International

Virtual

Observatory

Alliance

Simple Image Access Protocol V2 Analysis, Scope and Concepts

Version 0.2

IVOA Note 2008 October 24

This version:
ThisVersion-YYYYMMDD
Latest version:
http://www.ivoa.net/Documents/latest/latest-version-name
Previous version(s):

Author(s):
D. Tody (ed.), F. Bonnarel, M. Dolensky, J. Salgado (others TBA in next version)

Abstract

Version 2 of the Simple Image Access interface and protocol (SIA, SIAP) is a major upgrade to bring SIA into conformance with the technology developed over the past several years for the second generation IVOA DAL (data access layer) interfaces, while at the same time adding important new functionality. In this document we seek to the establish the scope of SIAV2 in terms of the functionality to be provided, as well as develop the concepts for how new functionality such as data cube access will be provided. A first look at the query

interface and query response required is presented, preparatory to defining a first working draft of the SIAV2 interface itself.

Status of This Document

This is an IVOA Note. The first release of this document was 2008 May 15.

This is an IVOA Note expressing suggestions from and opinions of the authors. It is intended to share best practices, possible approaches, or other perspectives on interoperability with the Virtual Observatory. It should not be referenced or otherwise interpreted as a standard specification.

A list of current IVOA Recommendations and other technical documents can be found at http://www.ivoa.net/Documents/.

Acknowledgements

“Ack here, if any”

Contents

1 Introduction

Version 2 of the ***Simple Image Access*** interface and protocol (***SIA, SIAP***) is a major upgrade to bring SIA into conformance with the technology developed over the past several years for the second generation IVOA DAL (data access layer) interfaces, while at the same time adding important new functionality. In this document we seek to the establish the scope of SIAV2 in terms of the functionality to be provided, as well as develop the concepts for how new functionality such as data cube access will be provided. A first look at the query interface and query response required is presented, preparatory to defining a first working draft of the SIAV2 interface itself.

As with all the second generation DAL interfaces, SIAV2 will no longer be a “simple” interface when considered overall, despite the name. A fully featured interface is needed to support sophisticated applications, hence we need a powerful query interface, comprehensive metadata, sophisticated data access capabilities, and the addition of Grid capabilities for authentication, asynchronous data generation, and so forth. Nonetheless the most basic interface can remain fairly straightforward, much as for the first generation interfaces, supporting both a simplified service implementation for limited service functionality such as whole image access, as well as a simple query mode for basic data access.

In time, through the provision of easy to use service frameworks which do most of the work transparently to the data provider, we hope to get the user community and data providers to provide the robust production services which the VO needs to succeed. In the meantime provision of a basic service profile within the overall SIAV2 interface will continue to encourage data providers to provide at least basic access to their data.

2 Architecture

In considering the design of an interface such as SIAV2, it is important to bear in mind the overall architecture of the DAL interfaces (as established over the past five years or so of effort by the DAL working group) to understand what the role of SIAV2 should be.

The DAL interfaces form a class hierarchy as follows:

Generic Dataset
- ***Table*** (TAP)
- ***Image*** (SIA)
- ***Spectrum*** (SSA)
- ***Time Series*** (a variant of SSA)
- *etc*.

In the next section we discuss the *Generic Dataset* concept, to better understand what this will provide and hence what should *not* be included in a typed interface such as SIAV2.

2.1 Generic Dataset Concept

Although there is as yet no actual DAL interface for the ***generic dataset,*** it has always been a part of the design, and is taking shape as we develop the second general typed data interfaces such as SSA, TAP, and now SIAV2.

Unlike the typed interfaces for *image*, *spectrum*, etc., the generic dataset class describes any type of dataset, can be used for global data discovery, and can be used to describe complex data associations by logically associating multiple individual typed datasets. For example, a "complex dataset" might include one or more spectral data cubes in different spectral bands, some 2-D projections or continuum images of the same field, extracted spectra or object catalogs, and so forth.

Ultimately the generic dataset class will have an access protocol like any other DAL interface, including a query interface and query response. The query parameters will form the basis for the typed interfaces such as SSA and SIA, including all "generic" parameters - which is much or most of a typed query interface such as the current SSA. SIAV2 will share the same generic query interface as SSA and the generic dataset. Likewise, much of the query response metadata currently seen in SSA is generic, and would be common with the generic dataset query and other interfaces such as SIAV2.

The generic dataset protocol will however go beyond the typed interfaces in various ways. A major difference is that since it can describe multiple types of data, it can be used to describe complex data associations - probably by forming logical "associations" of multiple records in the query response, as is currently done on a more limited basis with SSA. In other words, the query response will describe both individual typed datasets, as well as higher order associations of those datasets. Metadata extension, which allows arbitrary metadata to be added to the query response [ref: DAL2 architecture] can be used in combination with association to provide a richer description of the data.

Another important feature of the generic dataset query may be the provision for an ADQL-based query, in addition to the parameter based query. The use of ADQL is a better fit to the generic data discovery query than for one of the typed DAL interfaces, since the generic dataset query deals only with static archival datasets: there is no virtual data access at this level, hence no need to specify the parameters required for precision data access (slicing and dicing a cube for example.)

The generic dataset query will be used for global discovery of any type of data, and for describing complex data associations. The typed interfaces (SSA, SIAV2, TAP, etc.) would then be used for the actual data access, including generation of virtual data. In the most general case, a typical access pattern would be as follows:

- Use the generic dataset query to discover any data available meeting the specified dataset attributes, including any complex data associations of the described datasets. Only generic dataset metadata and association metadata is returned at this stage.

- Use one of the typed interfaces, e.g., SIA, to get more detailed metadata for a dataset of interest (this can be limited to a single dataset using the dataset identifier, if desired). This more detailed type-specific metadata, e.g., the image geometry and WCS, may be needed to plan how to access the data.

Finally, do the actual data access, first repeating the typed data query with more detailed parameters to specify the detailed data access to be performed, and then doing a "get" to compute and return the dataset. A *StageData* operation might optionally be performed to initiate an asynchronous job to generate the data, if significant computation is required. The second two steps above can be repeated any number of times, to access the different types of data in an association, or to access different parts of a single dataset, for example repeatedly viewing different spectral bands of a spectral data cube. In some simple cases a direct retrieval option may be included, allowing a dataset to be directly downloaded as the query response.

2.2 Implications for SIAV2

Now that we understand better how the generic dataset query will ultimately be used, the role of a typed interface such as SIAV2 should be more clear.

While SIAV2 (like SSA) can describe logical associations of related images, without requiring a generic dataset query, it does not attempt to describe more complex multi-typed associations, leaving this to the generic dataset query.

As with the current SIAV1, and other typed data interfaces such as SSA, the *queryData* operation provides both a data discovery capability as well as a mechanism (via query refinement) for specifying the attributes of an individual virtual dataset to be generated, e.g., a 2-D image cutout, or a slice of a 3-D cube.

This precise description of the virtual dataset to be generated can then be used to generate and retrieve the data, either directly with a getData if the operation can be performed synchronously (as in all current implementations), or by first issuing a *stageData* request to initiate an asynchronous operation to generate the data. Monitoring of an asynchronous job initiated with stageData would be performed with the standard UWS pattern, but a conventional synchronous *getData* could still be used to retrieve the generated dataset once the job completes. Integration of the UWS pattern into the SIA protocol will be similar to the way it is done in TAP and other second generation DAL protocols. Alternatively, data could be staged to a VOSpace, using VOSpace transport methods to move or retrieve data.

3 Major Capabilities and Features

In this section we summarize the major capabilities proposed for SIAV2. This preserves and extends the more basic capabilities provided by SIAV1, adding an updated query interface, new capabilities for cube data access, and integration of Grid capabilities to provide scalability and authenticated access.

3.1 Basic Capabilities

The following basic capabilities are required:

- Updated query parameters, consistent with the generic dataset. This is already largely implemented in SSA, hence much of the SSA interface will be common with SIAV2 as well. Since SIAV2 is a typed interface, naturally it will have some image specific query parameters as well, e.g., for specifying the attributes of an image to be generated.
- Updated query response, consistent with the generic dataset. As for the query parameters, much of this "generic dataset metadata" has already been implemented for SSA, and we can merely carry this over to SIAV2. This includes things like metadata for dataset identification, characterization, and so forth. Other image-specific metadata will also be required, for example to describe the image WCS and geometry.
- Simple 2-D access, e.g., to whole images or cutouts, should be retained similar to what is currently provided, both for backwards compatibility of existing applications, and to continue to provide simple access for applications which don't require the advanced capabilities of SIAV2.

While not very exciting, these basic capabilities are probably the highest priority features for SIAV2! This part is relatively easy to provide, as it largely carries over from the work already done in connection with SSA.

3.2 Basic Whole-Image Discovery and Access

The simplest SIA services merely describe whole images and allow them to be retrieved. No negotiation of the parameters for virtual data generation is required. This is an important use-case in its own right, but also defines a minimum capability which is relatively easy for a service implementer to meet. The main work required is to precompute some image metadata, which would normally be stored in a DBMS table. The service required is then pretty simple to implement, especially if a service framework is used to handle all the protocol-specific functionality.

The following are proposed to support this mode of access:

- Retain the current POS, SIZE parameters, used for basic position-based discovery and also to define the default region for image generation. As for SSA however, POS would no longer be required in a query - any other query parameters could be used to form the query. The coordinate system reference frame is generalized as for SSA, e.g., allowing galactic coordinates to be specified, or other coordinates such as for solar and planetary data (this is all based upon the STC defined spatial reference frames).

- Add a new REGION parameter as an optional advanced capability. This would allow use of an STC/S (or maybe eventually an uploaded STC/X) region description for discovery of data in arbitrary regions. Most notably this would permit simple discovery based upon shapes such as circle and polygon, but more complex cases become possible as well. This has also been proposed for TAP, and would be a likely addition to all the DAL interfaces.

3.3 Image Cutouts and Mosaics

A major capability introduced already in SIAV1 is support for image cutouts, to provide efficient access to small regions of image data, e.g., for analysis of specific astronomical targets. For a 2-D image the service functionality required is still pretty simple, requiring the image access (access-URL based getData operation) to generate an image cutout at access time, rather than merely return the entire image.

The following are proposed to support this mode of access:

- As before, the distinguishing characteristic of "cutout" type access is that pixel values are not modified; the original pixel or sample values are returned.
- The region of interest (ROI), which was position based for SIAV1, is generalized as for SSA to include all physical measurement axes, i.e., POS,SIZE, BAND, and TIME can all be used to define the "cutout" ROI (BAND and TIME probably only apply to 3-D cutouts).
- We further propose to add support for polarization, e.g., via a new POL parameter. This would support subsetting of polarization data, e.g., for a full-Stokes radio image, which is quite a common case for radio data (probably increasingly so for the newer instruments soon to come online).
- Note that the concept of a cutout is easily generalized to cube data, potentially including spectral, polarization, or time measurement "axes" in addition to spatial. The spectral axis at least may require several choices of units, e.g., velocity as well as wavelength/frequency/energy.
- What to provide for polarization is to some extent a TBD, as we can see for example in the recent discussions on the FITS WCS mailing list. Currently neither FITS nor STC provide full support for polarization; FITS provides some support (sufficient at least for most radio data), however this is not fully general. FITS currently provides only limited support for specifying time; STC is better in this area at present. Neither model fully

addresses all of these measurement axes. Polarization will have a discrete set of possible values (e.g., Stokes, I, Q, U ….). Some of them are exclusive to each other.

When the geometry of the output image is crossing the boundaries of the stored image we are not facing a simple cutout problem but a mosaicing facility. Although most of the parameters will be similar to the cutout case we may have to add a parameter describing the coaddition/resampling mode.

3.4 Cube Data Access

A major new feature planned for SIAV2 is support for access to regularly sampled N-D data, otherwise known as image cube data. This is badly needed as it is a major class of data which is not currently supported adequately by VO, which will become much more common over the next several years as a new generation of radio and O/IR instruments come online.

Adding support for cube data will expand the scope and operational complexity of SIA significantly (for those services that implement this capability), but this does not necessarily mean that the service interface is significantly more complicated. All that is required is to generalize the current 2-D interface to N-D; for 2-D use-cases little changes. The biggest impact is in the semantics of data access, as we need to be able to slice, dice, cutout, etc., cube data.

The following are proposed to support this mode of access:

- Generalize to add support for N-D images (2-4 image axes). Typical cases include a conventional 2-D spatial image, 2-D images with non-spatial axes, a spectral or time cube (3 axes), or a spectral cube including polarization (4 axes). As noted earlier, there are issues concerning how flexible we want to be in specifying the spectral axes (velocity etc. - do we support redshift, velocity dispersion, spectral index, etc., as well?).
- Note that the logical model presented by the service may differ from the storage model used in the actual archive. SIA may view N-D data as a simple cube, but data may be stored in the archive as multiple files, particularly for large datasets or data with multiple polarization or time samples.
- The axis order as seen in data returned to the client probably needs to be arbitrary. For example we might have axes 1-2 be spatial and 3 spectral or time, or we might have any transposition of these axes.
- Access to cube data (or an image of any dimension) is based upon specifying the geometry and WCS of the data the client wants to get back (SIAV1 can already do this for 2-D data, via the “image generation parameters”). Some degree of compatibility with both FITS WCS and STC is required, but neither addresses the full problem currently.

- The primary modes of access are “cutout”, “resample”, and “reduction” (ignoring the trivial case of whole image access).
- Specification of a cutout requires only the ROI, potentially in all measurement axes (POS,SIZE, BAND, TIME, POL)
- Specification of a resample (e.g., changing the sky projection) requires specification of the WCS and possibly image geometry for the output image to be generated. Probably an option should be provided to specify the type of interpolation to be used. Interpolation generalizes fairly easily to more than 2 dimensions, but there may need to be restrictions on rotations involving dissimilar axes.
- A possible feature which has been suggested is use of a convolution kernel, e.g., to interpolate or filter the data during resampling. There is still discussion however of whether we want the interface to specify how interpolation is done to that level of detail; it may be better to leave the detailed interpolation algorithm up to the server code, e.g., to make it easier to re-use legacy code for cube data access.

- By “reduction” we mean reducing the number of samples in an axis. A common case is reducing all data along one axis of a 3-D cube to a single value, thereby producing a 2-D projection of a 3-D cube. Other degrees of reduction are however possible; in general the axis is reduced a factor of N, where some combination of the N samples is performed (this differs from resampling where the scaling is a fractional value). It is probably necessary to specify some standard techniques for how the reduction is performed, i.e., sum, mean/variance, mode, max, min, etc. (we should survey current practice in this area).
- Reduction can also involve filtering of data on an axis. For example if we reduce the spectral (or time) axis of a cube to a single value, we may exclude certain ranges of spectral values using a range list value for BAND. This could be used for example to filter out night sky lines when computing the total flux projection of a spectral cube. A similar operation on the time axis might be used for high energy (X-ray) data, where individual exposures can extend over periods of days.
- Some combination of these access modes are also of interest: e.g., a spatial cutout followed by a 2-reduction on the spectral axis is the cumulative spectrum in a rectangular spatial box.

In the cases above we are limiting the type of operations which can be performed to things which fit within the concept of data “access”, not data analysis. While the distinction is not that well defined, we mean things such as subsetting, filtering, or transforming the data using well defined generic techniques not specific to the type of data being accessed. Actual analysis could be performed in a similar fashion as well, but to keep complexity to a manageable level is out of scope for a data access service (a workflow for example could apply an analysis operation following data access, with both steps being processed on the remote server).

3.5 Multi-Position Access

Multi-position access is a capability desired for all the DAL services to provide a scalable way to process data for many sources. For example the user may have a list of several thousand sources (positions on the sky) for which they want to compute an image cutout (SIA), a spectrum (SSA), or a cone search (TAP). We want to be able to do this in one operation so that a suitable server can use parallel techniques to scale up the computation.

An approach for handling multi-position queries has already been proposed for TAP. In the case of TAP we upload or otherwise reference a table containing positions for multiple sources, as a generalization of the POS parameter to multiple values. Any table could be used so long as it contains positional information with one position per table row.

The same approach could be used for SIAV2 to provide a uniform approach consistent with TAP, e.g., allowing the same source table to be used as input to both services. This is a likely use-case, so we probably want to support this.

In the most general case however, there is no reason (in the case of image access) to limit the computation to only spatial positions. Instead we could upload a table wherein each row specifies all the desired SIA parameters; each row would specify an independent SIA “job” (if a SIA service can do this once it can do it for N cases almost easily as well, plus it is embarrassingly parallel hence easy to scale up). Any parameter which varies for each job would be a table FIELD, and any parameter which is the same for all jobs would be a PARAM (in the VOTable sense). This is an example of a parallel parameter set.

A case can be made to support both of these cases, with the simple spatial position case probably being the highest priority to address first.

Uploading a multi-position table of either sort could be done inline as part of the query, using a POST to submit the query, which would probably want to be run asynchronously as well. VOTable could also be used to provide persistent storage for such a table. Techniques for doing this sort of thing are discussed in more detail in the TAP proposal.

3.6 Grid Capabilities

Addition of “Grid” capabilities is a high priority for SIAV2, much as in the case of TAP (for SSA for example it is not a high priority as dataset generation for 1-D spectra can normally be done easily in real time, but ultimately these capabilities should be available for all the second generation DAL services).

By Grid capabilities we mean the following:

- Asynchronous execution. For SIAV2 this means using queryData to define the data product to be produced, i.e., the “job” to be run, and stageData to submit the job request, referring to the virtual dataset specified by a prior queryData. StageData would return the job ID of the job to be run. Once execution begins the standard UWS facilities would be used to monitor job progress. Multiple images could be produced in a single asynchronous job, as outlined in the previous section on multi-position access. Once execution completes a standard synchronous access-URL based getData could be used to retrieve an image, or staging to a VOSpace could be used, using VOSpace directly to manage the data.
- Authentication. In general this is required whenever asynchronous execution or VOSpace is involved, although there are cases where anonymous access can be provided on a per-client basis, to avoid the need for real authentication. Authentication would use the standard VO facilities for SSO authentication.
- VOSpace integration. In the case of SIAV2, VOSpace could be used either to store generated data, or as a source of input images which the SIA service would be used to access. VOSpace could also be used to upload multi-position tables as outlined in the previous section.

In addition to the above capabilities SIAV2 will require standard VOSI support, including getCapabilties to return service metadata, and getAvailability to allow a client or grid framework to monitor the status of a service. There might also be a facility to query the table columns (as output by queryData) which the a given service instance supports. All of the Grid stuff is very similar for all the DAL services, with minor specializations such as the use of a prior, type-specific queryData to define the “job” to be run.

3.7 Complex Data (Associations)

As was discussed above in the architecture section, the general case of data associations and complex data is left to the generic dataset capability. However, SIAV2 should be able to support a more limited form associations via the query response, much as is already done by SSA. The main restriction is that associations can only be described where all the members are images described by the SIA query response. Associations could be used for example, to describe the images comprising a multiband survey field, or a set of multiresolution images of the same field.

4 Interface

An initial attempt has been made to begin defining the query interface for SIAV2, including both query parameters and query response metadata. Much of this carries over from the generic dataset query and generic dataset metadata, most of which has already been defined for SSA (but is not specific to spectra). Further detailing of this is currently our most active area of work.

4.1 Query Parameters

Briefly, the query parameters required for SIAV2 appear to be as follows:

- ***POS, SIZE, BAND, TIME*** are the same as for SSA or any other data service.
- We probably want to add ***POL*** (or whatever) to provide support for polarization.
- Adding support for ***REGION*** is proposed, to provide a more powerful spatial discovery search capability. This may want to interact with what in SIAV1 was called the INTERSECT parameter. This would be based upon STC region support and would be consistent throughout all the data and footprint services.
- ***FORMAT*** is the same as in SSA and other interfaces.
- Specification of the ***image geometry*** is required for image generation. In SIAV1 this is done with NAXES and NAXIS, which are vector valued parameters. Something similar will probably do for SIAV2.
- Specification of the image ***WCS*** is required for certain cases of image generation, including resampling (spatial reprojection), and cube data access. How to do this is still TBD (see side document), but it should probably be parameter based as for SIAV1, with a clear mapping to both FITS WCS and STC.
- Some new parameters will be required to drive image generation algorithms, e.g., for specifying the type of interpolation or the axis “reduction” algorithm to be used. Both however can be defaulted at the service level.
- ***SPATRES, SPECRP, TIMERES*** - same as for SSA.
- ***TARGETNAME, TARGETCLASS*** - same as for SSA.
- ***SNR***, or some such parameter for specifying the sensitivity or limiting flux of an image, is important to select data for image analysis (SNR as defined for SSA is not really what is needed for image data since there can be many objects in a field; what we actually need is some estimation of the typical or limit flux with respect to noise level).
- ***FLUXCALIB, WAVECALIB, TIMECALIB*** - same as for SSA. There is also ASTCALIB (status of astrometric calibration), which we probably want to add for SIA.
- ***PUBDID, CREATORDID, COLLECTION***, etc. - same as for SSA.

- ***TOP, MAXREC, MTIME, COMPRESS, RUNID*** - same as for SSA.

Queries are no longer required to be positional; any parameters can be used to compose a query.

The biggest issue remaining at this point to specify the SIAV2 query parameters is how to handle the WCS, particularly since we want to support dimensionality as a variable, and hopefully support time and polarization as well as spatial and spectral axes. How to specify the sensitivity or limiting flux also needs more work.

4.2 Query Response Metadata

For any DAL service (except maybe TAP), the query response consists of 1) generic dataset metadata which is common to any data service and which derives from the generic dataset model, and 2) “dataset” metadata, which is specific to the type of dataset, providing a convenient place to describe things which are specific to image data, spectral data, or whatever.

We won’t attempt to detail all the query response metadata here; for the most part this has already been done in the work on data models and for SSA. We will merely look at the so-called “component” data models.

- ***Query metadata*** - same as for SSA.
- ***Association metadata*** - same as for SSA.
- ***Access metadata*** (access reference, estimated dataset size, etc.) is the same as for SSA.
- ***Protocol metadata*** (type of protocol, version) - same as for SSA.
- ***Dataset metadata*.** Since this is whatever is not generic, that is specific to images, this is one of the main things to be detailed for SIAV2. For SIAV1 this included things like the image geometry, scale, and so forth. Now that we have the Characterization model it is not clear how much can be done using generic dataset characterization, and what should be done with the image dataset metadata. In some cases it may be useful to duplicate key image metadata as Dataset Metadata, if only for convenience and to simplify queries. Other useful information (image geometry or pixel type for example) may not be addressed elsewhere).

- ***WCS metadata*.** While not specific to images (spectra for example could also have a WCS), this is a much more important issue for SIAV2 than in any other DAL interface to this point except for SIAV1. As has already been discussed above this is one of the main areas that needs to be detailed. Supporting polarization and time as well as spatial and spectral axes will require extending both FITS WCS and STC, and a clear mapping to both models is needed to support analysis software which uses either representation. FITS WCS probably comes the closest to what is required since it is what is used for spectral data cubes, however stronger support for time is required. STC is the solution for the description of the coordinate system and coordinates on which the mapping is relying.

- ***Dataset identification*** - same as for SSA.
- ***Curation metadata*** - same as for SSA.
- ***Target metadata*** - same as for SSA (although images are less likely to actually have a single target).
- ***Derived metadata*** - this is different for images than what was defined for SSA. Needs further investigation.
- ***CoordSys metadata*** - same as for SSA.
- ***Characterization metadata*** - much the same as for SSA since both use the same Char model, however Char has not been used yet in a serious way for images (so far as I know), so may require some additional work.

In summary, most of the work on dataset metadata connected with SSA carries over to SIAV2. Additional work is required on an image WCS model with clear mappings to both FITS WCS and STC, and on specification of the image specific dataset metadata and the relationship of this to more generic models such as Characterization.

References

[TBA – See mainly the SSA specification, and the associated data models documents such as Characterization and the Spectrum data model. The various GWS specifications for the VOSI interfaces, UWS, etc. are also relevant. The draft TAP specification (aka “TAP/Param” is relevant, especially so far as integrated Grid functionality is concerned).

[1] R. Hanisch, *Resource Metadata for the Virtual Observatory* , http://www.ivoa.net/Documents/latest/RM.html
[2] R. Hanisch, M. Dolensky, M. Leoni, *Document Standards Management: Guidelines and Procedure* , http://www.ivoa.net/Documents/latest/DocStdProc.html