Balancing Performance and Preservation
Lessons learned with HDF5

Mike Folk
The HDF Group
1901 S. First St.
Champaign IL 61820
1-217-244-064
mfolk@hdfgroup.org

Elena Pourmal
The HDF Group
1901 S. First St.
Champaign IL 61820
1-217-333-0238
epourmal@hdfgroup.org

ABSTRACT

Fifteen years ago, The HDF Group set out to re-invent the HDF format and software suite to address two conflicting challenges. The first was to enable exceptionally scalable, extensible storage and access for every kind of scientific and engineering data. The second was to facilitate access to data stored in the HDF long into the future.

This challenge grew out of necessity. Some of the most ambitious scientific projects, such as NASA's Earth Observing System, need scalable solutions to their data generation and data gathering activities. At the same time, data consumers in these projects need assurances that their data will retain its value and accessibility for decades to centuries into the future. The HDF Group has worked to discover and pursue technological and institutional strategies that address these requirements for the broadest possible range of data applications.

To achieve this objective, care and resources must be applied in the design, development, and maintenance of the technologies, and attention must be paid to integration with complementary technologies. This technical rigor must be complemented by an institutional model that will provide resources for current activities and sustainability for the long term, as well as active involvement with data producers and consumers to understand and respond to their needs.

The paper describes how The HDF Group balances its commitment to providing the best solutions to today's data challenges against the need to meet data preservation requirements.

Categories and Subject Descriptors

E.5 [**Data**]: Files – *organization/structure*

General Terms

Management, Standardization.

Keywords

Long-term data preservation, HDF, HDF4, HDF5, open source, data archive.

1. INTRODUCTION

1.1 Data challenges

Among the greatest challenges to managing data today are the ability to manage big data, diverse types of data, and complex data relationships. This data must be processed, analyzed, viewed, and queried on computers of every type and size, and must be accessible from a large and sometimes widely dispersed number of data sources. The technologies for collecting and generating data are evolving at a rapid rate, as are the data technologies for addressing these challenges. At the same time, a great deal of the data we collect and create has a very long shelf life.

The juxtaposition of the immediate data needs with the need to be able to access data for the long term raises important questions about how we develop data technologies that address both today's data challenges and the challenge of long-term data preservation.

Some of the most ambitious scientific projects exemplify this dilemma. NASA's Earth Observing System, for example, needs scalable solutions to its data generation and data gathering activities. At the same time, the project needs assurances that its data will retain its value and accessibility for decades to centuries into the future. The HDF Group has worked to discover and pursue technological and institutional strategies that address these requirements for the broadest possible range of data applications.

1.2 What is HDF?

The term "HDF" (Hierarchical Data Format) refers to two different technology packages, each consisting of a data file format and a suite of software for managing data stored in the format. HDF4, originally called "HDF," was developed at the National Center for Supercomputing Applications (NCSA). Originally released in 1988, it continues to be supported by The HDF Group. HDF5 was also developed at NCSA, with significant support from the Department of Energy and NASA. HDF5 is the successor to HDF4, and was first release in 1998. In the following discussion, we focus on HDF5, but many of the same principles and capabilities also apply to HDF4, in particular those regarding data preservation.

The HDF5 file format is designed to allow generic self-description of data objects stored within it. The HDF5 data model includes a

directed graph structure, which allows objects to be organized in whatever way is most meaningful to an application or family of applications. Data granules are stored in HDF5 "datasets," which are multidimensional arrays, with an extremely rich variety of data types available for representing array elements. A structure called "attributes" is available for attaching application metadata to HDF5 objects. This combination of structures makes it possible to store virtually any kind of data object, and any combination of data objects, in an HDF5 file.

The HDF5 technology platform has three parts:

- Software for managing, analyzing, acquiring, and querying HDF5 data. This includes an open source HDF5 library, a set of command line tools, and a large number of third party applications.

- The HDF5 data model, consisting of the building blocks for organizing and storing data.

- The HDF5 file format, a specification for the bit-level organization of an HDF5 file.

1.3 The HDF support system

HDF4 and HDF5 are backed by The HDF Group. In 2006 The HDF Group left the University of Illinois expressly to strengthen its ability to support HDF users, to evolve and sustain HDF-based technologies, to foster preservation of data stored in HDF, and to achieve institutional sustainability. As a non-profit 501(c)(3) organization, The HDF Group continually pursues technical, legal, and business strategies aimed at achieving these goals.

2. IMPACT

The HDF community – organizations that use HDF, contributing developers, and The HDF Group – make up a support system that helps users overcome data challenges effectively and at low cost. By building on an HDF foundation, organizations automatically reap the benefits of HDF solutions, rather than having to bear the cost of inventing and maintaining those solutions themselves. Because of its broad base of users and contributors, HDF embodies capabilities and features that most individual data management projects would lack the resources or knowledge to implement.

The ability of HDF efficiently to store almost any kind of data simplifies the packaging of data and means that all of the information associated with an application can remain in one package. This capability facilitates data integration, interoperability, and long term preservation.

2.1 Current target audience

HDF5 is used in every scientific discipline, and in many applications outside of science and engineering, throughout academia, the government, and industry. NASA's Earth Observing System uses both HDF4 and HDF5, and many related projects use, or plan to use, HDF as well. Many simulation codes on today's largest systems use HDF5. Applications as diverse as electron tomography, high throughput DNA sequencing, aircraft flight testing, military vehicle testing, and film-making rely on HDF for handling high volume and complex data. HDF has become a standard for many communities, and an ISO standard is under development for product data representation and exchange in HDF5.

A common use of HDF is as an underlying platform for data that can occur in other formats. For instance, netCDF-4 is a re-implementation of netCDF using HDF5 to take advantage of HDF5 scalability that was unavailable in the original netCDF format. MATLAB uses HDF5 as its format for similar reasons.

2.2 Motivation for HDF preservation features and processes

The original design of HDF did not anticipate the need for long term data preservation. It focused instead on organizing and sharing scientific data, and on managing data in high performance computing environments. Even before its selection as the standard format for NASA's Earth Observing System in 1994, users express interest in long-term preservation of data stored in HDF. In the mid 1990's NASA asked the HDF project to explore the issue of insuring long term accessibility to HDF data. Other applications, especially those involving critical test data, made it clear that long-term access to their data was important.

2.3 Target audience of the future

In response to NASA's concern, the HDF project developed a set of criteria to answer the question "What makes a good archive format." These were later presented in the paper "Attributes of File Formats for Long-Term Preservation of Scientific and Engineering Data in Digital Libraries"[1]. The paper identifies three communities that can have very different responsibilities and perspectives: data producers, archives, and data consumers.

Data producers are generally most concerned with making the data available as quickly as possible and in a form that is most usable by its immediate users.

Archives include both "active archives" and "long-term archives." *Active archives* serve the community of users who have immediate or intermediate-term use for the data. *Long-term archives* are primarily responsible for preserving data for future generations.

Data consumers include both active archive users and long-term archive users. *Active archive users* often have a good understanding of the instruments or other methods used to collect the data, as well the context in which the data has been collected.

The needs of *long-term archive users* are often unknown at the time when data is produced. They are unlikely to have first-hand information about the instruments or methods used to produce the data. The context in which the data was produced may be long forgotten, or perhaps hidden in notebooks or other archived documents. These users may be concerned about the integrity of the data, which may have been in storage for decades, may have been processed and re-processed many times, and may have been migrated through many different storage media. The hardware and software available at the time when the data were produced may have long since disappeared, and even the programming language used to produce the software may be a thing of the past.

3. HDF STRATEGIES FOR LONG-TERM PRESERVATION

The paper [1] presents 25 criteria for considering the suitability of a data format for long term preservation (Figure 1). This list has driven much of the strategy in developing and supporting HDF. Some HDF strategies are technology based and others are based on institutional policies and practices. They are covered in turn in the next two sections.

Ease of Storage
- o Compactness
- o Size
- o Ability to aggregate related objects.

Ease of Access
- o Raw I/O efficiency
- o Ease of subsetting

Usability
- o Popularity
- o Availability of readers
- o Ability to embed data extraction software
- o Ease of implementing readers
- o Simplicity
- o Ability to name file elements

Support for Data Scholarship
- o Provenance traceability
- o Rigorous definition
- o Self-describing
- o Citability
- o Referential extensibility
- o URN embedding

Support for Data Integrity
- o Source verification
- o File corruption detection & correction

Maintainability and Durability
- o Long-term institutional support
- o Suitability different storage technologies
- o Stability
- o Formal description of format
- o Multi-language implementation of library
- o Open Source software or equivalent

Figure 1. Criteria for considering the suitability of a data format for long term preservation.

3.1 Technical strategies

One of the greatest challenges for a living, evolving suite of technologies such as HDF is the ability to continually improve the technology and integrate it within a constantly changing environment, and at the same time make sure that it remains durable.

3.1.1 A simple, durable but evolvable model and implementation

The original design of HDF4 was driven by a need for flexibility and extensibility, and this was accomplished by defining simple objects, identified internally by "tags", then building larger objects out of these simple ones. As HDF4 evolved, the number of objects to be supported continued to grow, and hence the complexity of the format. HDF4 was hard to learn, hard to maintain, and optimizations implemented for a particular object could not easily be applied to other objects.

HDF5 addressed this by using a simple, comprehensive data model with only two primary structures: a grouping structure and

a multidimensional array of data elements whose datatypes can be very general. All HDF4 structures can be derived from these HDF5 structures, and new data structures can be created as well.

Another important consideration for evolution is scalability. HDF4 objects were limited to 2 gigabytes in size, and the number of objects limited to about 20,000. HDF5 allows objects of any size and number to be described.

3.1.2 Self-description

HDF4 and HDF5 are self-describing formats in that they contain information about their contents that in other formats must be obtained from external sources. Both formats contain descriptions of their internal structures and data, such as the structure of indices used for random access, and the data types of array elements.

An HDF5 file optionally may contain a "user block," which is an optional block at the beginning of an HDF5 file that can contain any information an application puts there. It could contain anything from provenance information to a Java HDF5 reader.

3.1.3 Specification documentation

A challenge with open source development is to maintain rigor in documentation. Funding is far more readily available for technical features than it is for the underpinnings that ensure durability over time. The HDF development process requires that any change in the format be fully documented in the format specification, and encourages any change to the library or public tools be fully documented with a User's Guide and Reference Manual.

It is also important to document the data model represented by any complex format. Especially important as the format evolves, there is a danger of making ad hoc changes that can complicate the use of the format. Although a specification of the HDF5 data model exists [2], recent interest in using HDF5 in connection with relational databases has revealed a need for a machine readable, logically rigorous specification. Following the success of the XML Information Set method [3] for creating well-formed XML documents, an effort is underway to define an HDF5 Infoset. Not only will this enable new and powerful uses of HDF5, but it will provide a highly durable language for describing HDF5 files and collections for future generations.

3.1.4 Preservation-based evolution

These changes exemplify a general principle for HDF5 development: "preservation-based evolution." Preservation-based evolution is a technology development strategy that allows the software and format to evolve, while preserving access to all data, and widening the software compatibility window sufficiently to give every application a chance to meet its users' needs.

This is a challenging principle in a world where the lifespan of a technology is often measured in months, and where it is expected that innovations will be superseded so quickly that little attention need be paid for compatibility with either the past or the future.

This requires a different mind-set for innovators, in particular for software developers. Guidelines have been developed among HDF designers and developers to keep evolution under control, and to make sure that questions of backward and forward compatibility are considered from the very beginning.

The underlying requirement for preservation-based evolution is to maintain compatibility across generations of the format and the

software that deals with the format. Access to legacy data must be preserved, and legacy software must be able to access as much future data as possible.

Compatibility must look both backward and forward. *Forward compatibility* from a format perspective means that older versions of software can reasonably read objects in an HDF file that were written by new versions of the software, despite changes in the format. From a software perspective, it means that applications written to work with older versions of the library can be rebuilt with newer versions of the library.

Similarly, *backward compatibility* preserves the ability of new library versions to read older files, and newer applications to build and run correctly with older versions of the library.

The meanings of forward and backward compatibility for HDF are summarized in Figure 2.

	Format	Software
Backward	Newer library versions can read a file and/or objects within a file that were created or written by older versions.	Applications written to work with a newer library version will compile, link, and run as would be expected with an older version.
Forward	Older library versions can read a file and/or objects within a file that were created or written by newer versions.	Applications written to work with an older library version will compile, link, and run as would be expected with a newer version.

Figure 2. The meanings of backward and forward compatibility.

To accomplish compatibility, each new version of the HDF5 format must be able to describe all data ever stored in HDF5 files, despite changes in technologies, media, and computing systems. Similarly, the latest HDF5 software should be able to access all data ever stored in HDF5 files, and must also anticipate future changes in the format, being able to ignore new objects that it cannot understand, while accessing all other objects as it normally would. This approach preserves the ability of HDF5 software to evolve to exploit and work with complementary technological advances while preserving long term availability and compatibility.

Preservation-based software evolution is not without considerable cost. Costs include regression testing in changing computational environments, monitoring the software lifecycle to make sure all parties adhere to compatibility principles, collaborating with users to find and address issues that occur, and bug fixing and other software development activities that result from these other activities.

3.1.5 *Providing different ways to view the same data*
One preservation risk associated with formats is that the software, the data model, or both will be lost through time. This risk can be mitigated by providing different ways to view and access data in the format. This only works, of course, when those different views or access methods endure, but having more options improves the chances that some option will survive.

We have engaged in this effort in three ways, which we will call *augmentation, format mapping,* and *conversion.*

Augmentation. The extensibility of the HDF formats makes it possible to augment an HDF file in ways that can sometimes provide alternate views. For example, an early HDF project in 1992 added structural information to the HDF format that made it compatible with the netCDF data model, and added the netCDF API to the HDF library. This meant that netCDF applications, including those in the future, could access HDF data within the netCDF framework. This API continues to be supported and used heavily by some applications.

Earlier we described how netCDF-4 uses HDF5 as its underlying format. If the netCDF data model survives and the HDF5 model does not, this data will endure.

Another example, which takes advantage of netCDF-4, is a project in with we are augmenting certain HDF-EOS files by adding structures to make those files compatible with netCDF-4 software.

Simplified access to data via independent format mapping. A key drawback for long-term archiving is the complex internal byte layout of HDF files, requiring one to use the library and API to access HDF data. This makes the long-term readability of HDF data for a given version dependent on long-term allocation of resources to support that version. To address this issue with respect to NASA data, The HDF Group and NASAs Earth Science Data Centers are developing methods for producing a map of the layout of the HDF4 files using a markup-language-based HDF tool. The resulting maps allow a separate program to read the file without recourse to the HDF API [4].

A prototype of this approach has demonstrated that by producing maps today of complex files such as HDF, future accessibility should be achievable quickly and at very low cost compared to the cost of reproducing the original library. A full production-ready version of an HDF4 map writer for EOS files is currently under development.

Conversion. A common way to improve the usability of data in a particular format is to provide tools that can convert the data into appropriate other formats. This can be a long-term preservation strategy when it is expected that the destination format will either endure longer than the source, or that the future community of users will find the destination format easier to use.

With a format such as HDF, conversion can be easy if objects to be converter are simple and common to both formats. For example converting an HDF4 image to PNG is simple. If a file contains more complex objects or collections of objects, the task becomes more difficult. For example a tool to convert from HDF4 to the FITS format required that a careful conceptual mapping be created between HDF4 objects and their organization to the corresponding FITS objects and organization.

The HDF project has developed conceptual maps and conversion tools for a few formats, most notably image formats such as JPEG and GIF, simple text formats, and XML.

3.1.6 *Integration with preservation frameworks*
Obvious opportunities to enable long term preservation lie in the many digital preservation projects and technologies in existence or under development. The Open Archive Information System

(OAIS) reference model provides a framework and tools that HDF can capitalize on, as exemplified by the following three activities.

In a NOAA sponsored Scientific Data Stewardship project, the HDF Group and the National Snow and Ice Data Center developed an HDF5 Archival Information Package (AIP) to archive data and metadata from the Earth Observing System [5] [6].

In a project to integrate HDF5 with iRODS data grid technologies, an HDF5 AIP for METS was implemented, together with a tool to extract HDF5 metadata and create the corresponding AIP [7].

The format mapping project described above plans to include a component in which the mapping information will be encapsulated within a PREMIS [8] framework.

3.2 Institutional strategies

In addition to technologies and technical strategies, the HDF Group believes that certain institutional strategies, principles, and practices are needed to preserve access for the long term. As the number of institutions committing their data to HDF grows, it is important that they engage with The HDF Group in developing and supporting these strategies, principles, and practices.

3.2.1 Long-term institutional support
The HDF Group believes that long term institutional support for HDF is needed to preserve long-term access to HDF data. There is great risk in letting HDF fend for itself, as is commonly the case with open source software, in which code is simply released to an open source repository, trusting that a community will somehow emerge and sustain the technology.

Important applications already rely heavily on being able to access data that is stored in HDF for decades and perhaps centuries in the future. As HDF grows in capabilities and acceptance, even more will join them. Without reliable long term institutional support, HDF technologies are at risk of fragmenting, deteriorating in quality, or disappearing altogether. As a result, anyone whose data is in HDF is at great risk of losing access to that data. Likewise, anyone whose data management operations depend on HDF technologies is at risk of losing the ability to effectively manage their data.

These risks are not just to organizations, but to society as a whole. Everyone benefits from good data preservation and management, and conversely can suffer from the loss of data and the ability effectively to manage data.

3.2.2 One keeper of the format and software
The HDF Group believes that having one well-funded, sustainable keeper of the format and software brings important advantages.

It becomes much easier and more likely that the software evolution process will attend to long-term preservation requirements. Centralized coordination, quality control, maintenance, and testing lower the overall cost. Knowledge, techniques, and auxiliary code that enhance the core software become more widely available. And the technologies are at less risk of fragmenting into incompatible, separately maintained products.

3.2.3 A mission-driven business
The purpose of the HDF Group is to play the role just described, namely to provide self-sustaining long-term institutional support for HDF. The institutional model embodies a mind-set committed to the long-term preservation mission. Originally a small University research group, The HDF Group decided that this could best be achieved by incorporating as an independent not-for-profit business.

3.2.4 Open source
The open source strategy of HDF increases the likelihood that HDF data will be available long into the future. Making the basic HDF formats and software free and open gives confidence that data can be accessed even if institutions that support HDF library and tools disappear. It gives confidence that the cost of accessing data will not change when supporting businesses change their pricing models.

3.2.5 Free as in speech, not as in beer
Although the basic formats and software are open and free, long-term support for HDF requires sustainable financial security for the stewards of HDF. The HDF Group's strategy has been to pursue a business model that can ultimately develop a financially sustainable institution. Funds are needed for both day-to-day management and to provide long-term security for the organization and its mission.

Financial security depends upon a mixture of funding mechanisms. Currently those mechanisms include income produced by offering services to HDF users. Current funding sources include special projects for hire, comprehensive support for major users, maintenance contracts, and consulting agreements. There are serious questions as to whether this is a viable long-term business model, and other options need to be explored.

3.2.6 Legal and financial foundations
The legal and financial underpinnings of HDF are designed to remove barriers to access for all data that is meant to be freely available, and the HDF license structure (currently based on the BSD license model) provides for free use and distribution of basic HDF software without cost or license restrictions. It is important to note that other options continue to be explored, as the commitment is to long-term access and usability, not to any particular IP strategy.

3.2.7 Cross the chasm to new users and applications
Another principle driving the HDF project is that the more users and applications there are of HDF, the more likely there will be continuing support and acceptance of HDF. Increasing the number and types of users increases in turn the strength and sustainability of the technology.

Growing the number of users and applications will increase the number of standardization efforts, which in turn promotes interoperability, data sharing, tools support, and activities supporting long-term preservation. Equally importantly, the potential sources of funding needed to ensure ongoing support for HDF is increased.

HDF strategies for growing its user base include expanding HDF into more new application domains, and working with developers and vendors to facilitate development and use of complementary

products, such as data analysis tools (e.g. MATLAB), data grids (iRODS) and database systems.

3.2.8 Promoting standardization

The benefits of standardization to long-term preservation are well known, and are a fundamental part of the HDF strategy. Standardization efforts include standardizing how HDF is used, as well as the two versions of HDF themselves.

A number of application domains have engaged in efforts to standardize how HDF is used, including earth science [9], computational fluid dynamics [10], bioinformatics [11], neutron scattering[1], netCDF[2], and product information (an ISO standard)[3].

From the HDF perspective, the building blocks of standardizing how HDF is used for a particular community include

- Agreement on a common format (or a small set of formats) that can meet a community's current and long-term needs.

- Development of a unified data model that embodies the structures and access methods of all applications, and effectively hides underlying implementation details.

- Specification of how the data model will be implemented: what HDF structures it will use and how it will organize them.

- Development and implementation of an API that embodies the data model, preferably in all programming languages considered important to the community.

- Creation of tools that meet the full range of user's needs, and adaptation of vendor and other software that can stretch the usability.

As for standardizing the two versions of HDF themselves, there are three different components that can be standardized: the specification of the format, the data model, and the API that implements the data model. An HDF5 data model standard has been defined for NASA [2], but has not yet reached a wider scope. Standardization of HDF4 and HDF5 themselves is an important goal, and there is much work to do in this area.

4. DEVELOPMENT PLAN

4.1 Lessons learned when developing HDF

The HDF5 project provided many opportunities to apply lessons learned from the original HDF project. Some of these are listed briefly here.

Recognize conceptual overlap. For long term preservation, simple models are better than complicated models, all else being equal. The original HDF project implemented a number of different objects that ultimately proved to have a great deal in common. For example, HDF includes an 8-bit raster image and a 24-bit raster object, each implemented completely separately, with

different code bases, different tests, different documentation. In HDF5, a single multidimensional array structure is the basis for all objects with essentially a regular grid shape, which simplifies the model appreciably.

Focus on being one good layer in the stack. Enthusiastic users will ask for every imaginable capability to be added to a file format. For example, there were requests for HDF to include very specialized metadata about images, such as the name of the instrument used to produce an image. This kind of information causes unnecessary clutter in the format that is of no value to the vast majority of users, often becomes obsolete over time, and is costly to maintain. In HDF5, improved structures are available for application-specific information, and applications are responsible for its inclusion and maintenance. Preservation efforts can focus on maintaining the desired information at the appropriate level.

Pay heed to backward and forward compatibility of both the format and the software. By articulating principles of backward and forward compatibility from its inception, the HDF5 project was able to design the format, the software, and testing and maintenance protocols in ways that facilitate long-term preservation.

Open source doesn't assure preservation. Valid arguments can be made to the value of an open source approach in facilitating long-term preservation. At the same time, if left to the whims of its supporters, an open source approach can make preservation more difficult. This has been evident in the HDF case, as the open source community is far more interested in features that address today's challenges than in making sure that those features are consistent with principles of long-term preservation. The HDF Group addresses this concern by controlling any changes to the format or software.

4.2 Future development plan

The HDF Group will continue to operate in support of the principles and processes described here to build a strong foundation for sustainability. Special attention will be given to the following.

Standards. The HDF Group plans to invest in developing domain-specific standards in the use of HDF, as well as to pursue standardization of HDF5.

Data model. We plan to work toward greater rigor in the data model, such as defining an HDF5 Infoset.

Backward/forward compatibility. We will continue to develop our processes and technologies for improving compatibility across generations of the formats and software.

Full support for HDF4, with few changes. As long as resources are available, The HDF Group will continue full support for HDF4, but with little or no changes to the software or format.

Preservation-based evolution for HDF5. The HDF Group will continue to actively develop the HDF5 file format and libraries to address current and future data challenges as outlined in this paper.

A sustainable institutional model. The HDF Group will continue to explore organizational models that can improve sustainability of the institution and its technologies.

---

1 NeXus (neutron, x-ray and muon science); http://www.nexusformat.org.

2 NetCDF-4 (network Common Data Form); http://www.unidata.ucar.edu/software/netcdf/.

3 List of STEP (ISO 10303) parts; http://en.wikipedia.org/wiki/List_of_STEP_(ISO_10303)_parts

Mappings. We will seek resources to implement layout mapping technologies for HDF5 similar to those being developed for HDF4.

Integration with preservation frameworks. We will continue to integrate HDF with preservation frameworks, such as those of the OAIS model.

Value-added products. HDF alone is of limited value in comparison to an HDF complemented by value-added products, standards, and services. The HDF Group must work with communities, vendors, and other technologies to meet a broader range of needs.

5. SUMMARY

HDF is a suite of data technologies for addressing some of our greatest data challenges. The HDF project has two objectives in its support and development of HDF. The first is to enable exceptionally scalable, extensible storage and access for every kind of scientific and engineering data. The second is to facilitate access to data stored in the HDF long into the future.

The HDF Group has worked to discover and pursue technological and institutional strategies that address these requirements for the broadest possible range of data applications. Technical strategies include creation of an evolvable data model and implementation, a self-describing format, preservation-aware software evolution, providing alternate views of data, and integration with existing preservation frameworks.

Institutional strategies include long-term institutional support, having one keeper of the format and basic software, a mission-driven organizational model, provision of open source software and formats, attention to legal issues, integration with complementary technologies, and promoting standardization.

6. ACKNOWLEDGEMENTS

7. REFERENCES

[1] Barkstrom, B., Folk, M. 2002. Attributes of File Formats for Long-Term Preservation of Scientific and Engineering Data in Digital Libraries. An HDF White Paper. http://www.hdfgroup.org/projects/nara/Sci_Formats_and_Archiving.pdf.

[2] Pourmal, E., Folk, M. 2007. HDF5 Data Model, File Format and Library – HDF5 1.6. ESDS-RFC-007v1 Recommended Standard. January 2007. http://www.esdswg.org/spg/rfc/ese-rfc-007/ESDS-RFC-007v1.pdf.

[3] Cowan, J., Tobin, R. (Editors). 2004 XML Information Set (Second Edition) W3C Recommendation. 4 February 2004. http://www.w3.org/TR/xml-infoset/.

[4] Duerr, R., Cao, P., Crider, J., Folk, M., Lynnes, C, and Yang, M. 2008. Ensuring Long-Term Access to Remotely Sensed Data with Layout Maps. IEEE Transactions on Geoscience and Remote Sensing, vol. 47, issue 1, pp. 123-129.

[5] Duerr, R., Yang, M., Lee, C. In press. Towards a standard archival format for Earth science data: Storing NASA ECS data using HDF5 Archival Information Packages (AIP). *Proceedings of the 2008 IEEE International Geoscience and Remote Sensing Symposium*.

[6] Yang, M., Duerr, R., Lee, C. 2009. Investigation of using HDF5 Archival Information Packages (AIP) to store NASA ECS data. Presented at the 89th AMS Annual Meeting, January 2009, Phoenix, Arizona.

[7] Cao, P. 2006. HDF5 Archival Information Package (AIP) – A METS Implementation. An HDF White Paper. http://www.hdfgroup.org/projects/hdf5_aip/hdf5_aip_wp.html

[8] PREMIS Editorial Committee. 2008. PREMIS Data Dictionary for Preservation Metadata, Version 2.0. http://www.loc.gov/standards/premis.

[9] Klein, L., Taaheri, A. 2007. HDF-EOS5 Data Model, File Format and Library. ESE-RFC-008v1.0 Recommended Standard. November 2007. http://www.esdswg.org/spg/rfc/ese-rfc-008/ESDS-RFC-008v1.0.pdf

[10] CGNS. 2009, CFD General Notation System Standard Interface Data Structures. Document Version 2.5.2. CGNS Version 2.5. http://www.grc.nasa.gov/WWW/cgns/sids/index.html.

[11] Dougherty, M., Folk, M., Zadok, E., Bernstein, H., Bernstein, F., Eliceiri. K, Benger, W., Best, C. 2009. Unifying biological image formats with hdf5. Communications of the ACM (CACM), 52(10):42–47, October 2009