Table of Contents

NOTE: if the text is italicized, its essentially notes/thoughts for a section, and shouldn't be taken particularly seriously. Feel free to comment if you like on whichever parts of the document interest you (including notes).

Why FITS is failing to meet the needs of modern astronomical research

Draft - 38 {Sep 27 -2013}

Brian Thomas[1], Frossie Economou[1], Perry Greenfield[2], Paul Hirst[3], Tim Jenness[4], David S. Berry[5], Erik Bray[2], Norman Gray[6], Demitri Muna[7], James Turner[8], Miguel de Val-Borro[9], Juande Santander-Vela[10], David Shupe[11], John Good[11] and G. Bruce Berriman[11]

Abstract

The Flexible Image Transport System (FITS) standard has been a great boon to astronomy, allowing observatories, scientists and the public to exchange astronomical information easily. The FITS standard, however, is showing its age. Developed in the late 1970s, the FITS authors made a number of implementation choices that, while common at the time, are now seen to limit its utility with modern data. The authors of the FITS standard could not anticipate the challenges which we are facing today in astronomical computing. Difficulties we now face include, but are not limited to, having to address the need to handle an expanded range of specialized data product types (data models), being more conducive to the networked exchange and storage of data, handling very large datasets, and the need to capture significantly more complex metadata and data relationships.

There are members of the community today who find some (or all) of these limitations unworkable, and have decided to move ahead with storing data in other formats. This reaction should be taken as a wakeup call to the astronomy community to make changes in the FITS standard or to see its usage fall. In this paper we detail some selected important problems which exist within the FITS standard today. It is not our intention here to prescribe specific remedies to these issues; rather, we hope to call attention of the FITS and greater astronomical computing communities to these issues in the hopes that it will spur action to address them.

[1] NOAO Science Data Management, 950 N Cherry Ave, Tucson, AZ 85719, USA
[2] Space Telescope Science Institute, 3700 San Martin Drive, Baltimore, MD 21218, USA
[3] Gemini Observatory, 670 N. A'ohoku Place, Hilo HI, 96720, USA
[4] Department of Astronomy, Cornell University, Ithaca, NY 14853, USA
[5] Joint Astronomy Centre, 660 N. A'ohoku Place, Hilo, HI 96720, USA
[6] School of Physics & Astronomy, University of Glasgow, Glasgow, G12 8QQ, United Kingdom
[7] Department of Astronomy, The Ohio State University, Columbus, OH 43210, USA
[8] Gemini Observatory, Casilla 603, La Serena, Chile
[9] Department of Astrophysical Sciences, Princeton University, Princeton, NJ 08544, USA
[10] Instituto de Astrofísica de Andalucía, Glorieta de la Astronomía s/n, E-18008, Granada, Spain
[11] Infrared Processing and Analysis Center, Caltech, Pasadena, CA 91125, USA

1. Introduction

The Flexible Image Transport System standard (FITS, Wells *et. al.* 1981; and more recently, the definition of the version 3.0 FITS standard by Pence *et. al.* 2010) has been a fundamental part of astronomical computing for a significant part of the past four decades. The FITS format became the central means to store and exchange astronomical data, and because of hard work by the FITS community it has become a relatively easy exercise for application writers, archivists, and end user scientists to interchange data and work productively on many computational astronomy problems. The success of FITS is such that it has even spread to other domains such as medical imaging and digitizing manuscripts in the Vatican Library (Rasband 2010; Allegrezza 2012).

Although there have been some significant changes, the FITS standard has evolved very slowly since its genesis in the late 1970s. New types of metadata conventions such as World Coordinate System (WCS, Greisen & Calabretta 2002, Calabretta & Greisen 2002, Greisen *et al* 2006) representation and data serializations such as variable length binary tables (Cotton, Tody & Pence 1995) have been added. Nevertheless, these changes have not been sufficient to match the greater evolution in astronomical research over the same period of time.

Astronomical research now goes beyond the paradigm of a set of observational data being analyzed only by the scientific team who proposed or collected it. The community routinely combines original observations, theoretical calculations, observations from others, and data from archives on the internet in order to obtain better and wider ranging scientific results. Many research projects now involve many diverse data sets from a range of sources. Instruments in astronomy now produce many orders of magnitude larger datasets than were common at the time FITS was born, in some cases requiring parallelized, distributed storage systems to provide adequate data rates (Alexov et al. 2012) .

Astronomers have increasingly come to rely on others to write software to help process and analyze their data. Common libraries, analysis environments, pipeline processed data, applications and services provided by third parties form a crucial foundation for many astronomers toolboxes. All of this requires that the interchange of data between different tools needs to be as automated as possible, and that complex data models and metadata used in processing are maintained and understood

through the interchange.

These changes in research practices pose new challenges for the 21st century. We must address the need to handle an expanded range of specialized data product types and models, be more conducive to the distributed exchange and storage of data, handle very large datasets and provide a means to capture significantly more complex metadata and data relationships.

Already some of these limitations have caused members of the community to seek more capable storage formats, both in the past (Starlink HDS, Disney & Wallace 1982; XDF, Shaya et al. 2000 and FITSML, Thomas et al. 2000 and Thomas, Shaya and Chueng 2001), and in the present and future (Anderson et al. 2011). There is a danger that the use of FITS will fall should it not adapt to these new challenges.

It is our intended goal in this paper to highlight some selected, important, problems which exist in the FITS core standard today. It is not our intention to prescribe specific remedies to these issues; rather, we hope to call the attention of the FITS and greater astronomical computing communities to these issues in the hopes that it will spur action to resolve them.

2. Deficiencies of FITS for Modern Astronomical Research

As technologies and research techniques in astronomy have evolved FITS has not kept pace. As a result, gaps between FITS utility and the needs of research have opened up and widened over time. In this part of the paper, we detail many issues, or deficiencies of FITS, which we see in this regard.

The deficiencies of FITS appear to fall into one of several groups. These may be summarized as poor support for information interchange, missing critical data models, inflexibility in representing both metadata and data and poor support for large and/or distributed data. We address each of these in turn below.

2.1. *Poor support for information interchange*

FITS originated as a delivery format for observatory data. It was the format of choice when transporting data between different data reduction environments such as IRAF, Starlink, AIPS and MIDAS.

However, requirements for interchange have evolved. Effective interchange, as we shall illustrate, now includes things like the ability to declare of models for use in higher level processing, validation of models within the file and, at the most basic level, the ability to declare which version of the FITS serialization is being used.

It's not as if we have not needed these things until now. In fact, parts of the astronomical community have designed data formats to satisfy some of these requirements. The ADC's XDF, LOFAR's HDF5 data model (Alexov et al 2012), and Starlink's NDF (Currie 1988; Warren-Smith & Wallace 1993 and see ADASS Poster P91) all serve as examples in this regard. XDF was created primarily to support the development of FITSML an XML version of the FITS data model which could utilize XML schema for validation. NDF was developed in the late 1980s as a means of organizing the hierarchical structures that were available via the Starlink HDS format when it became apparent that arbitrary hierarchies could lead to chaos and lack of ability for applications to interoperate.

2.1.1. Lack of serialization/format versioning

The designers and maintainers of FITS have espoused that a file is "once FITS, forever FITS"[12]. Certainly some in the community see this as a strength for the format as it appears to promote long term stability and archivability of FITS data (Allegreza 2012, US Library of Congress[13]). The assumption that the format will be static has consequences for its further development and use.

First, and most clearly, omitting the ability to version the file means that currently no FITS reader knows what version of the format it is reading. The usual assumption is that the latest version (v3.0) is being read, however, there are still many data which predate this version of FITS serialization making this supposition often incorrect.

---

[12]The rule dates back to the earliest FITS standard document that was produced by by the NOST at Goddard in 1993. This rule, in turn, probably had its origin in the paper by Grosbol et al. 1988, based on the statement:

"The most important rule for designing new extensions to FITS is that existing FITS tapes must remain valid. We are not permitted to alter the basic format in such a way as to make existing FITS tapes invalid or unreadable by standard FITS tape reading programs"

From a private communication with Bill Pence, 2013.

[13] see http://www.digitalpreservation.gov/formats/fdd/fdd000317.shtml

The lack of versioning also limits the ability of our community to move forward constructively with developing new FITS versions. Because we must maintain slavish conformance to older format conventions, it means that we accrete over time more and more “design rules” (or limitations, if you will) which limit our ability to implement new and needed features. Consider the awkward way, for example, in which the HIERARCH convention was implemented (Wicenec, Grosbol & Pence 2009). In order to provide the needed functionality, the designers were forced into utilizing COMMENT keywords for non-comment functionality. This behavior by the community will only increase, fragmenting the FITS format through inaction rather than coordinated design.

Both of these issues needlessly drive greater and greater complexity in implementation of FITS parsers. Furthermore, lack of versioning arguably works counter to the purpose of increased archivability of the files. Counter to the maxim “once FITS, forever FITS”, FITS reader software are not able to read just any version of FITS. Certainly, were someone to utilize code developed in the early 1980s to read a FITS file containing a binary table, or a FITS file with extensions, it would fail because it would have no idea how to handle these data structures. Suitability for archiving not only means that software are backward compatible, but that older software can gracefully handle newer versions of the format.

Versioning should be made part of the FITS standard in order to address these problems. We cannot simply solve the lack of versioning by adding a convention; conventions are not required to be implemented and this is fundamental metadata about the serialization of FITS itself. Furthermore, without the enforcement of being part of the standard, versioning is unlikely to be implemented where it is needed most, in the generation of new FITS files. Since any group can define and use their own conventions to store real, meaningful data, it is almost certain that since these conventions are not defined as part of the file format (or even in a central location), at some time in the future these definitions, along with the usefulness of the data, will be lost.

2.1.2. Inadequate declaration and validation of content meaning

Related to, but separate from the lack of versioning of the serialization, is the lack of ability to declare the presence of data models and their associated semantic meaning. By ‘data model’ we adopt the first meaning available in the Wikipedia which is:

“a description of the objects represented by a computer system together with their properties and relationships; these are typically ‘real world’ objects such as products, suppliers, customers, and orders.”

Of course, objects in astronomy are more likely to involve things like observations, instruments, and actual astronomical objects such as stars. Likely properties one will encounter in a FITS file include things like observational parameters (start/end times), astronomical coordinates, name and properties of the observing instrumentation, and so forth. In FITS-speak, we can say that any FITS keyword outside those defined in the FITS standard is a data model parameter, and collections of related FITS keywords form a data model. Ideally a data model should be associated with a given, unique, namespace so that collisions in naming of the models and requisite parameters are avoided.

Important questions such as “are all of the data here?”, “is the information in the file correct?” (no bad values for metadata/data), “do the data conform to expected data model?”, and fundamentally “what are these data you gave me about?” are all, generally speaking, questions of “validation” which are related to the data model(s) contained in any given file.

To illustrate better how FITS fails in this regard, consider that the FITS community already has many shared, common data models. WCS and some other FITS conventions such as OIFITS (Thureau et al. 2006), MBFITS (Muders et al. 2006), and FITS-IDI (Griesen 2001) declare data models. The declaration of keyword dictionaries[14] is also essentially an act of declaring a namespace and related data model(s). So it is clear that we have need of these things, yet, the only way in which we can currently determine that a file conforms to one or more of these models is for a human to communicate this to one another, for a human to inspect the file personally, or for special purpose software, targeted to particular data models/namespaces, to be written to detect the presence of the model(s) in the file.

Given the data volumes that we have in astronomy, the later choice is clearly in the direction we should go, but is not practical in the general case. The writing of generalized software to detect any data models present in a given FITS file is currently a hopeless act as there are constantly new data models being created and modified. Keeping the code up to date will be problematic, at best. Furthermore,

[14] Some collected data dictionaries with FITS keywords may be seen at the GSFC FITS site, see http://fits.gsfc.nasa.gov/fits_dictionary.html

there is the possible complication of more than one data model being present. Without explicit signposts in the file, it is likely hopeless to disambiguate which data models are present.

A means to capture the data model/namespace information, in the file, should be found so that validation can be possible without human inspection or specifically written software (similar, perhaps, in the way that JSON or XML formats have schema). This approach has additional benefits downstream. First, it will help to avoid misinterpretation because it is better for the creator of the data and/or data model to provide the machine readable information rather than a downstream programmer. Second, there is a saving in effort in that the model is done once and need not be repeated by numerous downstream programmers.

2.1.3. *No support for alternative encodings*

The allowed character set for metadata and data in FITS is overly restrictive and is limiting its application. The restrictions between metadata and data do not differ significantly. For metadata, the keywords which occur in the FITS header, FITS only supports the lower 7 bits of the US-ASCII encoding for keyword values and comments. For data encoding, authors may again only use the lower 7 bits of ASCII for text (string) capture in either ASCII or binary FITS tables although the NULL character is allowed in certain cases in binary tables.

The world of astronomy has evolved beyond capturing of scientific information in 7-bit ASCII encoding. The FITS community has grown. Now many data are captured by instruments designed, built and run by non-English speaking investigators and astronomical research has grown significantly elsewhere in the world. Whether as original observational data products, reduced data, information from new services or in capturing theoretical data, FITS is now required to hold many data which are no longer originating in English-speaking countries.

The current restrictive character set is really an anachronism. Consider that the most common language on the planet, Chinese, cannot be utilized easily in a FITS file. Forcing information into English can easily result in loss of valuable meaning or force the author to use some other format to store their data.

It is clear to us that support for alternative encodings is needed. Simple issues which revolve around the value of keywords like the expression of a person’s name (with accents, for example), or the ability to utilize special scientific and mathematical

symbols (like the degree) should be handled. Tabular text values should similarly be allowed alternative encodings for the same reasons. Furthermore, while not as critical, the FITS community should also consider the utility of allowing FITS keywords themselves to be expressed in a broader range of characters.

2.1.4 Informal variants

Although not explicitly part of the official FITS standard, the community has, in effect, accepted many variants from the standard as being acceptable through their wide use. In practical terms this means general FITS libraries must support these variants, significantly complicating the software, and introducing inconsistencies between libraries in which particular variants are not uniformly supported. This is partly due to the lack of good validating tools early on, but also due to a unwillingness to enforce the standard. As an example, the newer VOTable format had a clearer specification, and much better tools available for validation, yet a significant fraction of existing VOTables being provided do not adhere strictly to the standard. This issue will not be solved with a new format unless the community takes the standard more seriously than it has in the past.

Preventing this from happening with a new format will be aided significantly if:

- A strict validation tool is provided from the very beginning so that those making files have an easy-to-use tool that indicates what problems exist with the files they are creating.
- Those that write libraries to support a new format take a hard line against adding support for illegal variants.
- Discouraging developers from making ad hoc tools to create files in the new format rather than adopting use of a well-tested library (such ad hoc tools are often the source of problem files)

2.2. Missing critical data models

It is not the case that FITS is missing all needed data models. In terms of structures which represent basic organizations of the information used in astronomy FITS includes such things like “table”, “image” or “data cube”. These items are really simple representations of the data at a primitive level, and are certainly needed for basic

access to the information within the file.

Nevertheless, these structures, by themselves, do not contain much in the way of necessary detail and semantic information which tells the consumer exactly “what” it is they are actually consuming. While these structures do provide for the capture of semantic information (such as in the form of table column keywords) an additional data model (or more) must still be created or implemented by the author of the FITS file if there is a desire to convey “what” it is. Consider the many types of images or tables which are possible. Semantic information, in the form of a data model, allows the author to properly label the information so that it can be scientifically consumed without human interpretation. Such information includes the WCS, for example.

Our concern here is that there are either data models which are insufficient or missing in FITS and which are very basic to scientific (and in particular, astronomical) research. *Other astronomical groups have recognized the need for some of these models*. We detail several important missing data models in this section below.

2.2.1. Scientific Errors

The measurement of physical properties is fundamental to astronomy research. It is thus ironic that FITS, which is purposely designed for supporting astronomical research, has no standard data model for capturing information about scientific errors.

We could easily list a great number of possible error types which might be useful but trying to encompass all of the needs of the community at once is likely to create an unwieldy data model. We suggest that the community needs to provide for the most common needs, and target that subset as a first, shared model. Earlier efforts which might inform and help this work include local data models at sites such as NOAO and STSci, the error models implemented in other data formats like NDF, and software efforts underway in scientific programming communities such as AstroPy. Each of these have valuable insight into the requirements.

Nevertheless, we can anticipate that the following general characteristics might be part of the model:

- ❑ Allow for both metadata and data to have errors.

❑ Allow for extensible classification of the error type. For example, “Gaussian” errors are also a subclass of “statistical” errors.

❑ Allow association of more than one error class/type per measurement. For example, allow for both systematic and statistical errors to be associated with each measurement.

❑ Allow for additional properties to be associated with each error class. For example, “statistical” errors may have assigned “sigma” value.

2.2.2. Limited WCS

The existing FITS WCS data models illustrate some of the limitations associated with FITS. The “once FITS, always FITS” idea required that the current WCS standards were developed as an extension of the older AIPS standard, and so inherited many of the inherent limitations of that system. Even so they took a long time to be agreed. They are complex yet incomplete and inflexible. They are inadequate for many modern telescopes, and restrict creative use of novel coordinate transformations in subsequent data analysis. For instance, raw data must handle more distortion issues than the FITS WCS standard projections can handle. There are some provisions for handling more arbitrary distortions, but they are either cumbersome or too simple. Perhaps the biggest limitation is that different transformations of coordinates cannot be combined in flexible ways. One is effectively limited to choosing only one of the solutions available.

This is unfortunate. Not only does it reduce the range of transformations that can be described, but it also makes it harder to analyse the total transformation into its component parts thus making understanding and manipulation of the total transformation harder. The alternative approach - a “toolkit”-style system that creates complex transformations by stacking simpler atomic mappings - is usually the most efficient representation as far as data storage is concerned.

As an example, HST data requires multiple distortion components. Some are small but discontinuous. Other are linear but time varying. There is no FITS WCS compatible solution that handles these needs well. [*Other examples welcome*].

Another case with poor support is IFU data. Many of these data sets have

discontinuous WCS models. The only way to support these in FITS now is to explicitly map each pixel to the world coordinates. Besides being space inefficient, it is difficult to manipulate in any simple way.

In addition to limiting the description of raw telescope data, FITS WCS also restricts what can be done with such data during subsequent analysis. There are many potentially interesting transformations that would result in the final WCS being inexpressible using the restrictive FITS model. For instance, transforming an image of an elliptical galaxy into polar or elliptical coordinates. Another example may be attaching an alternate coordinate system to an image to represent the pixel coordinates of a second image covering the same part of the sky. These may not be common requirements, but they illustrate the wide range of transformation that should be possible with a flexible WCS system.

The inflexibility in the FITS solution arises from multiple issues, but lack of namespaces is a serious barrier to providing a more flexible solution. If one has multiple model components each with similar parameters, how does one distinguish between them? One may use the letter suffix, but that is also used to distinguish between alternate WCS models. The limitation on keyword sizes presents limitations on how many coefficients can be supported. The lack of any explicit grouping mechanism requires complex conventions on how to relate whole sets of keywords. With more modern structures, such contortions and limitations are not necessary.

The reality is that to solve these problems, many software systems have chosen alternate solutions and save their WCS information in FITS files in other ways (or in separate files). For example, the AST library (Warren-Smith & Berry 1998) is not subject to these limitations, but is forced to use non-standard FITS representations when representing data in its own native FITS encoding.

2.2.3. History/Provenance

IVOA provenance model as an example of usage?

The FITS standard encourages people to store processing history information in the header using a pseudo-comment field named *HISTORY*. This works from the perspective of making the information available to an interested human (assuming that each step in the data processing adds information to the end of the history section of the header) but the free-form nature of the entries makes it essentially

impossible for a software system to understand what was done to the data. This may be possible within the constraints of a single data reduction environment but it is highly unlikely that the content of the *HISTORY* block can be understood by any other software. History needs to be treated as a first class citizen with a standardized way of registering important information such as the date, the software tool and any relevant arguments or switches. A library interface should also be encouraged to make it easy for people to add history entries.

A related issue is data provenance. For a given processed data product it is, in general, impossible to determine which data files contributed. A data consumer would like to be able to find out which raw data files were included and any relevant intermediate products. There is no metadata standard for specifying this information in output files and ideally this should again be supported by a programmatic interface to minimize the mental overhead in tracking ancestor data.

2.2.4. Data Quality

One of the more pressing needs in our era of shared and distributed data is the need to know which data are “good” or, put another way, of sufficient quality. We are long past the era when the data volume was so small that it is practical to download all of the possible data of interest and examine it locally.

Some might insist that this is an easily solved problem. Simply declare a keyword, like “DQUALITY”, and allow it to take a boolean value. That example is an exaggeration, to be sure, but it helps to illustrate that simplicity in this regard, is simplistic. Data quality cannot be judged on a single, or even a small set, of parameters. The data which are adequate for one type of use, may be wholly inadequate in another usage context. Consider that engineering data generally are unsuitable for science and vice versa. Many science data may be unsuitable for other types of science (for example, studies of sky background vs pointed source science).

A data quality model then, should be an ensemble of common statistical measures of the type of dataset which may be used to derive higher-level judgements of the quality/suitability of the data for some other declared purpose. There are many higher types of data quality models which will need be created from the lower-level measures (image data quality, pointed catalog data quality, etc) and from these particular, targeted, statistical measures data quality may be judged by the dataset consumer without directly examining the data themselves.

2.2.5. Units

The FITS support for units is syntactically flexible (albeit with a few specification ambiguities which would be resolved by an explicit grammar), but the model does not accommodate the full range of contemporary astronomical data. This is evident from the significant fraction of unit strings in current databases which do not conform to the FITS prescriptions. The solution is not simply to expand the list of recommended units since, as well as being slow, this fails to distinguish between, for example, different definitions of the second, or to communicate places where the distinction does or does not matter.

There is also a provenance issue to defining new units, since -- to pick an example of a unit of 'jupiter core mass', two different groups may prefer different values for the mass, and indeed may not regard it as simply an abbreviation for a certain number of kilogrammes, but as a scaling mass whose value is to be fixed by a separate, subsequent, calibration.

The purely syntactical issues surrounding unit strings are to some extent being addressed by the IVOA's 'VOUnits' work, but the higher level questions -- of communicating and defining new units, of indicating documentation, and of converting between them in a scientifically meaningful manner -- are out of scope for that work and should be considered by the FITS community.

2.3. Inflexibility in organizing and representing metadata and data

What may be construed to be “FITS data” has changed significantly since the founding of FITS. The original FITS specification mandated only the capture of astronomical images. Almost a decade later, FITS extensions (Grosbol et al. 1988), allowing for gathering multiple related data structures in one FITS file, and ASCII tables (Harten et al. 1988) were introduced. Binary tables followed in the next decade (Cotton, Tody & Pence 1995). Changes have also occurred in metadata capture. Over the intervening years the FITS community has added new metadata conventions (for example, HIERARCH and GROUPING; Jennings et al. 2007) which have allowed for greater flexibility in capturing metadata.

This expansion is to be expected and is all to the good. Nevertheless, as we shall illustrate below, the expanded abilities to capture information are still insufficient relative to actual need.

2.3.1. Constrained metadata representation

The basic element of metadata capture in a FITS file is the FITS “card” which comprises a keyword name, a keyword value and a comment. All FITS cards must contain the keyword name and keyword value pair while comments in cards are optional. Comments may sometimes contain information about units of the metadata value.

Metadata may comprise a rich assortment of data structures and single-valued metadata are only the beginning. There is need to capture sets, lists, vectors and objects within metadata (to name only the most basic of structures). Yet FITS cards, without additional conventions, are only capable of capturing single, scalar keyword-value pairings. The expression of non-scalar, multi-valued keywords is difficult in FITS and data model designers often resort to special conventions to achieve this. For example, in order to hold a keyword with either a ‘set’ or ‘list’ value, a common, non-standard, convention adopted is to create a set of keywords sharing the same base name followed by a integer value which may (or may not) indicate order of the values (such as “ICMB001”, “ICMB002”, and so on).

This limitation on the range of metadata value structures is not the only problem, even the expression single scalar values are constrained. Consider that FITS cards are limited to 80 chars and FITS keyword names may be no longer than 8 chars. The result of these constraints is that keyword values may be no longer than 68 chars. Of course, if you utilize all of the space for keyword values, then the comment, or keyword values longer than 68 characters will need another convention in order to capture it (such as creating a continuation line in the header using the CONTINUE convention[15]).

Let us now consider the impact of keyword name constraints. Not only are keyword names limited to a small set of characters but keyword names are restricted to no more than 8 characters. Often these restrictions prevent clear labelling of the metadata element because authors are forced to map longer, more descriptive,

[15] see http://fits.gsfc.nasa.gov/registry/continue/continue.pdf

names into the truncated size. Non-English authors are additionally forced to map into the limited character set. If you doubt this leads to problems, try the following experiment: open any non-trivial FITS file and scan the header. Unless you are an expert in the data models present in the file (and sometimes even if you are!) its easy to find that the cramped names of the keywords often leads to arcane and confusing metadata.

These restrictions on the FITS card have other impacts as well. They can and do limit the utility of FITS conventions which may be implemented. Take, for example, the previously mentioned issue with the HIERARCH convention and its use of COMMENT keywords for non-comment purposes.

Finally, while being the least problematic of these issues, but still important, is the problem of the 2880 character logical record. While it is largely a low-level oddity of FITS now, it does have some small practical impact. Consider that this formatting requirement negatively impacts many FITS files today which have significant metadata. It often results in odd sections of the FITS file containing large blocks of white space. Certainly, in FITS files which have very large images (> 10 Mb or more) this is not a particularly pressing problem. However, for FITS files which have small amounts of data (or are oriented primarily to metadata transport or capture) this is simply wasteful.

The issues with metadata expression are serious for FITS with the problems caused by the limitations on allowed FITS keyword names and value data structures being the most significant. Certainly, as we have already noted, in many cases it is possible to create one or more conventions to expand the expression of the metadata but doing so will result in files which are unparsable by downstream software which don't implement these conventions. There is no apparent solution, using a FITS convention, than can solve the limitation on the restricted expression of keyword names.

All of these metadata limitations are contrary to both the "interchange" and "archivability" use cases for FITS. If the FITS standard is followed with additional conventions we still need specialized software and/or a human to successfully parse the content of the metadata present. If the FITS standard is rigorously followed sans conventions, then the author is severely limited in the metadata they may express.

Keyword names and values must be allowed to be longer, and an expanded character set would be useful as well. FITS metadata should be able to express, as part of the FITS standard, many more data structures as metadata. These include, but

are not limited to things such as sets, lists and objects.

2.3.2. Awkwardness in representing the associations present in both data and metadata

As data acquisition and data reduction systems have become more complex there has been a move to storing multiple image data components in extensions within a single FITS files. The FITS extension mechanism provides a scheme for having multiple images but in essentially a flat structure. If you have 12 images in the file there is no way of indicating that 3 of them are data, 3 are an error and 3 are a quality mask. Indeed, there is no way of specifying which triplets are related. You can use the EXTNAME header to indicate relationships but this relies on convention and string parsing rather than being a standard part of the format.

The conversion of NDF format files (see Poster P91) to FITS and back to NDF has demonstrated that you can represent a hierarchical data grouping in the FITS multi-extension format, but this is done using EXTNAME conventions combined with headers representing the extension level in the hierarchy and the type of component and so is not understood by other FITS tools. A standardised way of specifying relationships between extensions would be extremely valuable to data and application interoperability.

2.3.3. No support for declaring byte order

Although in the base FITS implementation (Wells et al. 1981) a particular byte order was not designated, current versions of FITS store data in big endian format only. Sometimes the byte order needs to be checked and swapped to the native byte order in little endian architectures when non-native data formats are not supported (as in implementations of FITS readers which utilize C , Cython, and so on). Programmers on little endian platforms who work with large data volumes may find this limitation results in a significant performance cost as marshalling data to and from the FITS big endian ordering will be required. This is a frequent problem for astronomical software. Little endianness is found on x86 and x86-64 processors that are commonly used in Universities and research laboratories.

FITS should be able to declare the byte ordering of data by using a FITS keyword defined in the standard dictionary.

2.4. *Inadequate support for large, distributed data*

Introduction to this section needed here…

2.4.1. No support for capturing indeterminately sized data sets via streaming

Frequently there is a need to store data from an instrument or remote site that is being transmitted over a network. It is common that when the transfer begins the final size of the data set is not known. Those using FITS have handled this by writing such data to a file without specifying the size of the last dimension in an image or table, and when the stream is completed, the header is appropriately updated.

Nevertheless, there are applications for which one would like to access all the other information before the file is complete. This may be to integrate the data that is being read out, or to monitor metadata. A library supporting the data format should support such usage.

2.4.2. No support for declaration of virtual and distributed components

When FITS was created, the ‘file’ (bytes stored on durable physical medium such as spinning disk or magnetic tape) was more or less the only way to store and transfer data. The networked solutions which we enjoy today were absent from the world of astronomy and storage of astronomical data in databases was unusual. Code was run locally by experts and the results, if shared at all, were usually only reported in published papers. In the intervening years, computer and information technologies have evolved and broadened; we now enjoy many new means of accessing, providing and storing data. FITS should join this revolution.

We should start to consider thinking of FITS as a ‘container’ of astronomical information which is not necessarily a file. Is there any reason to prevent our FITS ‘file’ from overlaying a portion of a database? Why not allow FITS to be a wrapper about bytes held within a distributed mass store such as IRODS or a cloud? Similarly, we would want FITS to contain, and adequately access, data generated by a service (think simulation data, for example). Possibly FITS could itself execute simple stored

algorithms to generate a portion of its data[16].

These use cases are only examples of where we might go and some may be arguably of limited value. Nevertheless, the generalized use case that may be derived from all of these is of certain importance, namely, FITS should be able to seamlessly specify data both contained locally within itself and reference (and transparently access) data held in non-local entities such as cloud storage, databases, and services and other files.

The grouping convention includes a provision for specifying URI's, but the implementation is not adequate. There is no provision for directing the machine on how to access the resource specified by the URI.

3. Summary - Significant problems exist in the FITS standard

The problems which we have described are real and significant, however we don't wish to recommend a particular solution here to any of these. Action to correct these issues should flow from constructive community discussion and offered solutions to these problems. Possible solutions may involve moving existing FITS conventions into the core standard, modification of the FITS standard to remove limitations or possibly transferring the FITS data model over into a new serialisation, or some selection of these actions.

These technical problems will be solved one way or another. If the community is not willing to do the hard work of hammering out a universal (or widely-adopted) approach, individual projects will continue to make their own ad-hoc solutions. Data formats will become increasingly fragmented and we will no longer enjoy the easy interoperability that FITS has provided for many years.

4. References

Anderson, K. et al., 2011, in "Astronomical Data Analysis Software and Systems", vol.

[16] This probably implies the need for a "FITS language" to generate these data

442, ASP Conf. Ser., 53

Alexov, A. et al., 2012, in “Astronomical Data Analysis Software and Systems”, vol. 461, ASP Conf. Ser., 283

Allegrezza**,** S., 2012, “*Flexible Image Transport System: a new standard file format for long-term preservation projects”,* presentation given at EWASS

Berry, D.S., 2001, in "Astronomical Data Analysis Software and Systems X", Vol 238, ASP Conf Ser., 129

Calabretta, M. R., and Greisen, E. W., 2002, Astronomy & Astrophysics, 395, 1077

Cotton,W.D., Tody, D., and Pence, W.D., 1995, A&A Supplement, 113, 159

Currie, M.J., 1988, Starlink Bulletin, 2, 11

Currie, M.J., Draper, P.W., Berry, D.S., Jenness, T., Cavanagh, B., Economou, F., 2008, in "Astronomical Data Analysis Software and System XVII", Vo. 394, ASP Conf. Ser., 650

Currie, M.J., Wallace, P.T., & Warren-Smith, R.F., 2009, Starlink General Paper 38, Starlink Project, SERC

Disney, M.J., and Wallace, P.T., 1982, QJRAS, 23, 485

Greisen, E. W., 20011, AIPS Memo 114r: The FITS Interferometry Data Interchange Convention

Greisen, E. W., and Calabretta, M. R., 2002, Astronomy & Astrophysics, 395, 1061

Greisen, E. W., Calabretta, M. R., Valdes, F. G., and Allen, S. L., Astronomy & Astrophysics, 446, 747, 2006

Grosbol, P., Harten, R.H, Greisen, E.W., and Wells, D.C., 1988, A&A Supplement Series, vol. 73, no. 3, p. 359

Harten, R.H., Grosbol, P., Greisen, E.W., and Wells, D.C., 1988, A&A Supplement Series, vol. 73, no. 3, 365

Jennings, D.G., Pence, W.D., Folk, M., and Schlesinger, B.M., 2007, “A Hierarchical Grouping Convention for FITS”, http://fits.gsfc.nasa.gov/registry/grouping/grouping.pdf

Jenness, T., Berry, D.S., Cavanagh, B., Currie, M.J., Draper, P.W., Economou, F., 2009, in "Astronomical Data Analysis Software and Systems XVIII", Vol. 411, ASP Conf. Ser., 418

Muders, D., Hafok, H., Wyrowski, F., Polehampton, E., Belloche, A., Konig, C., Schaaf, R., Schuller, F., Hatchell, J., & van der Tak, F., 2006, A&A, 454, L25

Pence, W. D., Chiappetti, L., Page, C.G., Shaw, R.A., and Stobie, E, 2010, A&A, 524

Rasband, W., 2010, Astrophysics Source Code Library, record ascl:1206.013

Shaya, E., Thomas, B., Gass, J., Blackwell, J. and Cheung, C., 2000, Bulletin of the American Astronomical Society, 32, 1600

Thomas, B., Shaya, E., Gass, J., Blackwell, J. and Cheung, C., 2000, Bulletin of the American Astronomical Society, 32, 1600

Thomas, B., Shaya, E., and Cheung, C., 2001, in “"Astronomical Data Analysis Software and Systems X", Vol. 238, ASP Conf. Ser., 487

Thureau, N.D., Ireland, M., Monnier, J. D., & Pedretti, E., 2006, in Advances in Stellar Interferometry, vol. 6268 of Proc. SPIE, 62683C

Warren-Smith, R.F., & Wallace, P.T., 1993, in "Astronomical Data Analysis Software and Systems II", Vol. 52, ASP Conf. Ser., 229

Warren-Smith, R.F., & Berry, D.S., 1998, in “Astronomical Data Analysis Software and Systems VII”, Vol. 145, ASP Conf. Ser., 41

Wells, D.C., Greisen, E.W., and Harten, R.H., 1981, A&A Supplement, 44, 363

Wicenec, A., Grosbol, P., and Pence, W., 2009, “The ESO HIERARCH Keyword Conventions”, http://fits.gsfc.nasa.gov/registry/hierarch_keyword.html