that need to be used to tie those quantities to numerical values. The syntax for specifying a coordinate system has been presented, and its use for specifying the Region of Interest and the structure of Data Products has been described.

In addition, some of the considerations that went into the design being proposed have been set out for discussion. Some points on which we do not yet have an internal consensus have also been noted.

In the following Parts of the series additional features based around these concepts will be developed. In particular, the following will be proposed:

- Additional query input parameters. Of particular interest to us is ability to specify a limit on the acceptable *resolution*.
- Including projections and coordinate system modifiers that depend on a parameter such as a projection point.
- A list of quantity/coordinate system/unit labels.

We are expecially interested in stimulating discussing and comments in the following areas:

- The general framework presented: expanding query input parameters like POS and SIZE to list values for a number of quantities; specifying quantities/coordinates/units; use of the Service Metadata query to indicate the types of data a service will accept.
- Quantities that should be included. Hierarchies of related quantities, relation to UCDs; names.
- Format of the coordinate system specifier labels: upper/lower mixed case; component organization; delimiters; should data be structured?