

STANDARD FORMATTED DATA UNITS

FRANK J. LOPINTO

*Computer Sciences Corporation, 10000A Aerospace Road
Lanham-Seabrook, Maryland, USA*

DONALD M. SAWYER

*NASA/Goddard Space Flight Center
Greenbelt, Maryland, USA*

Standard Formatted Data Units (SFDUs) were developed consensus among the worlds space agencies. They are intended to facilitate the exchange and archiving of space science data. The result of the effort is a packaging standard that is well suited for packaging general data objects. In addition to the packaging standards, we have or are developing standards for describing the syntax of data, the semantics of data, for registering descriptions in permanent databases maintained by the individual space agencies (Control Authorities), and for relating these descriptions to data exchanged among heterogeneous systems.

1 Purpose Of This Paper

The purpose of this paper is to introduce the work of the Consultative Committee on Space Data Systems (CCSDS) Panel 2 to the High Energy Physics community. We describe the type of work performed by Panel 2 of the Committee. In particular, we focus on the Standard Formatted Data Unit Recommendation which forms the basis for all of the other related standards produced by the Panel.

We discuss our accomplishments as well as our current activities. It is hoped that we can expand the dialogue between the Earth and Space Sciences community and the High Energy Physics community by discussing data management issues that are common across the disciplines.

2 The Consultative Committee On Space Data Systems

The Consultative Committee on Space Data Systems (CCSDS) was organized in 1982 to address "space data" interchange issues. Membership in CCSDS is limited to space agencies. Individuals who attend CCSDS meetings represent their respective agencies. These individuals can be from government, industry, or academia but they must invited to participate by a space agency and must represent their agencies during meetings. The member agencies include: CNES/France, CSA/Canada, BNSC/UK, DLR/Germany, ESA/Europe, INPE/Brazil, NASA/USA, NASDA/Japan.

CCSDS is organized as three panels, each of which may contain sub-panels. The panels are designated P1, P2, and P3. P1 deals with Telemetry, Tracking and Command issues. Its major accomplishment is the development of protocols for communications between space-based and ground-based systems.

P2 deals with Standard Data Interchange Structures issues as discussed below. P3 deals with Cross Support Operations which are basically management standards and procedures that facilitate interpretability among systems from different agencies.

3 Basic Terminology

In our concept, “space data” refers to scientific and engineering data. These data can be acquired directly from spacecraft observations, or they can be ground observations of space phenomena. In other words, space data includes data about phenomena that occur in space (regardless of where the phenomena are observed). However, “space data” can also include data about conditions on the ground insofar as they relate to or, specifically, calibrate spaceborne instruments.

Ground truth measurements are used to check results obtained from space-based instruments. For example, a remote earth sensing instrument may be designed to distinguish between snow, sand, forest, asphalt, and plowed earth. But to check the results of a satellite observation of the ground, it is necessary to go to the spot on earth observed by the satellite and see what actually is there. In this way, one can gain confidence that our systems are working properly and can accurately describe the earth’s surface. This definition of “space data” is quite resulting in standards that tend to be quite general.

The term “metadata” has important meaning for us. It is defined to be “data about data”. Relating data and metadata is a key concept in our work. Intuitively, data and metadata must be related to yield understanding. A image recorded in space is probably useless for scientific purposes unless we know where the recording device was pointed and when the image was taken. Intuitively, the photograph is the data and the location and time information is the metadata. Alternatively, the photograph is the data and the format of the records used to describe the image is the metadata.

Notice the two distinct alternatives presented in the previous paragraph. The first example of metadata is about a particular instance of data. In other words it is about a particular photograph. The second instance is about the structure of the photograph, i.e., what kind of process is needed to develop and view the data. The information about this structure can be as technical as desired but it cannot know anything about what was photographed. Our standard formally distinguished between instance information and data type information.

4 Basic Assumptions

We assume that data must be archived for long time periods. This is particularly obvious for earth observations intended to monitor global change. In a sense, the value of the data actually increases with time. It is assumed that the data will be useful long after the hardware and software used for acquisition becomes obsolete. One might say that we can simply copy data from old systems to new ones and thus eliminate the problems of obsolete systems. However, the cost of doing so is likely to be prohibitive.

The community of users who are interested in the data is quite diverse. While high energy physics data is interesting only to high energy physicists space data, and especially earth observation data, is interesting to a diverse community. This includes: scientists from many disciplines, commercial shipping, fire fighters, space mission operators, and the military. It is unlikely that they will share a common terminology.

5 Current Situation

Though there are exceptions, to varying degrees, the current state of affairs regarding descriptions of data is as follows. Either documentation doesn't exist or, if it does, it is of poor quality. Most often, software substitutes for documentation (i.e., "If you really want to know how it is, look at the code.") This is fine if you have access to the code or other documentation but often you do not unless the author is a close colleague. In any event, if you don't understand the programming language (not everyone uses FORTRAN) or are not good at it, then you are out of luck. So what does exist is not readily available or understandable.

This last remark is becoming less true now that we have the World Wide Web which has made heretofore private archives on desktop disks available to the public. However, merely placing a document on the Web does not ensure that anyone will read it. The document has to be found before it can be read and, unless it is discovered by chance, its existence has to be announced before anyone will search for it. Alternatively, there can be many copies of a document replicated across the Web. It may be difficult to determine the currency and authenticity in many cases.

Therefore, everyone tends to invent their own data formats and then develop their own software to handle their formats. It is easy, and somewhat automatic, for a programmer to conclude that what he needs doesn't exist and therefore a program has to be written. Programmers are not trained to use other people's code. The money spent reinventing formats and software could be used more productively.

6 Ideal Data Interchange System

The ideal data interchange system would first allow a user to determine the existence of data. It should not be assumed that the user knows how to ask (this is particularly important for multi-disciplinary work.) The user should be able to find out where the data is and be able to retrieve the data without learning specialized query procedures.

Users should be able to generate new data either by combining data sets, by data reduction, or by recording new measurements. The ideal system should provide easy-to-use templates into which data can be written. Software to perform transformations and data reduction should be available and it should be easy to tell what software goes with what templates.

The templates themselves should help the user understand the data. The fact the a piece of data is in a particular spot in a template should add value to the data. In other words, the template (or packaging) should enhance the amount of information about the data. Finally, the ideal system should support the use of data. It should be easy to run graphical viewers and other applications programs that help the user gain

knowledge. Therefore, the ideal data interchange system should be designed to be compatible with and supportive of the tools that end users want to use.

7 Panel 2 Standards

CCSDS Panel 2 develops standards to facilitate data interchange. That is not to say that our current recommendations, if implemented completely, would result in the ideal system described above. But they are a decent start. There are three standards which have been approved by the formal process of reaching consensus among participating agencies. The panel members obtain input from their respective national agencies during the standards development process. The consensus that they reach reflects the needs of various constituencies within their home countries. There is no voting and all agencies are equals.

At present, P2 has published three Recommendations: The Parameter Value Language (PVL), Control Authority Registration Procedures, and the SFDU Structures and Construction Rules. PVL is a simple recommendation that can be used to transfer lists of parameters. In addition to defining single parameters and associating values and units with them, the syntax allows the definition of sets and sequences of parameters. The other published recommendations are discussed in the next section.

The panel is working on recommendations for Data Description Languages (DDL), Data Entity Descriptions, and Filenaming Conventions. Data Descriptions Languages will allow users to read media and reconstruct basic data types (i.e., integers, floating point, etc.) as well as user-defined data types. It is not our intention to invent new conventions. Rather, we want to recommend how software, presented with a long string of bits, can recognize the bits as “higher level” objects. In other words, the Data Description Languages deal with the syntax of the bit stream.

The Data Entity Dictionary (DED) deals with the semantics of the bit stream. In other words, what do the integers, floats, etc. mean? Is it a sequence of temperature readings of a list of particle energies? Clearly there is a relationship between the functionality provided by a DDL and the functionality provided by a DED. It is not completely obvious where to draw the line between the two (at least based on the discussions we have had) but all agencies agree that there is a line to be drawn and that the issues can be separated into two categories.

8 Packaging and Linking Concept

The SFDU recommendation as discussed above is a packaging standard. It associates data with metadata. Perhaps a moment’s reflection will show the distinction between data and metadata is a human bias. From a formal perspective, the packaging simply relates one data object to another. However, it will be useful to continue to think of data and metadata as separate things. The following might help reinforce this human bias.

We could have a restaurant menu written in several languages (French, Italian, etc.) A label on the menu could have the name of the language in which the menu is written. The label is the metadata and the menu is the data. Suppose we want to read the menu but can’t speak the language. The metadata would tell us how to read

the menu, i.e., “find someone who speaks French”). Therefore, our process would be: 1) read the metadata, 2) find a system (person) that understands the language in which the menu is written, 3) get the person to read the menu and tell you what it says.

One might wonder how one reads the label on the menu. After all, the label might be written in a language that we don’t understand (i.e., Chinese). In other words, if we have a menu with a label that says French in Chinese, we will not know that we have to find a French speaker to read the menu for us. This problem is recursive in that we need to know to find a Chinese speaker to tell us to find a French speaker to read the menu for us (if we still have an appetite). Somewhere it has to end. At some point there has to be a label written in a language that we understand.

Returning to SFDUs, we say that each object contains a label and a value. The label points to a data description which we can retrieve from a Control Authority (an archive of descriptions managed according to CCSDS rules). The data description will be an object with a label. The label will contain a pointer to the description of the description. We can retrieve the description of the description from either the same Control Authority or from another Control Authority (they are distributed in space agencies around the world). Then we will have the description of the description of the description. This chain will end when the description points to the SFDU Standards and Construction Rules Document (also known as ISO 12175). At that point it is assumed that we can understand the description ourselves. In practice it means that we have software that understands the (handful) of data structures specified in the standard.

9 Conclusion

The Standard Formatted Data Unit concept was developed to facilitate the interchange and long term archiving of space data. However it is simple and therefore general enough to be applied more broadly to many kinds of science and non-science data. Fundamentally, the standard allows us to organize and relate data in any format. It places no restrictions on the data or on the data descriptions. It requires only that data be associated with a label that can be used to form the association. Use of this standard, and the other standards developed by CCSDS Panel 2, can help reduce to cost of data management by facilitating the use of common software for creating, processing, and transferring collections of data objects.