N94- 33798

Role Of Formats In The Life Cycle Of Data

Don Sawyer
Code 633
NASA Goddard Space Flight Center
Greenbelt, Maryland 20771
Phone: (301) 286-2748
Fax: (301) 286-1771
sawyer@nssdca.gsfc.nasa.gov

1.0 Introduction

This paper's perspective is based on the author's experience generating, analyzing, archiving, and distributing data obtained from satellites, and on the experience gained in data modeling and the development of standards for data understanding under the Consultative Committee for Space Data Systems (CCSDS).

Data formats are used to represent all information in digital form, and thus play a major role in all interchanges and access to this information. The need to more efficiently manage and process rapidly growing quantities of data, and to preserve the information contained therein, continue to drive a great interest in data formats.

The purpose of this paper is to examine the role of formats as they support the use of data within a space agency. The life-cycle identified is only one of many variations that would be recognized by those familiar with the 'space business', however it is expected that most of the issues raised will be pertinent to other 'space business' life cycles and to other 'non-space' disciplines as well.

2.0 Space Data Life Cycle Outline

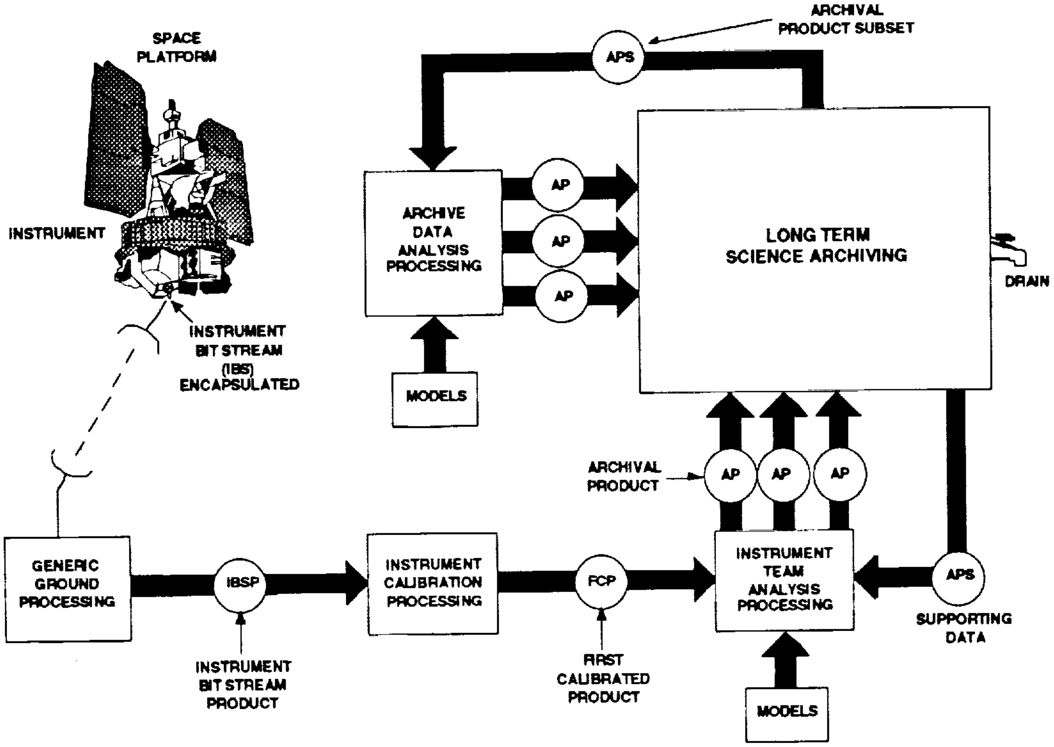
This life-cycle has a clear beginning, but an ill-defined end, if it exists at all. In outline form, as shown in Figure 1, it begins with the generation of an Instrument Bit Stream (IBS) by a science instrument flown on board a space platform, the collection of these bits, and their transmission to a ground system. Ground processing is performed to recover the original IBS and to produce an Instrument Bit Stream Product (IBSP). This is followed by the application of various types of processing to correct for instrument characteristics and to produce a First Calibrated Product (FCP). The further application of various types of processing to compare the data with models and data from other sources usually results in some Archival Products (AP) suitable for long term archiving. The IBSP may also be a suitable product for long term archiving although this is not shown explicitly in the figure. Parts or all of these products are distributed over time spans of decades to data requesters for further analysis and for the archival of additional products derived from this analysis. Finally, at various points in time, there is the expected removal of these archival products from the archive. This demise of data is currently ill-defined because to date most data draining from scientific archives has been accidental. It is an ongoing issue to arrive at effective policies and practices for data retention in such archives.

3.0 Role of Formats

3.1 Information, Not Just Data

The primary purpose for the birth of data in this life cycle is to provide new information that will be used to advance scientific understanding of our universe. (Throughout this paper, "information" is understood as any kind of knowledge that can be transferred among users, while "data" is understood as the representation forms of that information.)

FIGURE 1: A SPACE-DATA LIFE CYCLE


SPACE PLATFORM
INSTRUMENT
INSTRUMENT BIT STREAM (IBS) ENCAPSULATED
ARCHIVAL PRODUCT SUBSET
APS
ARCHIVE DATA ANALYSIS PROCESSING
AP
AP
AP
LONG TERM SCIENCE ARCHIVING
DRAIN
MODELS
ARCHIVAL PRODUCT
AP
AP
AP
GENERIC GROUND PROCESSING
IBSP
INSTRUMENT CALIBRATION PROCESSING
FCP
INSTRUMENT TEAM ANALYSIS PROCESSING
APS
SUPPORTING DATA
INSTRUMENT BIT STREAM PRODUCT
FIRST CALIBRATED PRODUCT
MODELS

Instruments onboard a space platform generate large volumes of data bits having repeating structures containing numbers representing a variety of observations and conditions. As the information represented by these number moves through the life cycle, it is augmented with other information and processed into new types of information. For example, a spinning particle detector on board a spacecraft counts particle events, but eventually this information is turned into count rates and then into particle fluxes. By combining this information with instrument looking direction information, particle fluxes in various directions are obtained. It is such particle fluxes that may be readily compared with our models of this space environment to validate and extend the models, thus increasing our understanding of the universe.

3.2 Formats and Metadata

At each stage in the life cycle, the information is represented in some way by data bits. "Format", or "Data Format" information typically refers to the way data bits are organized into recognizable data types (e.g., integers, reals, characters) and the way sequences of these data types are constructed to form ever more complex structures including whole data products that may cross multiple files on a physical volume. While this format type of information (i.e., metadata) is essential, much more metadata is needed to fully understand the information carried by a digital data stream or data product. For example, data types usually need additional attributes such as meaning (a text description), units, precision, and meaningful ranges or valid values. Information on the relationships among the data types, and the data structures, can be complex but must be known. Additionally, information on the context in which the data were obtained (e.g.,mission, processing history, instrument locations and pointing directions) is also required if the data are to be fully understood.

The amount and types of metadata that need to be recorded and formally associated with a data product, in order to fully understand it, depends on the knowledge of the intended users of the product. Clearly more metadata (supporting information) is going to be needed by high school students than by graduate students in the science discipline associated with the instrument being flown. Further, information that seems 'obvious' to those familiar with the production of a data product can rapidly become 'cloudy' when they have not worked with the product for many months or years. Experience suggests that clear categories of required metadata need to be defined, and then supposedly conforming instances need to be checked by independent reviewers to be sure the information is understandable and complete if the information is to be useable by others.

It is also true that space data products tend to become better understood as they are used over time. Some long term instrument drifts may become apparent and correlations with other data may improve the understanding of the space data product. Further, there may be new sets of metadata created to efficiently search the data in new ways. Thus the mechanisms used to represent and manipulate the metadata must support augmentation of the metadata over time.

Given the large volumes of data that need to be handled in the life cycle, efficient computer processing is a major consideration. Since writing, testing, and maintaining new software is a major expense, one might postulate that an ideal scenario is one in which all (or most) of the types of information, including relationships, that need to be represented should have generally agreed structural representations, or formats. This would mean that a computer interpretable language, capable of representing much of the information desired to be expressed in scientific data products, would be available. The extent to which such a language can be developed, and still provide sufficient storage and processing efficiency, is not clear to this author. That such a language is not available, coupled with the costs of unique software, can be viewed as the primary reasons for the great interest in data formats.

3.3 Access to Formatted Data

As there is no standard, formal, language for representing the kinds of scientific information we have been addressing, a number of techniques, each supporting a type of access to the information, are currently being used.

The most basic type of access is to build an understanding of a unique data structure, representing some set of scientific information, into the access software. This has the advantage of being very efficient for access, but has the severe disadvantage of being very costly since it promotes the generation of lots of unique software. It also makes information interchange difficult because recipients need to make modifications to their local software to be able to 'read and understand' each new data structure. For long term preservation of such data, good human understandable documentation going down to the bit level is needed to enable new access software to be written when needed. The existing software languages are inadequate for this task because it can be very hard to infer the underlying data structure from the software. This is understandable because these languages are designed primarily to transform data and not to provide an understanding of the data. In addition, the software languages usually do not address the needed bit level information.

Another type of access is to use software that supports a 'particular data model' (e.g., an n-dimensional array with dimensional and global attributes) having a private internal data structure. Information to be exchanged or stored is mapped to the data model, and then loaded into the internal data structure. This includes the data and some of the metadata needed to understand the data. The advantage for information interchange is that local software can be prepared to work with the data model, and thus be able to work with a variety of information as long as it can be usefully represented by the data model. The disadvantage is that no current data model can usefully represent all the types of scientific information that need to be exchanged, and the information's representation must be converted to the internal form of the data model. Further, for long term preservation, the information must be carefully checked to ensure no loss when changes occur to the local hardware, operating system, or version of the data model access software. It is not clear that this can even be accomplished for large data volumes. When the data model and its software is no longer supported, the information will need to be extracted and moved to a new model or a new technique for information preservation.

A third type of access is provided by a variation of the second type of access. In this type the data model is represented by a standard, not private, internal data structure. This has an additional advantage for long term preservation in that the information content's dependence on hardware and operating system should be clear from the standard, and thus much more easily controlled against information loss when hardware and operating systems change. Further, there is no need to move the information to a new model or mechanism when software supporting the model is no longer to be maintained. The creation of new access software or other techniques can wait until the information needs to be used as long as the document representing the standard still exists. In other words, the lack of working access software does not mean the information has been lost.

A fourth type of access is provided by software that understands a standard data description capability that is embedded with the data. This differs from the 'particular data model' in that it is able to support a much more varied set of data structures, but typically (as far as the author is aware) does not support the relationships that would provide the capabilities of the 'particular data models'. The lack of 'particular data model' support is likely to be addressed as these description languages mature. The advantage to this approach is the great flexibility of data structures that can be supported. The disadvantage is that embedding the description capability with the data may cause considerable data expansion and thus may not be practical for large data volumes. Further, access to the information may be less efficient than for 'particular data models' with data structures tuned to their needs. However data description capabilities tare especially good for information preservation since the information is preserved as long as the data description standard is available. It should be noted that the 'particular data models' use some type of internal data description capability, but only to the extent needed to support their data model.

The last type of access described here uses software that understands a separable data description language. This is like the embedded capability described above, except the description may be separated from the data. This has the distinct advantage of not expanding the data volume, and of allowing this metadata to be independently managed and updated. This also allows the structure of the data to be efficient for representing the information. The disadvantage is that access to the data, using the description language software, is likely to be less efficient than for the 'particular data model' case.

Some of the current constraints on the use of data formats and their access mechanisms can be seen by a closer examination of the format and metadata issues in the space data life cycle.

3.4 Format (and Metadata) Issues in the Life Cycle

The approach in this section is to examine the environments suggested by Figure 1, and to determine a number of data format considerations that can affect the identified data products

3.4.1 Space Platform Environment

An instrument on board a space platform generates an Instrument Bit Stream (IBS). There may be several constraints that determine how the data are formatted, including: 1) space platform resource limitations such as telemetry bandwidth, on board power and weight, 2) available space platform data handling services, and 3) reuse of data structures from previous versions of the instrument or from similar missions.

This resource constrained environment drives a very efficient use of bits to represent numbers, flags, modes, and other conditions. Seldom are these numbers 16 or 32 bits in length as they usually are when generated by ground based computers, and often they are also scaled in various ways. Complex instruments will use mode indicators to signal the presence of different data structures, or different interpretations of the numbers in a given data structure. To reduce the telemetry burden, only information not easily added on the ground will be included in the IBS. Such data structures will need a lot of additional metadata, not found in the IBS, to be fully understandable.

The IBS is encapsulated in some manner before it is transmitted to the ground. Traditional space platform major and and minor telemetry frames force an overall structure that each instrument must share, with the result that one instrument's data stream is multiplexed with that of others unless there is only one primary instrument on the platform. The new standards for the Consultative Committee for Space Data Systems (CCSDS) packets and frames allows an instrument to own individual packets containing hundreds or thousands of bits. This greatly simplifies space platform data handling on the ground and gives the instrument developer much greater freedom in designing a data format for the packet content. However the instrument packet designer must now explicitly insert time tags as needed because, in general, there is no guaranteed relation between packet generation and an external clock.

If similar instruments have been flown on previous missions by the same instrument team, then it is likely that the same or similar data formats for the IBS will be used and this will facilitate some software reuse.

In summary, an IBS tends to have a great variety of data type representations for numbers, and to be quite instrument specific in the organization and meaning assigned to these data types. Nevertheless the types of information represented, such as images, time series, and spectra, are more common across different instruments and missions than are the various representations or formats used. The use of data description languages may be the best approach to providing common access (i.e.,reusable software) while supporting a variety of bit efficient representations.

3.4.2 Generic Ground Processing

The function of Generic Ground Processing in this life cycle is to remove the artifacts of the space to ground transmission domain, to recover the original IBS, and to put out an IBSP which has a basic structure that is the same from mission to mission. Typically the IBS is collected for a previously agreed period, such as an orbit or a day, before an IBSP is released. The IBSP will include, in addition to the original IBS bits, attributes related to the accumulation period such as orbit number or time period covered, and possibly some quality information relating to the reliability of the recovered IBS. This information will be appended without altering the format of the IBS, which in general is transparent to Generic Ground Processing. This stage of the life cycle is a reasonable place to add one or more identifiers of the metadata needed to convert the IBSP into useful information. This has the benefit of stimulating the documentation of this metadata (which might include a formal description of the format using a data description language), and making the IBSP much more archivable as well as useful if it needs to be shared with a distributed set of colleagues. This will also provide a good start to the metadata that will be needed to support other products in the life cycle, and will provide a source of information to stimulate reuse in other mission's products. Note that it appears less desirable to actually include most of the metadata, as opposed to identifiers of the metadata, in the IBSP. Including this metadata may significantly expand the size of the product, and it freezes the metadata at an early stage of understanding of the product.

An IBSP instance may be one or a few files, and may be distributed via networks or via physical media (sequential or random access).

This environment may include a temporary, or semi-permanent, archive for the IBSP instances from many instruments and missions. It is assumed that any such temporary archiving is done without needing to understand the content of each IBS. For example, the archive catalog would be populated with information such as spacecraft identifications and orbit numbers that are obtained from sources other than the IBS. Considerations for permanent archiving are addressed in sections 3.4.4, 3.4.5, and 3.4.6.

In summary, the format of the IBSP includes that of the IBS, but adds additional attributes associated with the collection interval, mission, instrument, etc. to make the resulting product readily recognizable and archivable without having to parse the content of the IBS structure. The format should be efficiently accessible whether the distribution is via networks or physical media (both sequential and random access) since the next stage of processing will most likely be done in a pipeline approach and a media independent format will have the best chance of being a long lived output format from this Generic Ground Processing environment.

3.4.3 Instrument Calibration Processing

The functionality envisioned in this stage of the life cycle is primarily the conversion of the raw IBSP numbers (actually the IBS numbers) to more meaningful quantities. These conversions are most likely to be reversible (e.g., multiplying values by a constant), although some non-reversible calibrations may also be performed at this stage. Although not shown explicitly in Figure 1, this processing may require incorporation of other data streams derived from the space platform, such as orbit and attitude information, or data from ground observations, to complete the calibrations. Typically the result is an Initial Calibrated Product (ICP) whose format is organizationally similar to the IBSP, but with some information (perhaps quality) eliminated and other information (such as location and pointing direction) added.

This product may, or may not, have most of its values converted to 8, 16, or 32 bit quantities to be more easily processible by software. Even greater changes in the format are likely to take place if there has been prior agreements to push all data into a particular data model whose implementation software maintains its own internal format, or if there is a need to conform to input requirements for the next stage of processing. These decisions will most likely depend on trade-offs among the volume expansion that would take place, the availability of storage, and use of common mechanisms for access to this data product.

This processing would most likely be done at a mission or project facility, and typically an ICP instance would be one or a few files.

The metadata associated with the IBSP should be a good starting point for the metadata needed for this new product, but it needs to be suitably updated. It is very important that the insight gained from overseeing the calibration processing be recorded as supporting metadata. A new metadata identifier, for this new set of metadata, can be inserted in forming the ICP. Again it appears desirable not to include substantial amounts of metadata directly within the product for the same reasons given in section 3.4.2. The use of an overall format organization that is efficiently accessible from all types of media or from networks is also desirable if the mission is long lived and evolution of systems is a concern.

In summary, the format of the ICP may, or may not, be substantially different from that of the IBSP. This appears to depend primarily on the nature of the calibration performed, on the relative availability of storage space for this product, and on the planned use of access mechanisms such as use of a single data model or use of a data description language. As envisioned in this life cycle, the basic nature of the information contained in the ICP is not substantially different from that in the IBSP or IBS. In other words, the presence of an image, time series, or spectrum, for example, would be present throughout and through a simple mapping (calibration function) be related to the output of the instrument. More complex transformations of the information are assumed to take place in the next stage of the life cycle.

3.4.4 Instrument Team Analysis Processing

The two primary objectives of this stage of the life cycle are to extract new understanding of our environment from the data, and to produce useful APs that support future analysis. The functions envisioned to meet the objectives are wide ranging. They include reprocessing the IBSP with improved calibration information, the application of physics models to effect substantial transformations of the ICP into what are often referred to as "higher level products", and the selective subsetting of the products for incorporation into a variety of favorite data analysis and display tools. Subsets of data (Archival Product Subsets, or APS) from external archives may need to be folded into this processing either to create a new product or for comparisons. Papers for publication in the literature would be generated, and some of the products (including possibly the IBSP and FCP) generated should be suitable for preparation for long term archiving.

This processing environment will need to have its own archive to hold the ICP instances, support products ingested from long term archives, and intermediate products generated during the processing. This archive requirement is not explicitly shown in Figure 1, but may be met by some combination of project archive support and local archive support. In general, it must be assumed that data products in multiple formats will need to be archivable and accessible to this analysis processing environment. The formats of these products can ease the archiving function if they include an easily accessible set of attributes (e.g., time period covered, mission, instrument) that can be used to populate an archival catalog, and if they did so in a way that allowed them to be updated and accessed without having to modify or parse the rest of the data product.

The analysis processing environment's desires for the formats of the input products (ICP and APS) shown in Figure 1 can be widely varying. While this environment may be able to significantly affect the ICP format, such as ensuring that arrays are stored in an efficient way for the local hardware, this is much less likely for data acquired from long term archives. Thus this environment must work with data in multiple formats.

Given that this environment is likely to have a good set of resources, the detailed formats at the record level (bits and bytes) are less of a concern than overall product organization and a clear understanding of the information present in the data. Maintaining an up-to-date set of metadata linked to the ICP will help to avoid information loss, particularly when this stage of processing involves sending data to distributed colleagues, and it will aid in preparing data products for subsequent archiving. Processing the ICP and APS will generally involve applying selection criteria to values within these products, and the extraction of subsets of values when the selection criteria are satisfied. The focus will be on software to perform this subsetting and extraction. The output formats for these extractions may be driven by the input formats required by favorite analysis tools.

The generation of Archival Products (AP) will be constrained by the requirements of the intended archive, as well as by the internal formats used in this analysis environment. The archive should require formats that are as media independent as possible in order to facilitate management of the products within the archive. The archive may require a specific set of attributes with each instance of an AP to support automated data ingest cataloging and future subsetting. The extent of metadata, needed to support use by some minimally trained potential user (e.g., high school student, graduate student) and going down to the bit level, needs to be included with the AP.

A typical AP instance, submitted to an archive, may range up to tens of thousands of files. A great many such instances many be sent over several years to complete the archiving of a single AP.

In summary, the information processing within this stage is likely to deal with a number of formats. Constraints on these formats typically come from the input constraints of commonly used data analysis tools, and from archive constraints on the types of formats the archives will provide. Further constraints may come from satisfying local data management needs by providing a set of attributes with each product instance to ease local cataloging. Finally, for those products which are intended for long term archives, additional constraints are likely to be imposed by particular archives, including the association of complete sets of metadata with each product instance. The impacts of these constraints can be minimized if formats are adopted which support the archive's data ingest and metadata needs since these formats should also support local data management and cataloging needs and aid in the distribution of meaningful products to colleagues.

3.4.5 Long Term Science Archiving

The two primary objectives of this stage of the life cycle are to preserve information (not just data bits) for an indefinite period (assumed to be many decades, at least), and to provide requesters with a range of access services. The role of formats is a key element in both of these objectives.

The information preservation objective has proven to be quite difficult and tends to be greatly underestimated by new archives that have not felt the full impacts of technological change. For APs ingested into the archive, full data product metadata, down to the bit level, is still needed. Software access, alone, as the way to understand and work with bit structures has proven to be inadequate. It is very expensive to ensure that software, archived with a data product, performs properly against changing hardware and operating systems. Even software which supports multiple data products is unlikely to have sufficient resources behind it to permit the extent of testing needed to be very sure that all the data products are accessible without information loss. Large, stable vendors stand the best chance of having the resources to do extensive testing, but even here there is no guarantee against some data/information loss. An archive relying on such software must also have an extensive test plan involving accessing and comparing data values with the new and old software. Software which provides many types of information is particularly difficult to adequately test and this becomes more difficult as the archive data volume grows. In addition, software (i.e., data manipulation languages) are inadequate as data description languages because they rely on local representations of bit level data types which tend to change with new hardware and operating systems. Therefore it appears that the use of standard data description languages, coupled with human readable descriptions intended to be complete and understandable 50 years into the future, is the best current approach to addressing these aspects of information preservation.

Archives which do format conversions on AP ingest to an internal data model also risk information loss unless they have done a very careful mapping of the incoming information to that model. This can be difficult since archive personnel may not sufficiently understand the incoming information to ensure against information loss. Therefore it is safest to avoid format conversions for archival copies of APs, and to limit conversions to the provision of special data access services for some APSs. The provision of some of these services, such as rapid online access to large amounts of this information, is likely to require some format conversions given current technology. This implies a trade-off must be made among information preservation , efficient online access services to the information , and storage volume. The author believes that only two of these three can be optimized in any single system, at least with current technology. This appears t o be a major challenge for archives that is easily overlooked since the lack of desirable access services is felt immediately while the loss of information may not be noticed until there has been major technology evolution (e.g., a decade or more).

AP formats should also support the updating of metadata over time without having to rewrite the associated data. For example, new calibration coefficients may be defined, new interpretations of certain types of observations may need to be documented, and metadata errors may need to be corrected.

AP formats, including the associated metadata and its linkage to the data, should avoid or at least isolate, any media dependence. For example, the embedding of directory and file names in the data or metadata can produce name conflicts when the information must be moved, in response to technology evolution, to new media types within the archive. It also makes subsetting of the information (i.e. creating an APS) for distribution to requesters difficult to accomplish. Since all references to directory and file names can not be eliminated, the best approach appears to be to use formats which allow these names to be isolated and readily updated as needed. Directory names and file names should never be used in metadata text as it become very hard to update. Unfortunately this is common practice among data producers because it is convenient in a local system and there is no standard way accepted to name, and thus refer, to other data objects.

A difficulty in producing an APS with proper metadata results from having to extract that set of metadata, from the total metadata associated with the original AP, that is pertinent to understanding the APS. This puts constraints on the format of the metadata and suggests that the metadata should be broken into separate objects, some of which will apply to all possible APSs and some that will be need to be shaped for particular APSs.

The efficient ingest and cataloging of APs, which is needed to support access to these data as APSs, requires that catalog attribute objects accompany the data products and be linked to this data at a useful level of granularity. The data producers, after they understand the ingest requirement of archives, are in the best position to prepare products that include these attribute objects. Ideally, all data products would include such objects to facilitate local cataloging in both temporary and long term archives.

APs which are submitted to an archive on media volumes containing a great many data objects also need to include standard table-of-contents and/or index objects. The purpose of these objects is to permit efficient subsetting of the AP into an APS in response to requests. Again, it is much easier for the data producer to create these objects in consultation with the archive than it is for the archive to produce them after a great volume of data has been received.

In summary, archives should require AP formats to be as media independent as possible and to include complete metadata, down to the bit level, in order to maximize information preservation. This metadata should be updateable without having to rewrite the data. The inclusion of catalog attribute objects, including table-of-contents and index objects, can greatly improve archive efficiency and access services at little cost to AP producers. The provision of efficient online access to large amounts of information may require format conversions and special software, with the attendant increase in data storage volume over that devoted to information preservation. The use of data description languages can support information preservation while allowing access to a great variety of data structures, but current data description languages are not yet providing very efficient access.

3.4.6 Archive Data Analysis Processing

There are two primary objectives for this stage and they are the same as for Instrument Team Analysis Processing; to extract new understanding of our environment from the data, and to produce useful APs that support future analysis. The primary differences are that the data come from an archive as one or more APSs, and the available processing resources may be much less than for Instrument Team Analysis Processing.

An APS needs to have a complete metadata set associated with it as it may be decades since the information was put into the archive and there may not be anyone who is familiar with the data or even the mission. APS formats need to support incremental access to the information, such as through table-of-contents and index objects, when the volume of data in the APS is large (e.g., CD-ROM).

It is a great benefit to this analysis processing if there exists working access software for a given APS. An APS that includes a format description written in a standard data description language stands a good chance of having working software that may be used to access the data. An APS that conforms to a particular data model may also have working software, but this software should be supplied as an addition to the metadata, not as a substitute.

4.0 Summary

This space data life cycle is only one of many variations seen in the 'space business'. However it is expected that most of the issues and concerns raised will also be applicable to 'non-space business' life cycles.

The preservation of information (not just bits) throughout this cycle is a primary objective, and this requires appropriate metadata, down to the bit level, at each stage of the cycle. Software alone is not suitable for information preservation. The required metadata grows throughout this cycle, and must be associated with the data in ways which permit both the data and metadata to move easily to new types of media, including both random and sequential.

Data formats are used to represent the data and the metadata, and to link the two. The data formats are subject to various constraints as the information moves through the life cycle, and no single bit representations for science objects (e.g., image, time series) is practical at all stages of the cycle. The need to support subsetting of both the data and metadata is apparent in several of the stages, as is the need to support archival or repository ingest.

It is suggested that data description languages may be a good approach to supporting information preservation and some automated access, but are not yet up to providing efficient access for a range of archive online services. Therefore it may be necessary, given current technology, to use one copy of the information for preservation and a somewhat differently formatted version (in some cases) for efficient online access services. However the products sent out from these access services may be well served to have associated standard data description language metadata to support automated access. The use of particular data models have a role to play, particularly in terms of efficient access, but they can get in the way of information preservation if archives try to use them as their primary storage mechanism.