Data Archival and Retrieval Enhancement (DARE) Metadata Modeling and its User Interface

Jason J. Hyon and Rosana Bisciotti Borgen

Jet Propulsion Laboratory
M.S. 525 - 389
4800 Oak Grove Dr.
Pasadena, CA 91109
(818) 306-6054, (818) 306-6929 FAX, [illegible]
(818) 306-6142, (818) 306-6929 FAX, [illegible]

Abstract

The Defense Nuclear Agency (DNA) has acquired terabytes of valuable data which need to be archived and effectively and efficiently distributed to the entire nuclear weapons effects community and others. NASA's Jet Propulsion Laboratory (JPL) is developing the Data Archival and Retrieval Enhancement (DARE) system for DNA to deliver information to Department of Defense (DoD) and its contractors via electronic remote access. The DARE system architecture is based on a client-server model implemented using the TCP/IP, HTTP and WAIS protocols. The pillar of this architecture is its metadata model. With this model, the architecture offers versatility and extendability to interface with any type of database or web indexing applications. This paper discusses the DARE metadata model and explains how it is used as a source for generating HyperText Markup Language (HTML) or Standard Generalized Markup Language (SGML) documents for access through web browsers such as Netscape.

Contents

DN P-96-170
96-S-1639

Data Archival and Retrieval Enhancement (DARE) Metadata Modeling and its User Interface

Jason J. Hyon and Rosana Bisciotti Borgen

Jet Propulsion laboratory
M.S. 52.5-3$'9
4800 *Oak Grove Dr.*
Pasadena, CA. 91109
(6'18) .306- 60S4, *(818)* 306-6929 *FAX, jason.j.hyon@jpl.nasa.gov*
(818) 306-6142, *(818)* 306-6929 *FAX, rosana.b.borgen@jpl.nasa.gov*

Abstract

The Defense Nuclear Agency (DNA) has acquired terabytes of valuable data which need to be archived and effectively and efficiently distributed to the entire nuclear weapons effects community and others. NASA's Jet Propulsion Laboratory (JPL) is developing the Data Archival and Retrieval Enhancement (DARE) system for DNA to deliver information to Department of Defense (DoD) and its contractors via electronic remote access. The DARE system architecture is based on a client server model implemented using the TCP/IP, HTTP and WAIS protocols. The pillar of this architecture is its metadata model. With this model, the architecture offers versatility and expandability to interface with any type. of database or web indexing applications. This paper describes the DARE metadata model and explains how it is used as a source for generating HyperText Markup Language (HTML) or Standard Generalized Markup Language (SGML) documents for access through web browsers such as Netscape.

Contents

Introduction

DNA has invested heavily in the acquisition of data unique to the world and in systems to analyze these data. Kelley [5] DARE's objective was to develop a state-of-the-art information system to preserve and facilitate the utilization of data generated in nuclear weapons effects tests as well as simulation experiments. JPL's experience in handling massive amounts of data from NASA's planetary missions, and their familiarity with state-of-the-art data archival and retrieval systems were crucial factors in their being selected to build DARE. DARE is being implemented in four incremental deliveries, each delivery designed and implemented using the Rapid Development Methodology. The metadata model and user interface which this paper describes will be available in the September 1996 delivery of DARE.

In order to ensure the longevity of the DARE archive, it is essential that the data and metadata be stored in formats which are open, standard, multiplatform, and supported by data conversion and validation tools. Formats which meet this criteria are the ones most likely to survive as hardware and software technologies evolve. From our past experiences with relational database systems, we realize that it is not easy to implement a system that is simple to learn, use, and implement. Thus, we took a different approach and designed a model with the following characteristics:

1. It handles multimedia - document, photo, numeric data, video, etc.
2. Once data is indexed, it has a fast and user friendly interface.
3. It is simple to learn, use, and implement.
4. It supports cross linking, simulating a relational database model.
5. It is easy to parse, manipulate, and validate.
6. It is open, so that it could be easily loaded into a DBMS later.
7. It protects data by allowing access to only qualified users.

We also adopted the TCP/IP, HTTP, and WAIS [ML]: tools for the bases of the DARE system, because the Wide Area information Serve.rs (WAIS) search engine and the HyperText Transfer Protocol (HTTP) satisfied the above requirements assuming that there is a reliable metadata model. The model is based on ASCII representations since WAIS can't index binary representations.

The Portable Data Specification (PDS) provides a complexity of hierarchical objects; however, treating metadata as an object gave us an easy way to handle different types of data. For example, a document object should carry a keyword describing number of pages while a photo object should have a keyword to describe the color of the photo. Thus, we decided to adopt the PDS with appropriate modifications for the World Wide Web (WWW) server environment. Most of the modifications involved simplifying the object structure, developing new objects for multimedia, and supporting system requirements. For example, since the DARE system is based on HTTP, we needed a new set of keywords to accommodate hypertext linking and direct access to data. Further, since these data are very sensitive, there need to be keywords to identify the sensitivity of the data and to protect it

The DARE metadata *model* consists of four elements: Data Dictionary, Directory, Guide, and Inventories. In this paper, we will explain how they interact with each other and how they are tied to the data. An individual metadata element is called a label and is based on the PDS and the Object Definition Language (ODL). We will describe the label structure for various metadata elements and state the reasons behind their design. A label consists of objects, and an object consists of keywords and values. Furthermore, those keywords and standard values are specified by the Data Dictionary which is also written in ODL.

In order to prepare the metadata and also to *present* them to either a user or a system uniformly, various filters and interfaces are necessary in order to conform to the standard. Moreover, it is also important to standardize data formats. We will discuss some of the tools that we have developed to prepare and to validate the metadata and data. As a guide to new developers who are designing a metadata model, we would also like to give some insights and to recount our experiences in preparing and converting metadata. In conclusion, we will illustrate the DARE system as whole and how metadata elements and data are interacting with each other in order to deliver information to its users

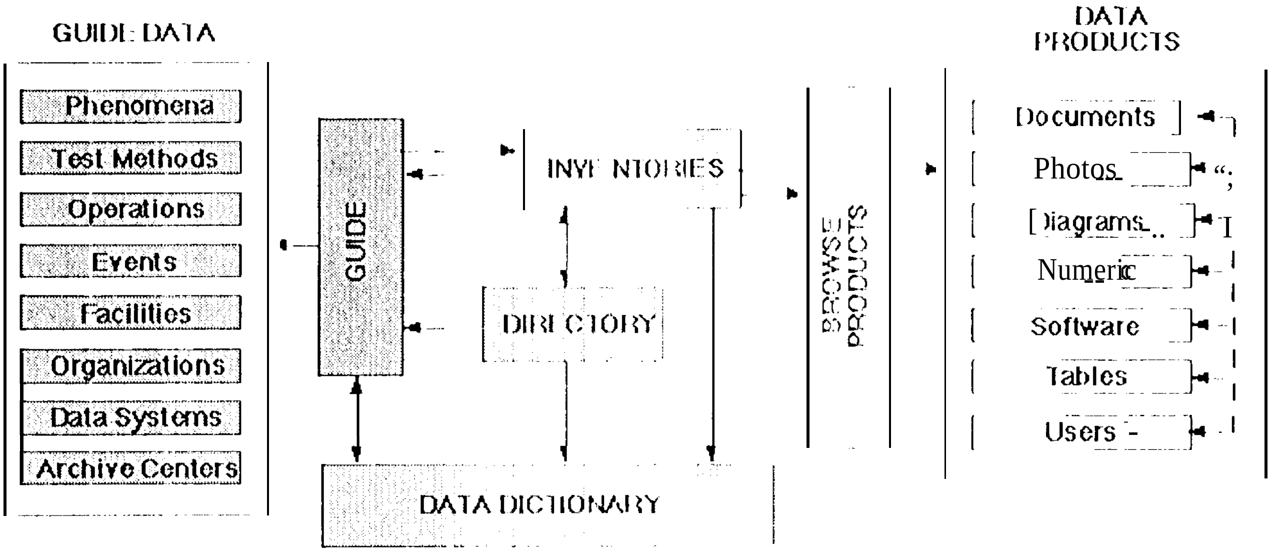

Figure 1. The DARE Metadata and Data Model

The Portable Data Specification (PDS) Standard

The PDS standard requires a label for each individual data product. [1] These labels are written in ODL and use a standard set of object definitions, keywords, and standard values for these keywords. These object definitions, keywords, and standard values are defined in the Data Dictionary which is also written in ODL.

The PDS standard requires a distinct data product label for each individual data product in order to describe the organization, format, content and quality of each data product. These labels are implemented as detached labels, which implies that the label is detached from its data and resides in a separate file which points to the data product.

Labels consist of a set of keywords and values separated by an equal sign. Keywords are composed of descriptor words (which describe what is being measured or presented in the value field) and class words. Keyword names are constructed using these components from left to right, from most specific (the leftmost component) to most generic (the rightmost component). Class words comprise the right most component in a keyword name. The class word identifies the basic information type of the data, where information type include both the data type (numeric, character, logical) and a size constraint. [1] For example, in the keyword INSTRUMENT-TYPE, INSTRUMENT is the descriptor word, and TYPE is the class word which implies that the values should be standardized and approved by DARE.

The use of a limited set of class words will:

1. Reduce the need for users and data processing software to access a data dictionary to parse, interpret, query or display values.
2. Add a greater level of structure and consistency to the nomenclature.
3. Constrain the selection and use of data values.
4. Promote automated operations such as validity checking.
5. Promote ***the*** development of intelligent software.

The PDS standard also provides rules and naming conventions for data sets, data set collections, volume names, file names, etc. It also defines the rules for constructing object and keyword names. The purpose of establishing a standard syntax is to create an environment where a number of individuals working independently are likely to select the identical keyword name for the same data item. This creates consistency throughout the system and correlative analysis is better supported.

The Object Definition Language

The ODL is a simple human readable language based on the concept of objects and "keyword = value" statements. [1] For example, AUTHOR_NAME = "Doe, John S.", "Doe, Joe S."}, This language is used both to label data products and to collect metadata for loading into DARE. The character set of ODL is the International Standards Organization's ISO 646 character set (i.e., the ASCII character set). ASCII labels can be read by users easily, modified by simple editors, and parsed by DARE tools.

Metadata Elements

The DARE system consists of Data Dictionary, Directory, Guide, and Inventories. The DARE metadata model was developed to support the interaction with metadata elements and to tie these elements to data themselves. Metadata provides descriptive information about components of DARE such as data sets (i.e., group of data) or data products. These descriptive entries are stored in the four elements within DARE. Inventories describe individual data products (e.g., *documents, photos, and videos*).

The metadata for the Data Dictionary, Directory, and Guide largely consists of descriptive text. The metadata for a data product includes all the information that is needed to uniquely identify, locate, and describe that particular product. The metadata could be categorized into three major parts: the description of the data, the format of the data, and the classification of the data. The description should contain information such as author, review notes, organizations, and date it was created. The format identifies what format was used to archive and to represent the data to the user. The Classification describes the DoD classification of the data. Detailed descriptions will be illustrated in the later sections.

For most categories of information *(Directory, Guide, Inventories)* there is a standard set of required descriptive keywords; these. categories of information are organized into objects. DARE provides an empty label, called a *template* for each major information category or object. This template is just a label that hasn't been filled out yet which contains each of the allowed keywords. Further, the descriptions of allowed keywords in each object and allowed values for each keywords are specified in the Data Dictionary.

Data Dictionary

The primary purpose of the Data Dictionary is to allow members of the DNA community to benefit from standards work done in the area of data product description. It also serves as an interface agreement between DARE and data providers in order to describe data in uniform ways with objects and keywords, and also values. To define an object for a specific category takes a lot of research and interactions with the community. An object should convey a full description of a given category with a define.d set of keywords. Thus, the selected keywords should contain the following properties:

1. Contain all the information which users can search.
2. Once retrieved, deliver information with more descriptive sentences.
3. Provide information to the system in order to process the information

The basic rule of a thumb for selecting keywords is that, if a value needs to be standardized or commonly searched, or used by the system software, a keyword is needed. Any other values should be described in free text fields. Therefore, these objects and keywords have to go through many iterations to achieve this goal. Some keywords require values to be standardized; these are called standard values. This needs to be done in order for users to search values in a simple and uniform way. For example, a user wants to search an organization name with "DNA" or "DNA - Alexandria". If the value of the keyword, Organization name, is "Defense Nuclear Agency" without its acronym, he won't be able to find an entry. By specifying official organization names in the Data Dictionary, the user can look up the Data Dictionary for its correct spelling and retrieve the desired data.

OBJECT = ELEMENT_DEFINITION
NAME = PHENOMENA_NAME
REVIEW_NOTE = "N/A"
REVISION_NOTE = "1994-06-09, K.Marsh, JPL Initial DD entry"
DESCRIPTION = "Provides the name of the phenomena."
GENERAL_CLASSIFICATION_TYPE = METADATA
VIEW_DEPENDENCY_FLAG = NO
VIEW_NAME = "N/A"
GENERAL_DATA_TYPE = CHARACTER
UNIT = "N/A"
VALID_MAXIMUM = "N/A"
VALID_MINIMUM = "N/A"
MAXIMUM_LENGTH = 20
MINIMUM_LENGTH = "N/A"
STANDARD_VALUE_SET = {AIRBLAST, CRATERING, EJECTA, FIREBALL, ... *for illustration purposes only*}
END_OBJECT = ELEMENT_DEFINITION
END

FIGURE 2.1 Data Dictionary Entry

The Data Dictionary is constantly modified to add new objects, keywords and values. Sometimes changes to the Data Dictionary require updating existing labels in the system. This can be a time consuming process. Unless there is a very important reason to change them, existing objects, keywords, and standard values should be stable.

Entries in the Data Dictionary themselves are described by the Data Dictionary object as shown in the figure 2. This Data Dictionary object is for the PHENOMENA_NAME keyword. STANDARD_VALUE_SET specifies the values for the keyword. UNIT specifies the unit of the value, and length keywords indicates the length of the value. The GENERAL_CLASSIFICATION_TYPE keyword identifies whether this keyword is used for description and classification (i.e., METADATA), format provided by the data provider (i.e., STRUCTURE), or format provided by the DARE system (i.e., SYSTEM). The VIEW_DEPENDENCY_FLAG and VIEW_NAME keywords identify which *views* this keyword supports. More on views will be discussed under Inventories.

Directory

The purpose of the Directory is to give users a high level overview of DNA's data and its contents and archive location. The data sets described in the Directory may be archived electronically in the DARE system or may exist in other archive centers either in hardcopy or electronic form.

A Directory entry describes a data set - a group of data. The organization of data can be quite complex. Producing an archive quality data set means including what is needed to understand and utilize the data.

The relationship between the data and everything else can be complex. In addition, data sets many be grouped together into larger entities to serve a particular scientific objective. Thus, an archive data set is defined as the accumulation of data products, supplemental data, software, and documentation, that will completely document and support the use of those data products. A data set can be part Of an other data set. A simpler version of a data set could be a document collection from someone's file cabinet, for instance.

Besides other Directory keywords, TITLE in the Directory object is the key to tie all the objects that arc related to this data set described by the Directory. As each object shares the same value for DIRECTORY_TITLE, the DARE software can build a link between all the related objects and TITLE from a Directory entry.

OBJECT = DIRECTORY
Other Directory ***keywords Plus***
TITLE = "This is a test title for a data collection (U)"
END_OBJECT = DIRECTORY
END

FIGURE 3. Directory Label

Guide

Guide entries describe tbc domain of nuclear weapons effects testing (e.g., phenomena, test methods, operations, events, facilities, organizations, data systems, etc.). This information can be very helpful for someone who has little. knowledge about nuclear weapons effects testing. Guide entries fall into several categoric.s such as EVENT_NAME, PHENOMENA_NAME, ORGANIZATION_NAME, etc. These keywords arc also in the objects. The DARE software compares the standard values for these keywords in each object with the values in the Guide entries. If it finds the matching value, it then builds a link to the Guide entry and allow users to access the information. This process mimics a relational data manager. information is stored in the Guide, and other data entries access this information through dynamic links created by the DARE software. Figure 4 is a typical example of a Guide label.

OBJECT = GUIDE	
Other Guide keywords Plus	
One of the Guide Category Keywords	ORGANIZATION_NAME, PHENOMENA_NAME, TEST_TYPE, TEST_METHOD_TYPE, LOCATION_ID, FACILITY_NAME, OPERATION_NAME, EVENT_NAME, EXPERIMENT_NAME, SENSOR_NAME, INSTRUMENT_TYPE, TARGET_NAME, TARGET_TYPE, ARCHIVE_CENTER_NAME, DATA_SYSTEM_NAME
BROWSE_FILE_NAME = "test.lltn~"	
END_OBJECT = GUIDE	
END	

FIGURE 4. Guide Label

For example, if ORGANIZATION_NAME = "Defense Nuclear Agency (DNA) (Alexandria, VA)", any objects which share this value for ORGANIZATION_NAME will be linked to this Guide entry. Once it is linked, the detailed descriptions of the value will be stored in HTML format. This file is pointed to by the BROWSE_FILE_NAME value.

Inventories

Metadata models that had to support correlative science resulted in complex models. For example, an instrument model could include complex relationships between events, number of experiments in an event, measured parameters, and physical components of the instrument. Hughes[4] We realized that there is much more interest in metadata that supplied detailed processing information or reviews, much of which was lost trying to fit the data into a fixed model. With the help of the WAIS full-text indexing capability, we were able to supply better textual descriptions and to offer a search capability on those descriptions. Thus, the metadata for an Inventory uses a minimal set of keywords to describe data and stores textual descriptions in either NOTE or DESCRIPTION keywords.

Usually, there is a one-to-one relationship between data and metadata. However, it is not effective to have a label for every memo or notebook from someone's file cabinet. By using a localized WAIS index we also have developed a feature to support a collection of various types of documents such as memos, letters, notebooks, etc. It is called *Document Aggregation*. AGGREGATION_FLAG and DOCUMENT_TYPE in the Document object indicates whether this metadata represents a single document or a collection of documents; thus, the DARE software can behave appropriately to deliver information to the user. This scheme will also be useful for other types of Inventories.

Views

While working with Numeric data, we found that we couldn't describe data with a limited set of keywords. Data acquired for the study of different nuclear weapons effects phenomena required some unique keywords to search it precisely. Further, looking through the Data Dictionary to find a standard value for a specific keyword could be very time consuming if most of the values are not applicable to a specific nuclear weapon effect; we call this way of looking at a specific nuclear weapon effect a view. Thus, view specific objects were developed to represent a view and a set of standard values for view specific keywords.

View specific objects use a set of keywords from a Generic object, in this case the Numeric object. For instance, the AIRBLAST_NUMERIC view specific object uses airblast specific keywords from the NUMERIC object. As we add more view specific objects, new keywords needed for the new view specific object are added to the generic NUMERIC object. Thus, the DARE software treats these view specific objects as a NUMERIC object with a set of view specific keywords for structured field searches. Further, a use of VIEW_NAME and VIEW_DEPENDENCY_FLAG in the Data Dictionary allows the DARE software to present view specific standard values in the Data Dictionary. In the Data Dictionary, if VIEW_DEPENDENCY_FLAG = YES for a specific keyword, VIEW_NAME values are provided in a sequence shown in the following example:

NAME = DATA_TYPE
VIEW_DEPENDENCY_FLAG = YES
VIEW_NAME = (GLOBAL, AIRBLAST, CRATER_EJECTA)
STANDARD_VALUE_SET = ((D,E,F,X,Y,Z),(D,E,F),(X,Y,Z))
Plus the rest of the Data Dictionary Keywords

Figure 5. A Data Dictionary Example

In the above example, the keyword DATA_TYPE has three views GLOBAL, AIRBLAST, and CRATER_EJECTA. If a user wants only the AIRBLAST view of the Data Dictionary, the DARE software extracts standard values of D, E, and F from the STANDARD_VALUE_SET. The GLOBAL view is the union of all the views.

Data Formats

Not only the metadata format but also the data formats had to be standardized in order to process and to deliver the data effectively. We used the following criteria in selecting the DARE data formats - TIFF, GIF, PDF, JPEG, MPEG, and QUICKTIME :

- Easily accessible by public domain software
- Contain a good description of data.
- Support various compression schemes (e.g., LZW, CCITT, JPEG, MPEG, etc.).
- Extensible and interchangeable.
- Simple to implement,
- Minimal amount of descriptor overhead required by the file format.
- Small amount of data access overhead

DARE supports the following Inventories:

- Document
- Photo
- Diagram
- Numeric Data
- Software
- Table
- Video

For example, the standard DARE archive format for scanned images is Tagged Image File Format (TIFF). There are two exceptions: images which can be viewed at a low resolution may be submitted in Graphical Image Format (GIF) and images which are in-line with the document text must be submitted in GIF. DARE also creates browse versions of the archive data for display by the DARE system. For files which are viewed as a series such as pages of documents or a sequence of photos, the TIFF files are converted to Adobe Portable Document Format (PDF). Thumbnail and browse versions of color and grayscale images (e.g., photos) are created in the GIF and JPEG File Interchange Format, respectively. Thumbnail and browse versions of black and white images (e.g., diagrams or plot of numeric data) are created in GIF.

Scanned documents, photos, videos, and diagrams require different resolutions and compression schemes. The DARE Data Submittal Manual (DSM) specifies the formats and recommends standard practices for preparing data.

Tools for Metadata and Data

To expedite the data ingestion process, we developed tools to generate, validate, and process metadata and data.

Generation

Label Editor Tool allows users to create metadata by allowing them to choose values from a list of standard values in a scroll bOX. The tool takes the Data Dictionary as an input, SO users can select a desired object and generate metadata labels easily. Other tools can generate labels from Tab separated ASCII output of a simple database manager (e.g., FoxPro, DBASE III), or a sophisticated on-line relational data management system (e.g., SYBASE, ORACLE). Or a user can download a template - label without values filled in from the DARE system, and use an editor to generate labels. For data generation, there are many ways to create the data in accordance with the DARE standards.

Validation

lvtool and *slvtool* are tools which read DARE labels and perform the following types of error checking:

- ODL Language Checking - ODL language checking detects errors in the usage of ODL, such as missing quote marks, invalid characters in names, other syntax errors, etc.
- Data Dictionary Checking - The keywords and objects used in DARE labels follow certain DARE standards, as well as rules specified by the label designer. Data Dictionary checking determines when these rules have been broken by comparing each label to the Data Dictionary.

Other tools include a SGML/HTML syntax checker. The DARE software that generates browse data also checks the formats of the archive data.

Process

Many tools have been written to process various types of archive data to generate browse data. As the DARE system relies on network transfer for delivering data to users, it is necessary to have browse data files as small as possible. Thus, archive data are converted to various formats such as GIF, PDF, JPEG, MPEG, and QUICKTIME. Further, runtime software can process metadata to generate SGML/HTML versions of the metadata.

System Architecture

Client-Server Model

The DARE system architecture is based on a client/server model. This architecture offers the following advantages:

1. Users can use existing desktop computers with little or no additional expense for hardware or software.
2. As the system grows, additional DARE servers can be added anywhere on the network.
3. Servers for classified and unclassified information can be placed on different networks.
4. Links to other database servers and analysis systems are relatively easy to implement.

DARE's technical approach is to maximize the use of Commercial Off The Shelf (COTS) hardware and software and Non-Developmental Item (NDI) software available as freeware, shareware, or in the public domain. Custom software development is minimized and is typically written to integrate COTS and NDI software components.

DARE uses open system technologies and standards. A few examples of open system technologies adopted by DARE are: SCSI interfaces for hardware archive storage devices; the UNIX operating system for the serve.r; TCP/IP network protocol, the free WAIS-sf (structured field WAIS) search engine and the HyperText Transfer Protocol (HTTP). World Wide Web (WWW) technologies provide the powerful user interface and server software capabilities required by DARE.

Custom Software

The custom DARE software builds HyperText Markup Language (HTML) pages for browsing data inventories, full-text indexes and structured field indexes. These HTML pages produced both at setup and runtime are what makes DARE unique. All the HTML pages are generated by custom Perl scripts while the underlying WAIS-sf search engine generates the indexes.

The HTML pages created by setup are static pages used for keyword/value searches. The HTML pages created by runtime are dynamic pages used for full-text and structured field search results as well as the display of Inventory information which is hyper-linked to ancillary information in the Guide and the Data Dictionary. All of these tasks are easily attainable due to the underlying structure of the DARE labels. The DARE labels could also be easily loaded into a Database Management System (DBMS).

The DARE custom software is composed of three main subsystems:

- Configure: Prepares the DARE software environment using configuration files to identify data inventories and supporting information.
- Setup and Setup Sequencer: Creates the static keyword/value HTML pages, WAIS indexes, and Perl arrays needed for runtime execution. Setup is run on whole inventories while Setup Sequencer is run when data products are added, updated or deleted.
- Runtime: Executes and summarizes WAIS searches, displays information about a single Inventory or supporting information items, and dynamically tags the displayed label.

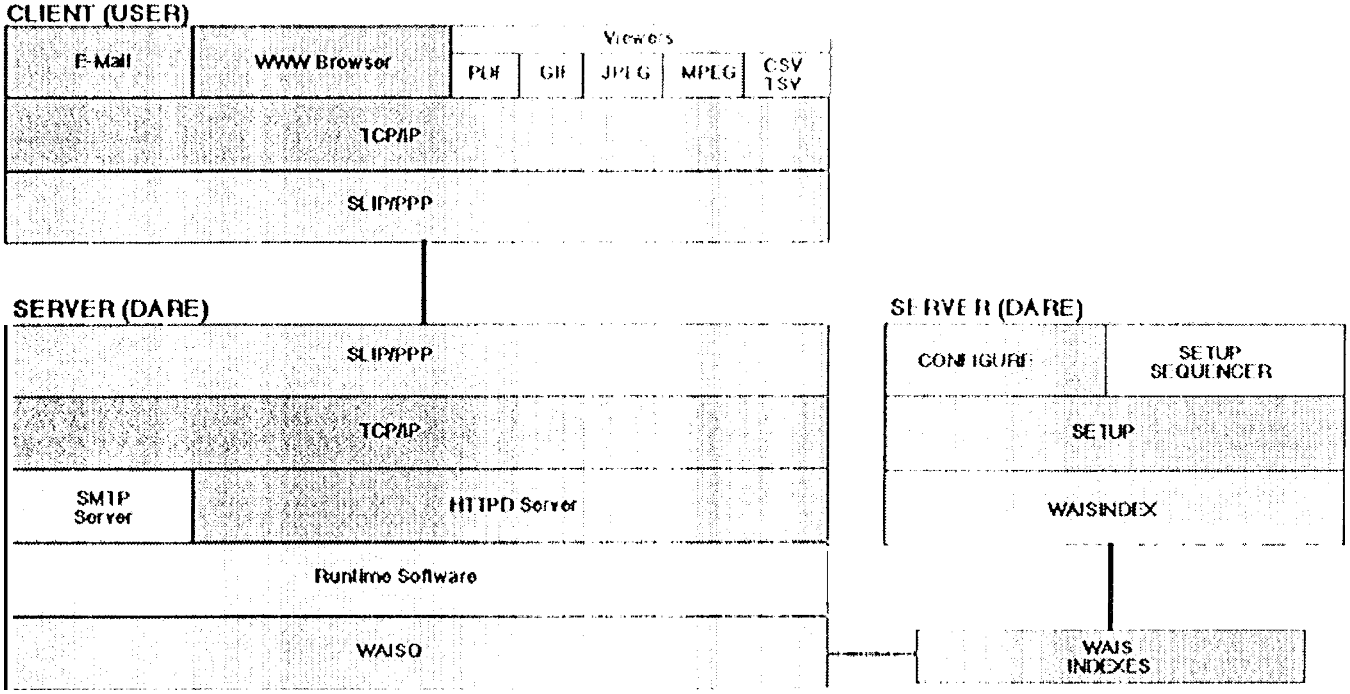

Figure 6. The DARE System Architecture

Search Modes and User Interface

There are basically two ways of searching

- By conducting a full-text search of all DARE metadata elements (Data Dictionary, Directory, Guide, Inventories).
- By selecting one metadata element (Data Dictionary, Directory, Guide, Inventory) as the starting point of the search.

Full-text search of all Metadata Elements

A full-text search will match search words that are present anywhere in a label in all metadata elements. When a full-text search is conducted the following pages are created by the runtime software and presented to the user:

1. An HTML full-text summary page . This full text summary page indicates which metadata elements have data that match the search criteria. The user can then select the category of interest by clicking on it.
2. An HTML title page. This title page lists the titles of the data products for the selected metadata element that matched the search criteria. The user can then select a title of interest by clicking on it.
3. An HTML layout page. The runtime software uses the label to generate the layout page. This layout page, generated from the label by the runtime software, summarizes the label information:
   - Provides the title.
   - Provides the classification.

o Presents the user with viewing options for the data product - options vary from one inventory to another. The options for the photo Inventory include viewing the online or full-resolution photo and browsing the archive directory while the options for the table Inventory include viewing the online table, downloading the archive table and browsing the archive directory.
0 Provides links to related information in the Guide and Directory.
0 Provides the option of displaying the label.

4. When the option of displaying a label is selected, an HTML tagged label is presented to the user. The runtime software links the tagged label to the Data Dictionary.

Figure 7 shows an example of the HTML title, layout and tagged label pages.

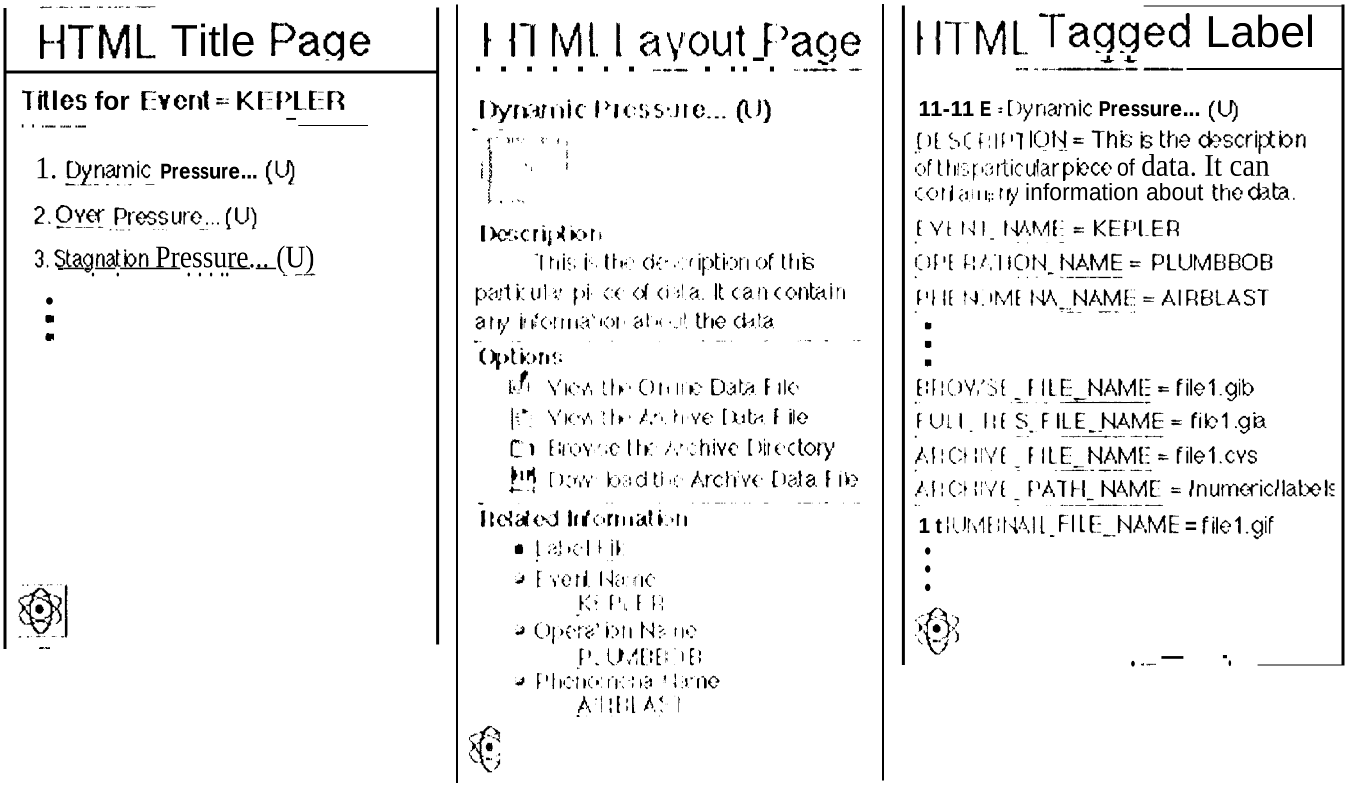

Figure ?. he DARE User Interface

Figure 7. The DARE User Interface

Single Metadata Element Searches

When a search is started from a metadata element three different types of searches are available:

1. Keyword/value search.
2. Full-text search.
3. Structured field search.

Kqww-d/Value Search

The keyword/value. search pages are generated at setup time by parsing the labels and organizing the keywords and values into HTML files. A user not familiar with the DARI data may start a search this way, This search is also valuable because it presents the user with all the titles that contain a certain keyword/value combination. Note that WAIS searches are limited to a maximum number of hits and some data may not be easily retrieved. For instance in DARE the maximum number of hits for a WAIS search is set to 1000. The following pages are presented 10 the user when a keyword/value search is Conducted:

1. An HTML keyword list. This list is metadata element dependent. A user can select a keyword by clicking on it.
2. An HTML list of values for the selected keyword. The user can then select a value by clicking on it.
3. An HTML list of titles for the keyword/value selected. The user can then select a title by clicking on it.
4. An HTML layout page as described above.
5. An HTML tagged label as described above.

Full-Text Search

The full-text search for an individual metadata element only differs from the all metadata element search in that the summary page is not produced When a full-text search is conducted the following pages arc created by the runtime software and presented to the user, these pages are the same as the ones described above:

1. The HTML title page,
2. The HTML layout page,
3. The HTML tagged label,

Structured Field Search

A structured field search allows the user to specify a value (or range of values in case of numeric and date keywords) for one or more keywords for a given search. The keywords available for selection are metadata element dependent. When a structured field search is conducted, the same pages described in the full-text search are created by the runtime software and presented to the user.

Summary

The DARE metadata model and the standards implemented allow for easy portability to other standards and indexing systems. For example, DARE is now using HTML for its browse pages and display formats. With minor changes to the code these HTML pages could be converted to SGML when the community feels that SGML is the standard to be adopted.

Because of the modularity of the DARE software and because the DARE labels are in ASCII format the current search engine (free WAIS-sf) could be replaced with minor impact to the underlying code. Furthermore, DARE has not selected to implement DBMS; however, the structure of its metadata will make it very easy to port to a relational database

'The DARE development staff recognizes that there are a lot of processing steps between selecting a data format and making the information available to users of the DARE system. Each time a new type of data is planned for incorporation into DARE, a data engineering process is required. During this process, the Data Engineer selects a data format and works with the data providers and users to determine the metadata requirements for the data." Kelley [5] We strongly recommend that people take the time to design metadata for new data type.s since only a good metadata model can support a good retrieval system.

Acknowledgments

This effort is sponsored by the Defense Nuclear Agency under DNA IACRO 963028. The work described in this paper was carried out by the Jet Propulsion Laboratory, California institute of Technology, under a contract with the National Aeronautics and Space Administration. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise dots not constitute or imply its endorsement by the 11. S. government or the Jet Propulsion Laboratory.

We would like to thank the DARE development staff - Carol Miller, Keevin Fisher, Mark Takacs, Kristy Kawasaki, Patrick Murphy, and Oscar DeMartino for their support in writing this paper.

References

1. PDS Standard Reference, JPL Internal Document D-7669, Jet Propulsion Laboratory, November 1992
2. "DARE Data Dictionary", JPL Internal Document, Jet Propulsion Laboratory, March 1996
3. Hyon, J., "DARE Data Submittal Manual", JPL Internal Document, Jet Propulsion Laboratory, October 1995
4. Hughes, J. S., Bernath, A. M., "Planetary Data System Data Model as Applied to WWW interfaces (WEBCAT)", JPL & NASA Case No NPO 19667, October 1995
5. Dr. Kelley, C. S., Miller, C. L., Walsh, G., "Data Archival and Retrieval Enhancement (DARE)", Science & Technology Digest, Volume 2, September 1995

Data Archival and Retrieval Enhancement (DARE) Metadata Modeling and its User Interface

Jason J. Hyon and Rosana Bisciotti Borgen

Jet Propulsion Laboratory
M.S. 525 - 389
4800 Oak Grove Dr.
Pasadena, CA. 91109
(818) 306-6054, (818) 306-6929 FAX, jason.j.hyon@jpl.nasa.gov
(818) 306-6142, (818) 306-6929 FAX, rosana.b.borgen@jpl.nasa.gov

Abstract

The Defense Nuclear Agency (DNA) has acquired terabytes of valuable data which need to be archived and effectively and efficiently distributed to the entire nuclear weapons effects community and others. NASA's Jet Propulsion Laboratory (JPL) is developing the Data Archival and Retrieval Enhancement (DARE) system for DNA to deliver information to Department of Defense (DoD) and its contractors via electronic remote access. The DARE system architecture is based on a client-server model implemented using the TCP/IP, HTTP and WAIS protocols. The pillar of this architecture is its metadata model. With this model, the architecture offer's versatility and expandability to interface with any type of database or web indexing applications. This paper describes the DARE metadata model and explains how it is used as a source for generating HyperText Markup Language (HTML) or Standard Generalized Markup . Language (SGML) documents for access through web browsers such as Netscape.

Introduction

DNA has invested heavily in the acquisition of data unique to the world and in systems to analyze these data. Kelley [5] DARE's objective was to develop a state-of-the-art information system to preserve and facilitate the utilization of data generated in nuclear weapons effects tests as well as simulation experiments. JPL's experience in handling massive amounts of data from NASA's planetary missions, and their familiarity with stak-of-the-;,tl t data archival and retrieval systems were crucial factors in their being selected to build DARE. DARE is being implemented in four incremental deliveries, each delivery designed and implemented using the Rapid Development Methodology. The metadata model and user interface which this paper describes will be available in the September 1996 delivery of DARE.

in order to ensure tile longevity of the DARE archive, it is essential that the data and metadata be stored in formats which arc open, standard, multiplatform and supported by data conversion and validation tools. Formats which meet this criteria are the ones most likely to survive as hardware and software technologies evolve. From our past experiences with relational database systems, wc realize that it is not easy 10 implement a system that is simple to learn, use, and implement. Thus, we took a different approach and designed a model with the following characteristics:

1. It handles multimedia - document, photo numeric data, video, etc
2. Once data is indexed, it has a fast and user friendly interface.
3. It is simple to learn, use, and implement
4. It supports cross linking, simulating a relational database model.
5. It is easy to parse, manipulate, and validate
6. It is open, so that it could be easily loaded into a DBMS later.
7. It protects data by allowing access to only qualified users.

We also adopted the '1'(3'/11', 11"1-1'1', and WAIS protocols for the bases of the DARE system, because the Wide Area Information Servers (WAIS) search engine and the HyperText Transfer Protocol (HTTP) satisfied the above requirements assuming that there is a reliable metadata model. The model is based on ASCII representations since WAIS can't index binary representations.

The Portable Data Specification (PDS) provides a complexity of hierarchical objects; however, treating metadata as an object gave us an easy way to handle different types of data For example, a document object should carry a keyword describing number of pages while a photo object SI1OU1C1 have a keyword to describe the color of the photo. Thus, we decided to adopt the PDS with appropriate modifications for the World Wide Web (WWW) server environment Most of the modifications involved simplifying the object structure, developing new objects for multimedia, and supporting system requirements. For example, since the DARE system is based on HTTP, we needed a new set of keywords to accommodate hypertext linking and direct access to data Further, since [tic.w data are very sensitive, there **need to be keywords to** identify the sensitivity of the data and to protect it.

The DARE metadata *model* consists of four elements - Data Dictionary, Directory, Guide, and Inventories. in this paper, we will explain how they interact with each other and how they are tied to the data. An individual metadata element is called a label and is based on the PDS and the Object Definition Language (ODL). We will describe the label structure for various metadata elements and state the reasons behind their design. A label consists of objects, and an object consists of keywords and values. Furthermore, those keywords and standard values are specified by the Data Dictionary which is also written in ODL.

in order to prepare the metadata and also to *present* them to either a user or a system uniformly, various filters and interfaces are necessary in order to conform to the standard. Moreover, it is also important to standardize data formats. We will discuss some of the tools that we have developed to prepare and to validate. the metadata and data. As a guide to new developers who are designing a metadata model, we would also like to give. some insights and to recount our experiences in preparing and converting metadat a. in conclusion, we will illustrate the DARE system as whole and how metadata elements and data arc interacting with each other in order to deliver information to its users

Figure 1. The DARE Metadata and Data Model

The Portable Data Specification (PDS) Standard

The PDS standard requires a label for each individual data product[1] These labels arc written in ODL and use a standard set of object definitions, keywords and standard values for these keywords. These object definitions, keywords, and standard values are defined in the Data Dictionary which is also

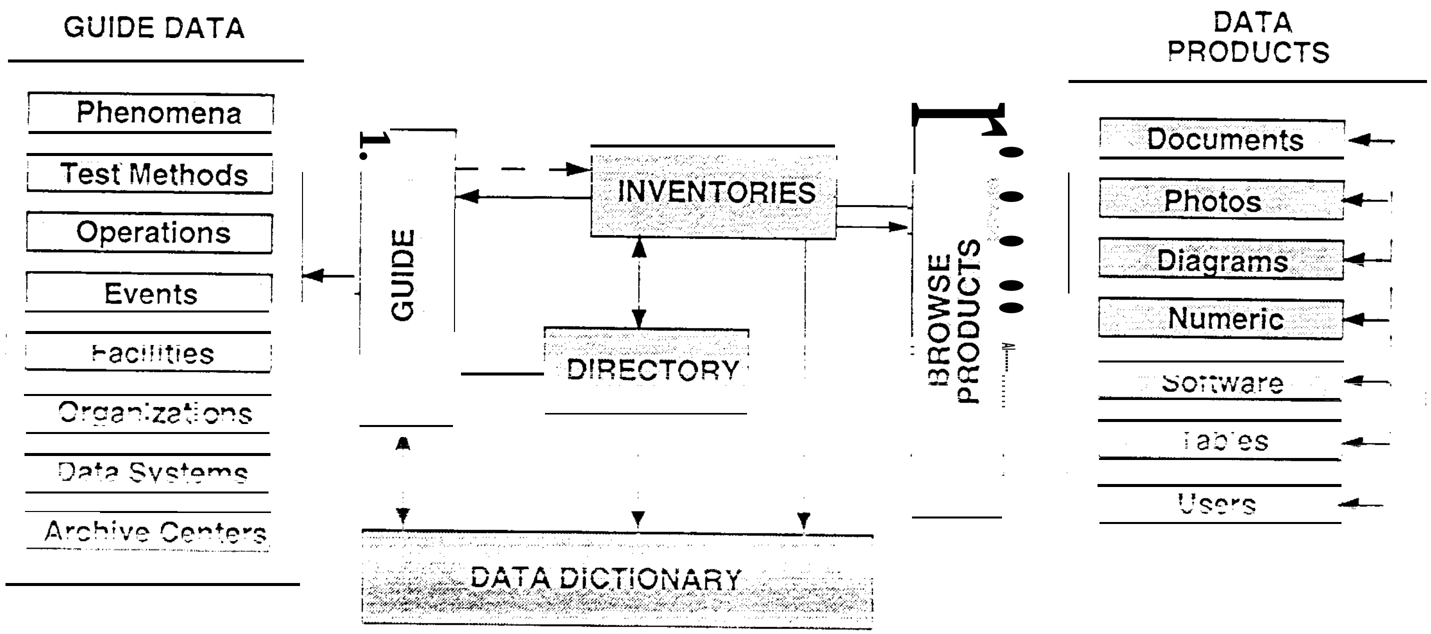

Figure 1. The DARE Metadata and Data Model

written in 01)1.,

The PDS standard requires a distinct data product label for each individual data product in order 10 describe the organization, format, content and quality of each data product. These labels are implemented as detached labels, which implies that the label is detached from its data and resides in a separate file which points to the data product

Labels consist of a set of keywords and values separated by an equal sign. K(:J'welds are composed of descriptor words (which describe what is being measured or presented in the value field) and class words. Keyword names arc constructed using these components from left to right, from most specific (the leftmost component) [0 most generic (the right most component). Class words comprise the right most component in a keyword name. The class word identifies the basic information type of the data, where information type include both the data type (numeric, character, logical) and a size constraint. [1] For example, in the. keyword INSTRUMENT_TYPE, INSTRUMENT is the descriptor word, and TYPE is the class word which implies that the values should be standardized and approved by DARE.

The List of a limited set of class words will:

1. Reduce the need for users and data processing software to access a data dictionary to parse, interpret, query or display values.
2. Add a greater level Of structure and consistency to the nomenclature.
3. Constrain the selection and use of data values.
4. Promote automated operations such as validity checking.
5. Promote the development of intelligent software.

The 1)l)S standard also provides rules and naming conventions for data sets, data set collections, volume names, file names, etc. It also defines the rules for constructing object and keyword names. The purpose of establishing a standard syntax is to create an environment where a number of individuals working independently are likely to select the identical keyword name for the same data item. This creates consistency throughout the system and correlative analysis is better supported

The Object Definition Language

The ODL is a simple human readable language based on the concept of objects and "keyword = value" statements.[1] For example, AUTHOR_NAME={"Doe, John S.", "Doe, Joe S."}. This language is used both to label data products and to collect metadata for loading into DARE. The character set of ODL is the International Standards Organization's ISO 646 character set (i.e., the ASCII character set). ASCII labels can be react by users easily, modified by simple editors, and parsed by DARE tools.

Metadata Elements

The DARE system consists of Data Dictionary, Directory, Guide, and Inventories. The DARE metadata model was developed to support the interaction with metadata elements and to tie these elements to data themselves. Metadata provides descriptive information about components of DARE such as data sets (i.e., group of data) or data products. These descriptive entries are stored in the four elements within DARE. Inventories describe individual data products (e.g., *documents, photos, and videos*).

The metadata for the Data Dictionary, Directory, and Guide largely consists of descriptive text. The metadata for a data product includes all the information that is needed to uniquely identify, locate, and describe that particular product. The metadata could be categorized into three major parts: the description of the data, the format of the data, and the classification of the data. The description should contain information such as author, review notes, organizations, and date it was created. The format identifies what format was used to archive and to represent the data to the user. The Classification describes the DoD classification of the data. Detailed descriptions will be illustrated in the later sections

For most categories of information *(Directory, Guide, Inventories)* there is a standard set of required descriptive keywords; these categories of information are organized into objects. DARE provides an empty label, called a *template* for each major information category or object. This template is just a label that hasn't been filled out yet which contains each of the allowed keywords. Further, the descriptions of allowed keywords in each object and allowed values for each keywords are specified in the Data Dictionary.

Data Dictionary

The primary purpose of the Data Dictionary is to allow members of the DNA community to benefit from standards work done in the area of data product description. It also serves as an interface agreement between DARE and data providers in order to describe data in uniform ways with objects and keywords, and also values. To define an object for a specific category takes a lot of research and interactions with the community. An object should convey a full description of a given category with a defined set of keywords. Thus, the selected keywords should contain the following properties:

1. Contain all the information which users can search.
2. Once retrieved, deliver information with more descriptive sentences.
3. Provide information to the system in order to process the information.

The basic rule of a thumb for selecting keywords is that, if a value needs to be standardized or commonly searched, or used by the system software, a keyword is needed. Any other values should be described in free text fields. Therefore, these objects and keywords have to go through many iterations to achieve this goal. Some keywords require values to be standardized; these are called standard values. This needs to be done in order for users to search values in a simple and uniform way. For example, a user wants to search an organization name with "DNA" or "DNA Alexandria". If the value of the keyword, Organization name, is "Defense Nuclear Agency" without its acronym, he won't be able to find an entry. By specifying official organization names in the Data Dictionary, the user can look up the Data Dictionary for its correct spelling and [illegible] the desired data.

The Data Dictionary is constantly modified to add new objects, keywords and values. Sometimes changes to the Data Dictionary require updating existing labels in the system. This can be a time consuming process. Unless there is a very important reason to change them, existing objects, keywords, and standard values should be stable.

Entries in the Data Dictionary themselves are described by the Data Dictionary object as shown in the figure 2. This Data Dictionary object is for the PHENOMENA NAME keyword. STANDARD_VALUE_SET specifies the values for the keyword UNIT specifies the unit of the value, and length keywords indicates the length of the value. The GENERAL_CLASSIFICATION_TYPE keyword identifies whether this keyword is used for description and classification (i.e., METADATA), format provided by the data provider (i.e., STRUCTURE), or format provided by the DARE system (i.e., SYSTEM). The VIEW_DEPENDENCY_FLAG and VIEW_NAME keywords identify which *views* this keyword supports. More on views will be discussed under Inventories.

```
OBJECT = ELEMENT_DEFINITION
NAME = PHENOMENA_NAME
REVIEW_NOTE = "N/A"
REVISION_NOTE = "1994-06-09, K. Marski, JPL, Initial DD entry"
DESCRIPTION = "provides the name of the phenomena"
GENERAL_CLASSIFICATION = METADATA
VIEW_DEPENDENCY_FLAG = NO
VIEW_NAME = "N/A"
GENERAL_DATATYPE = CHARACTER
UNIT = "N/A"
VALID_MAXIMUM = "N/A"
VALID_MINIMUM = "N/A"
MAXIMUM_LENGTH = 20
MINIMUM_LENGTH = "N/A"
STANDARD_VALUE_SET = {AIRBLAST, CRATERING, EJECTA, FIREBALL, ... for illustration purposes only}
END_OBJECT = ELEMENT_DEFINITION
END
```

FIGURE 2. Data Dictionary Label

Directory

The purpose of the Directory is to give users a high-level overview of DNA's data and its contents and archive location. The data sets described in the Directory may be archived electronically in the DARE system or may exist in other archive centers either in hardcopy or electronic form.

A Directory entry describes a data set a group of data. The organization of data can be quite complex. Producing an archive quality data set means including what is needed to understand and utilize the data. The relationship between the data and everything else can be complex. In addition, data sets many be grouped together into larger entities to serve a particular scientific objective. Thus, an archive data set is defined as the accumulation of data products, supplemental data, software, and documentation, that will completely document and support the use of those data products. A data set can be part of another data set. A simpler version of a data set could be a document collection from someone's file cabinet, for instance.

Besides other Directory keywords, TITLE in the Directory object is the key to tie all the objects that are related to this data set described by the Directory. As each object shares the same value for DIRECTORY_TITLE, the DARE software can build a link between all the related objects and TITLE from a Directory entry.

OBJECT = DIRECTORY
Other Directory keywords Plus
TITLE = "This is a test title for a data collection (1)"
END_OBJECT = DIRECTORY
END

FIGURE 3. Directory Label

Guide

Guide entries describe the domain of nuclear weapons effects testing (e.g., phenomena, test methods, operations, events, facilities, organizations, data systems, etc.). This information can be very helpful for someone who has little knowledge about nuclear weapons effects testing. Guide entries fall into several categories such as EVENT_NAME, PHENOMENA_NAME, ORGANIZATION_NAME, etc. These keywords are also in the objects. The DARE software compares the standard values for these keywords in each object with the values in the Guide entries. If it finds the matching value, it then builds a link to the Guide-entry and allow users to access the information. This process mimics a relational data manager. Information is stored in the Guide, and other data entries access this information through dynamic links created by the DARE software. Figure 4 is a typical example of a Guide label.

OBJECT = GUIDE	
Other Guide keywords Plus	
One of the Guide Category Keywords	ORGANIZATION_NAME, PHENOMENA_NAME, TEST_TYPE, TEST_METHOD_TYPE, LOCATION_ID, FACILITY_NAME, OPERATION_NAME, EVENT_NAME, EXPERIMENT_NAME, SENSOR_NAME, INSTRUMENT_TYPE, TARGET_NAME, TARGET_TYPE, ARCHIVE_CENTER_NAME, DATA_SYSTEM_NAME
BROWSE_FILE_NAME = ("test.htm")	
END_OBJECT = GUIDE	
END	

FIGURE 4. Guide Label

For example, if ORGANIZATION_NAME = "Defense Nuclear Agency (DNA) (Alexandria, VA)", any objects which share this value for ORGANIZATION_NAME will be linked to this Guide entry. Once it is linked, the detailed descriptions of the value will be stored in HTML format. This file is pointed to by the BROWSE_FILE_NAME value.

Inventories

Metadata models that had to support correlative science resulted in complex models. For example, an instrument model could include complex relationships between events, number of experiments in an event, measured parameters, and physical components of the instrument. Hughes [4] We. realized that there is much more interest in metadata that supplied detailed processing information Or reviews, much of which was lost trying to fit the. data into a fixed model. With the help of the WAIS full-text indexing capability, we were able to supply better textual descriptions and to offer a search capability on those descriptions. Thus, the metadata for an Inventory uses a minimal set of keywords tode.scribe data and stores textual descriptions in either NOTE or DESCRIPTION keywords

Usually, there is a one-to-one relationship between data and metadata. However, it is not effective to have a label for every memo or notebook from someone's file cabinet. By using a localized WAIS index we also have developed a feature to support a collection of various types of documents such as memos, letters, notebooks, etc. It is called *Document Aggregation*. AGGREGATION FLAG and DOCUMENT_TYPE in the Document object indicates whether this metadata represents a single document or a collection of documents; thus, the DARE software can behave appropriately to deliver information to the user. This scheme will also be useful for other types of Inventories.

Views

While working with Numeric data, we found that we couldn't describe data with a limited set of keywords. Data acquired for the study of different nuclear weapons effects phenomena required some unique keywords to search it precisely. Further, looking through the Data Dictionary to find a standard value for a specific keyword could be very time consuming if most of the values are not applicable to a specific nuclear weapon effect; we call this way of looking at a specific nuclear weapon effect a **view**. "1'bus, view specific objects were developed to represent **a** view and **a** set of standard values for vie. w specific keywords.

View specific objects use a set of keywords from a Generic object, in this case the Numeric object. For instance, the AIRBLAST_NUMERIC view specific object uses airblast specific keywords from the NUMERIC object. As we add more view specific objects, new keywords needed for the new view specific object are added to the generic NUMERIC object. Thus, the DARE software treats these view specific objects as a NUMERIC object with a set of view specific keywords for structured field searches. Further, a use of VIEW_NAME and VIEW_DEPENDENCY_FLAG in the Data Dictionary allows the DARE software to present view specific standard values in the Data Dictionary. In the Data Dictionary, if VIEW_DEPENDENCY_FLAG = YES for a specific keyword, VIEW_NAME values are provided in a sequence shown in the following example:

NAME = DATA_TYPE
VIEW_DEPENDENCY FLAG = YES
VIEW_NAME = (GLOBAL, AIRBLAST, CRATER_EJECTA)
STANDARD VALUE SET = ((D,E,F,X,Y,Z),(D,E,F),(X,Y,Z))
Plus the rest of the Data Dictionary Kc>}t'()14 ls

Figure 5. A Data Dictionary Example

In the above example, the keyword DATA_TYPE has three views GLOBAL, AIRBLAST, and CRATER_EJECTA. If a user wants only the AIRBLAST view of the Data Dictionary, the DARE software extracts standard values of D, E and F from the STANDARD VALUE SET. The GLOBAL view is the union of all the views.

Data Formats

Not only the metadata format but also the data formats had to be standardized in order to process and to deliver the data effectively. We used the following criteria in selecting the DARE data formats - TIFF, GIF, PDF, JPEG, MPEG, and QUICKTIME:

- Easily accessible by public domain software
- Contain a good description of data.
- Support various compression schemes (e.g., LZW, CCITT, JPEG, MPEG, etc.).
- Extensible and interchangeable.
- Simple to implement.
- Minimal amount of descriptor overhead required by the file format.
- Small amount of data access overhead

DARE supports the following Inventories:

- Document
- Photo
- Diagram
- Numeric Data
- Software
- Table
- Video

For example, the standard DARE archive format for scanned images is Tagged Image File Format (TIFF). There are two exceptions: images which can be viewed at a low resolution may be submitted in Graphical Image Format (GIF) and images which are in-line with the document text must be submitted in GIF. DARE also creates browse versions of the archive data for display by the DARE system. For files which are viewed as a series such as pages of documents or a sequence of photos, the TIFF files are converted to Adobe Portable Document Format (PDF). Thumbnail and browse versions of color and grayscale images (e.g., photos) are created in the GIF and JPEG File Interchange Format, respectively. Thumbnail and browse versions of black and white images (e.g., diagrams or plot of numeric data) are created in GIF.

Scanned documents, photos, videos, and diagrams require different resolutions and compression schemes. The DARE Data Submittal Manual (DSM) specifies the formats and recommends standard practices for preparing data.

Tools for Metadata and Data

To expedite the data ingestion process, we developed tools to generate, validate, and process metadata and data.

Generation

Label Editor Tool allows users to create metadata by allowing them to choose values from a list of standard values in a scroll box. The tool takes the Data Dictionary as an input, so users can select a desired object and generate metadata labels easily. Other tools can generate labels from Tab separated ASCII output of a simple database manager (e.g., FoxPro, DBASE II), or a sophisticated on-line relational data management system (e.g., SYBASE, ORACLE). Or a user can download a template - label without values filled in - from the DARE system, and use an editor to generate labels. For data generation, there are many ways to create the data in accordance with the DARE standards.

Validation

lvtool and *slvtool* are tools which read DARE labels and perform the following types of error checking:

- ODL Language Checking. ODL language checking detects errors in the usage of ODL, such as missing quote marks, invalid characters in names, other syntax errors, etc.
- Data Dictionary Checking. The keywords and objects used in DARE labels follow certain DARE standards, as well as rules specified by the label designer. Data Dictionary checking determines when these rules have been broken by comparing each label to the Data Dictionary.

Other tools include a SGML/HTML syntax checker. The DARE software that generates browse data also checks the formats of the archive data.

Process

Many tools have been written to process various types of archive data to generate browse data. As the DARE system relies on network transfer for delivering data to users, it is necessary to have browse data files as small as possible. Thus, archive data are converted to various formats such as GIF, PDF, JPEG, MPEG, and QUICKTIME. Further, runtime software can process metadata to generate SGML/HTML versions of the metadata.

System Architecture

Client-Server Model

The DARE system architecture is based on a client server model. This architecture offers the following advantages:

1. Users can use existing desktop computers with little or no additional expense for hardware or software.
2. As the system grows, additional DARE servers can be added anywhere on the network.
3. Servers for classified and unclassified information can be placed on different networks.
4. Links to other database servers and analysis systems are relatively easy to implement.

DARE's technical approach is to maximize the use of Commercial Off The Shelf (COTS) hardware and software and Non-Developmental Item (NDI) software available as freeware, shareware, or in the public domain. Custom software development is minimized and is typically written to integrate COTS and NDI software components.

DARE uses open system technologies and standards. A few examples of open system technologies adopted by DARE are: SCSI interfaces for hardware archive storage devices; the UNIX operating system for the server; TCP/IP network protocol; the free WAIS sf (structured field WAIS) search engine and the HyperText Transfer Protocol (HTTP). World Wide Web (WWW) technologies provide the powerful user interface and server software capabilities required by DARE

Custom Software

The custom DARE software builds HyperText Markup Language (HTML) pages for browsing data inventories, full-text indexes and structured field indexes. These HTML pages produced both at setup and runtime are what makes DARE unique. All the HTML pages are generated by custom Perl scripts while the underlying WAIS-sf search engine generates the indexes.

The HTML pages created by setup are static pages used for keyword/value searches. The HTML pages created by runtime are dynamic pages used for full text and structured field search results as well as the display of Inventory information which is hypertext linked to ancillary information in the Guide and the Data Dictionary. All of these tasks are easily attainable due to the underlying structure of the DARE labels. The DARE labels could also be easily loaded into a Database Management System (DBMS).

The DARE custom software is composed of three main subsystems:

- Configure: Prepares the DARE software environment using configuration files to identify data inventories and supporting information
- Setup and Setup Sequencer: Creates the static keyword/value HTML pages, WAIS indexes, and Perl arrays needed for runtime execution. Setup is run on whole inventories while Setup Sequencer is run when data products are added, updated or deleted.
- Runtime: Executes and summarizes WAIS searches, displays information about a single Inventory or supporting information items, and dynamically tags the displayed label.

Figure 6. The DARE System Architecture

Search Modes and User Interface

There are basically two ways of searching:

- By conducting a full-text search of all DARE metadata elements (Data Dictionary, Directory, Guide, Inventories).
- By selecting one metadata element (Data Dictionary, Directory, Guide, Inventory) as the starting point of the search.

Full-text search of all Metadata Elements

A full-text search will match search words that are present anywhere in a label in all metadata elements. When a full-text search is conducted the following pages are created by the runtime software and presented to the user:

1. An HTML full-text summary page. This full text summary page indicates which metadata elements have data that match the search criteria. The user can then select the category of interest by clicking on it.

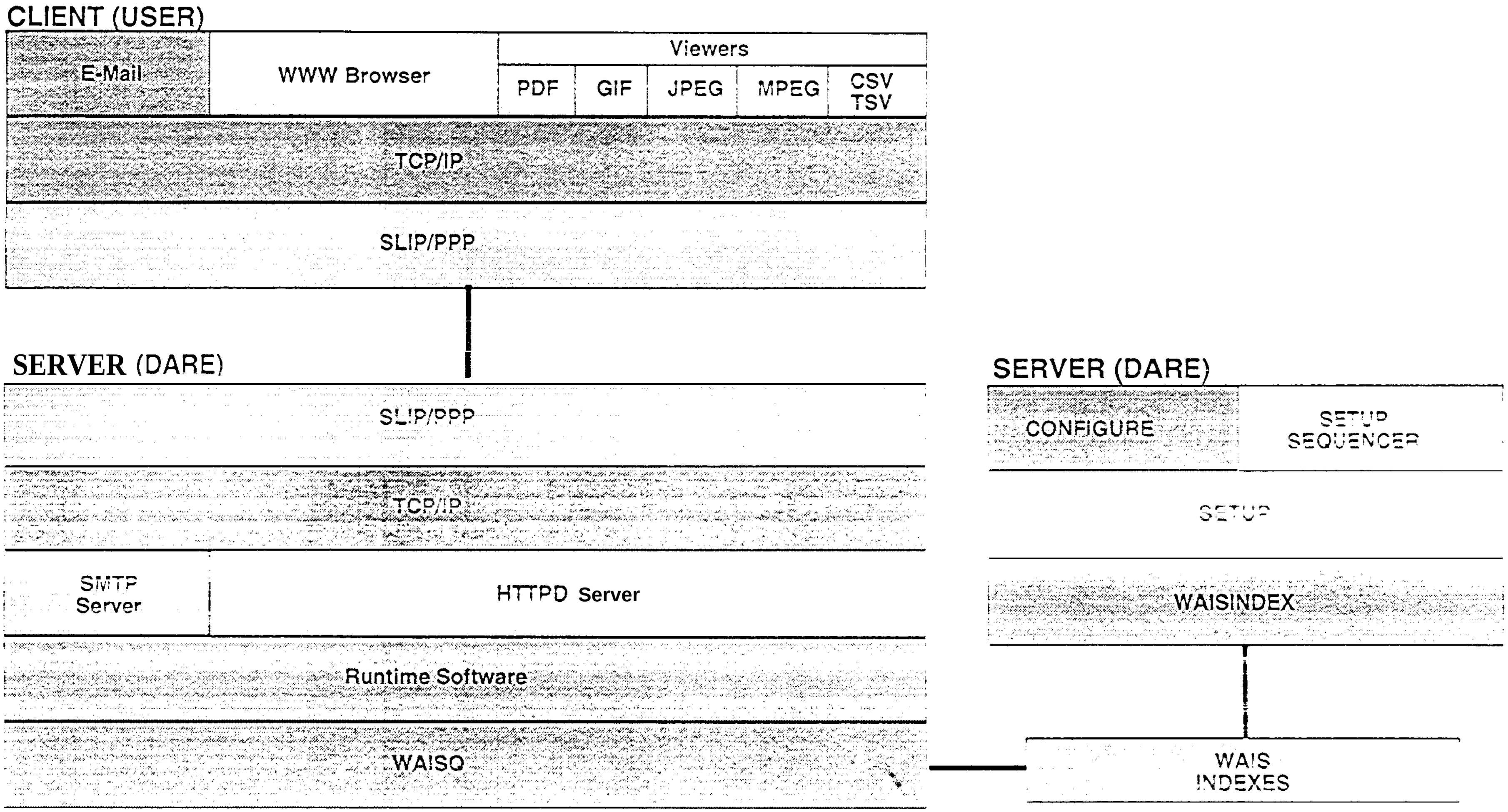

Figure 6. The DARE Software Architecture

2. An HTML title page. This title page lists the titles of the data products for the selected metadata element that matched the search criteria. The user can then select a title of interest by clicking on it.
3. An HTML layout page. The runtime software uses the label to generate the layout page. This layout page, generated from the label by the runtime software, summarizes the label information:
   - Provides the title.
   - Provides the classification.
   - Presents the user with viewing options for the data product options vary from one inventory to another. The options for the photo Inventory include viewing the online or fLlll-resolution photo and browsing the archive directory while the options for the table Inventory include viewing the online table., downloading the archive table and browsing the archive directory.
   - Provides links to related information in the Guide and Directory.
   - Provides the option of displaying the label.
4. When the option of displaying a label is selected, an HTML tagged label is presented to the user. The runtime software links the tagged label to the Data Dictionary.

Figure 7 **shows** an example of the HTML title, layout and tagged label pages.

Figure 7. The DARE User Interface

Single Metadata Element Searches

When **a** search is started from a metadata element three different types of searches are available:

1. Kc.ywor(i/value search.
2. l:ull-text search.
3. Structured field search.

Keyword/Value Search

The keyword/value. search pages are generated at setup time by parsing the labels and organizing **the keywords** and values into HTML files. A user not familiar with the DARE data may start **a** search this way. This search is also valuable because it presents the user with all the titles that contain **a** certain keyword/valLlc combination. Note that WAIS searches are limited to a maximum number of hits and some. da(a may not be easily retrieved. For instance, in DARE the maximum number of hits for a WAIS search is set to 1000. The following pages are presented to the user when a keyword/value search **is** conducted:

1. An HTML keyword list. This list is metadata element dependent. A user can select a keyword by clicking on it.
2. An HTML list of values for the selected keyword. '1 he user can then select a value by clicking on it.
3. An HTML list of titles for the keyword/value selected. The user can then select a title by clicking on il.
4. An HTML layout page as described above.
5. An HTML tagged label as described above.

Full-Text Search

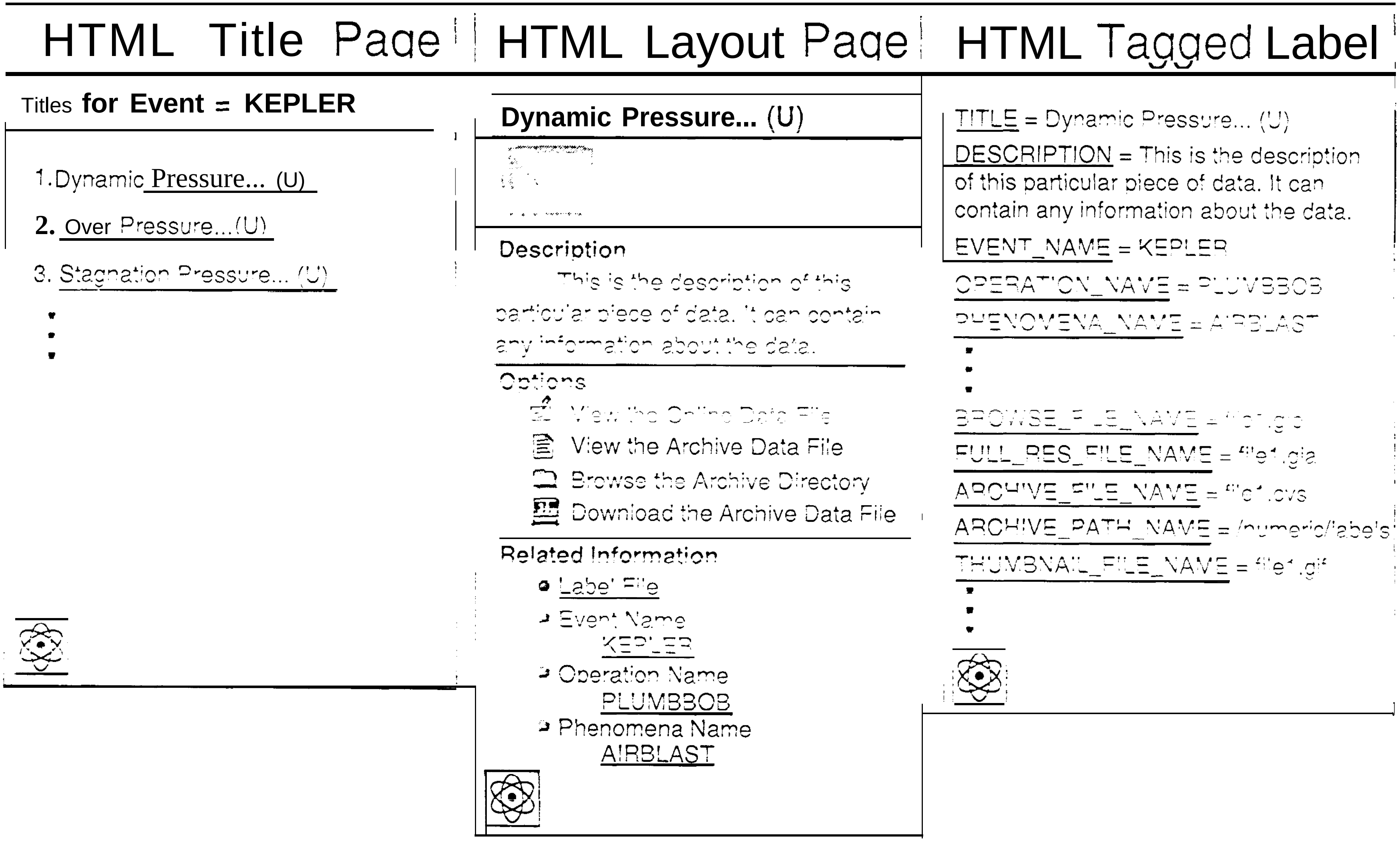

Figure 7. The DARE User Interface

The full-text search for an individual metadata element only differs from the all metadata element search in that the summary page is not produced. When a full-text search is conducted the following pages are created by the runtime software and presented to the user; these pages are the same as the ones described above:

1. The HTML title page,
2. The HTML layout page,
3. The HTML tagged label.

Structured Field Search

A structured field search allows the user to specify a value (or range of values in case of numeric and date keywords) for one or more keywords for a given search. The keywords available for selection are metadata element dependent. When a structured field search is conducted, the same pages described in the full-text search are created by the runtime software and presented to the user.

summary

The DARE metadata model and the standards implemented allow for easy portability to other standards and indexing systems. For example, DARE is now using HTML for its browse pages and display formats. With minor changes to the code these HTML pages could be converted to SGML when the community feels that SGML is the standard to be adopted.

Because of the modularity of the DARE software and because the DARE labels are in ASCII format the current search engine (free WAIS-sf) could be replaced with minor impact to the underlying code. Furthermore, DARE has not selected to implement a DBMS; however, the structure of its metadata will make it very easy to port to a relational database.

"The DARE development staff recognizes that there are a lot of processing steps between selecting a data format and making the information available to users of the DARE system. Each time a new type of data is planned for incorporation into DARE a data engineering process is required. During this process, the Data Engineer selects a data format and works with the data providers and users to determine the metadata requirements for the data." Kelley [5] We strongly recommend that people take the time to design metadata for new data types since only a good metadata model can support a good retrieval system.

Acknowledgments

The work described in this paper was carried out by the Jet Propulsion Laboratory, California Institute Of Technology, under a contract with the National Aeronautics and Space Administration. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not constitute or imply its endorsement by the U.S. government or the Jet Propulsion Laboratory.

We would like to thank the DARE development staff - Carol Miller, Keevin Fisher, Mark Takacs, Kristy Kawasaki, Patrick Murphy, and Oscar DeMartino for their support in writing this paper.

References

1. PDS Standard R eference, JPl Internal Document D-7669, Jet Propulsion Laboratory, November 1992
2. "DARE Data Dictionary", JPl Internal Document, Jet Propulsion Laboratory, March 1996
3. Hyon, J., "DARE Data Submittal Manual", JPL Internal Document, Jet Propulsion Laboratory, October 1995
4. Hughes, J. S., Bernath, A. M., "Planetary Data System Data Model as Applied to WWW interfaces (WEBCAT)", JPL & NASA Case No. NPO 196[]'/. October 1995
5. Dr. Kelley, C. S., Miller, C. L., Walsh G., "Data Archival and Retrieval Enhancement (DARE)", Science & Technology Digest, Volume 2, September 1995