Technical Document 1483
February 1989

High-Level Database Manager Design for the High Gain Initiative (HGI)

ORINCON Corporation

NAVAL OCEAN SYSTEMS CENTER

San Diego, California 92152–5000

E. G. SCHWEIZER, CAPT, USN
Commander

R. M. HILLYER
Technical Director

ADMINISTRATIVE INFORMATION

This work was performed for the Office of Naval Technology (ONT), Office of the Chief of Naval Research, Arlington, VA 22217, under program element 62314N. Contract N66001-87-D-0058 was carried out by ORINCON Corporation, 9363 Towne Centre Drive, San Diego, CA 92121, under the technical coordination of C.E. Persons, NOSC Code 732.

Released by
C.E. Persons, Head
Processing Applications Branch

Under authority of
J.H. Roese, Head
Signal and Information
Processing Division

FS

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) OCR88-U-0018	5. MONITORING ORGANIZATION REPORT NUMBER(S) NOSC TD 1483
6a. NAME OF PERFORMING ORGANIZATION ORINCON Corporation 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION Naval Ocean Systems Center
6c. ADDRESS (City, State and ZIP Code) 9363 Towne Centre Drive San Diego, CA 92121	7b. ADDRESS (City, State and ZIP Code) San Diego, CA 92152-5000
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Office of Naval Technology 8b. OFFICE SYMBOL (If applicable) ONT	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER N66001-87-D-0058
8c. ADDRESS (City, State and ZIP Code) Office of the Chief of Naval Research Arlington, VA 22217	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO.: 62314N PROJECT NO.: SW17 TASK NO.: RJ14020 AGENCY ACCESSION NO.: DN308 291

11. TITLE (include Security Classification)

HIGH-LEVEL DATABASE MANAGER DESIGN FOR THE HIGH GAIN INITIATIVE (HGI)

12. PERSONAL AUTHOR(S)

C.E. Persons

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Final	FROM Oct 1986 TO Jan 1988	February 1989	39

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	acoustic arrays and detection

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

This report defines the high-level database management design for the four basic High Gain Initiative (HGI) databases (environmental, raw, synthesized, beamformed, processed, and analyzed data). Also presented is a common user interface for all types of data.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION
☐ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT ☐ DTIC USERS	UNCLASSIFIED

22a. NAME OF RESPONSIBLE PERSON	22b. TELEPHONE (include Area Code)	22c. OFFICE SYMBOL
C.E. Persons	(619) 553-2023	Code 732

DD FORM 1473, 84 JAN

83 APR EDITION MAY BE USED UNTIL EXHAUSTED
ALL OTHER EDITIONS ARE OBSOLETE

OCR 88-U-0018

HIGH-LEVEL DATA BASE MANAGER DESIGN FOR THE HIGH GAIN INITIATIVE (HGI)

Technical Report
29 January 1988

Contract No. N66001-87-D-0058
Delivery Order No. 005
CDRL A011

Submitted To:

NAVAL OCEAN SYSTEMS CENTER
Attn: Receiving Officer, Code 222
271 Catalina Blvd.
San Diego, CA 92152

9363 Towne Centre Drive, San Diego, CA 92121 Phone: (619) 455-5530 FAX #: (619) 453-9274

TABLE OF CONTENTS

1.0 INTRODUCTION

This report presents a high level design for a Data Base Manager to support research under the High Gain Initiative program. Prepared under NOSC Contract No. N66001-87-D-0058, Task 5, the report draws upon the "Data Base Management Requirements of the High Gain Initiative (HGI)" document produced earlier under the same task. Research conducted during the requirements definition phase indicated that the volume of HGI data to be processed and the complexity of the beamforming, classification, and localization algorithms mandate careful attention to the design of an HGI data base manager if a productive research environment and an efficient eventual fielded system are to be achieved.

An important driver of the HGI data base manager design process is the as yet undefined nature of the eventual "best" HGI processing suite. Since the sensor being used, the High Gain Array (HGA) is experimental, ultimate fielded configurations are unknown at this time. Also, while the generic nature of the signal processing, beamforming, detection, localization, and tracking algorithms that will be employed are well understood, the implementations that will produce the best results, within the bounds of near-real-time processing requirements, are currently the subject of extensive research. Therefore, it is not only not possible, but also not desirable, to impose a data base management scheme that is rigidly defined as to the types and structure of the data it will manage and the processing algorithms that it will invoke. At the same time, it is important to provide enough structure and commonality of user interfaces to enable numerous researchers to work efficiently and pool their results. If this is not done, the time of individual researchers could be wasted in implementing individual versions of signal processing, display, and other software, and comparability of results could suffer from lack of consistency and configuration control. The goal of this report is to define an HGI data base management approach that strikes a happy medium between the extremes described above.

The remainder of this report is organized as follows: Sections 2 through 5 define the high-level data base management design for the four basic HGI data bases (environmental data, raw and synthesized data, beamformed data, and processed and analyzed data); and Section 6 presents a common user interface for all types of data, including commercially available interfaces that might satisfy HGI requirements.

2.0 MANAGEMENT OF ENVIRONMENTAL DATA

During the HGI requirements definition phase, the System for Predicting the Acoustic Response of SOSUS (SPARS) was selected as being the better choice for HGI use than a collection of unintegrated environmental data bases and models from the Fleet Numeric Oceanographic Center. SPARS is a self-contained system that is currently being ported from the HP1000 to the HP9000 computer under the UNIX operating system. SPARS migration to a SUN workstation as part of the Surveillance Direction System (SDS) is also planned.

SPARS is equipped with a menu-driven user interface that supports selection of input data bases, selection of models to be used, insertion of hand-entered data such as gulf stream and eddy positions, and specification of output data bases to be created and stored. SPARS processing is based on the Sonar Equation. If beam noise and signal gain are used as inputs, SPARS solves the following form of the equation:

$$SE = SL - TL - BN + SG - RD$$

where:

SE = Signal excess
SL = Source level
TL = Transmission loss
BN = Beam noise
SL = Signal gain
RD = Recognition differential

When omni-directional hydrophone noise and array gain are used as input, the following variation of the SONAR equation is solved:

$$SE = SL - TL + AG - RD - N$$

where:

SE = Signal excess
SL = Source level
TL = Transmission loss
AG = Array gain
RD = Recognition differential
N = Omni-directional hydrophone noise

SPARS processing follows a fixed hierarchy, depending on the version of the Sonar Equation chosen. This hierarchy is illustrated in Figure 2-1.

2.1 Structure of the Environmental Data

Since the SPARS data bases are already defined, this section will summarize the existing structures rather than make recommendations for new structures. SPARS employs two basic data base organizations: geographic grids and Indexed Sequential Access Method (ISAM) with user-specified keys. ISAM is used for some of the stored input data bases and also for a variety of output data bases that the user may wish to store for an indeterminate length of time. Keys for the gridded data bases have a hierarchical numbering scheme that reflects location within ocean basis, as follows:

1000 Pacific
- 1100 North Pacific
  - 1110 NORPAC
  - 1120 EASTPAC
  - 1130 WESTPAC
  - 1140 CENPAC
  - 1150 Bering Sea
- 1500 South Pacific

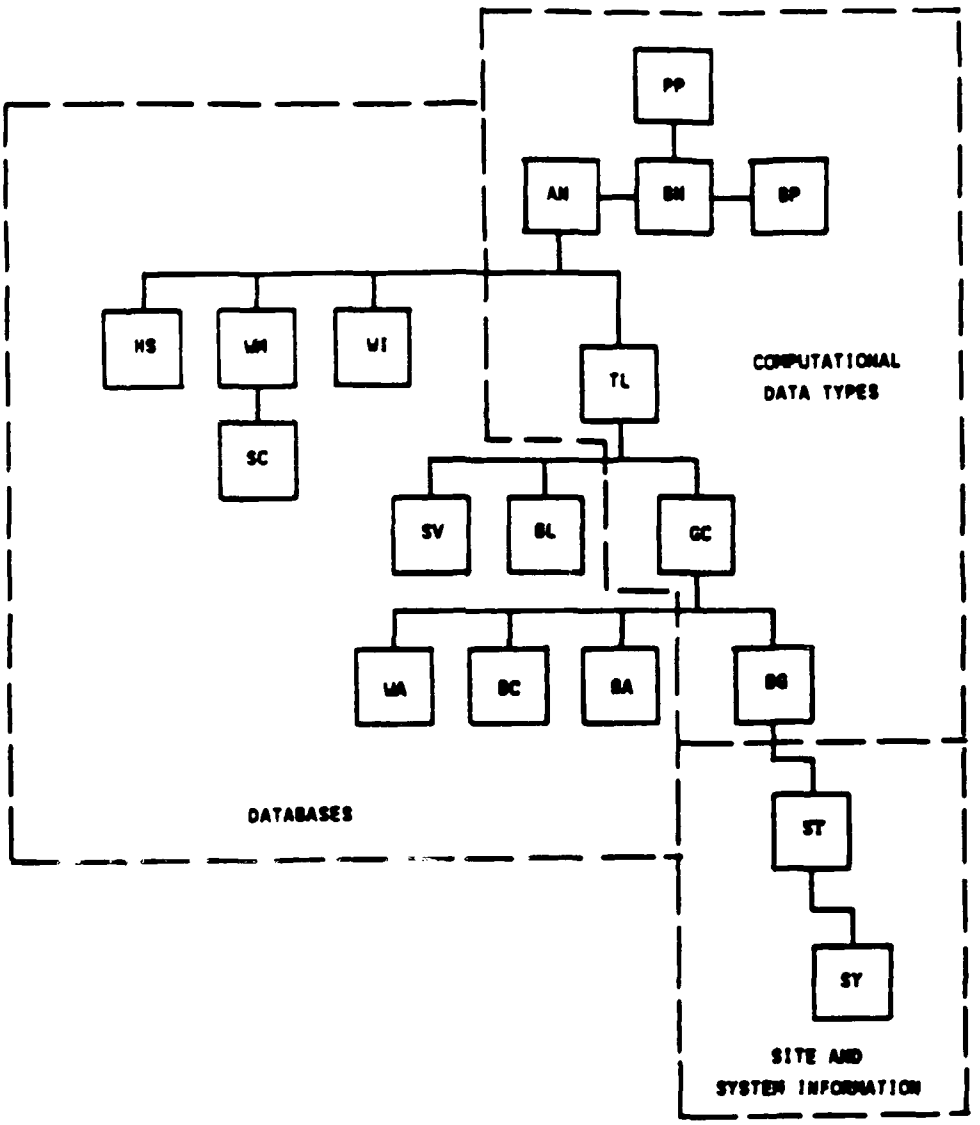

Figure 2-1. SPARS Computational Hierarchy

2000 Atlantic

2100 North Atlantic

2110 NORLANT

2120 EASTLANT

2130 WESTLANT

2500 South Atlantic

3000 Mediterranean

4000 Indian Ocean

Within this hierarchy, stored latitude and longitude points representing the corners of a given grid are compared with the user's input coordinates to identify the desired data. Key definitions for the ISAM files is very flexible, and current files have from one to six assigned keys. The ISAM file keys are shown in Table 2-1.

2.2 Size and Residence Requirements of Environmental Data

The sizes of the SPARS data bases for the largest ocean basin, The North Pacific, are given in Table 2-2. Since typical use of SPARS will be for a smaller area, these figures can be considered worst-case. With respect to disk and core memory residence requirements, these are determined dynamically by the SPARS computational processing hierarchy illustrated in Figure 2-1. For example, the site and system information data bases (ST and SY) will be core resident while bearings (BG) are computed, but can then revert to disk. The watermass (WA), bottom class (BC), and bathymetry (BA) data bases will be core resident along with the bearings (BG) data while environmental extraction (GC) data is computed. In turn, GC, sound velocity profiles (SV), and bottom loss (BL) data are used to compute transmission loss (TL); after that point, they are no longer needed in core, and so on. Statistically, the greatest core residence burdens occur during the computation of environmental extraction-data (GC), requiring 6,201,344 bytes of input data and 1,121,280 bytes of output data for the North Pacific basin, and during the final computation of performance prediction, requiring 15,360 bytes of input data and 5,059,584 bytes of output data. While on-line disk storage in the range of five to eight megabytes is easily obtainable, many current processing configurations have smaller amounts of primary memory. The result will be extensive use of "virtual memory," involving

Table 2-1. ISAM File Keys

Input ISAM Files	Keys
Sound Velocity Profiles (SV)	1. Season 2. Watermass
Bottom Loss (BL)	1. Bottom class no. 2. Curves for BL curves Profile for Geo-Acoustic profiles
WMO Ships (WM)	1. Call Sign
Systems Characteristics (SY)	1. Integer (-1 = Hydrophone positions, 1 = Bearing table, 2 = Beam numbers)
Output ISAM Files	
Transmission Loss (TL)	1. Receiver 2. Source depth 3. Frequency 4. Bearing index
Ambient Noise (AN)	1. Type of noise 2. Receiver depth 3. Frequency 4. Time from start of run
Beam Noise (BN)	1. Frequency 2. Time from start of run 3. Array heading
Bearings (BG)	1. Bearing table 2. Beam space table
Computed Beam Pattern (BP)	1. Frequency 2. Fixed or interstitial integer code
Performance Prediction (PP)	1. Receiver depth 2. Source depth 3. Frequency 4. Bearing index 5. Time from start of run 6. Array heading
Environmental Extraction (GC)	1. Bearing index 2. Environmental parameter two-character string (e.g., BA, WA, BG, BC)

Table 2-2. Estimated Sizes of SPARS Data Bases

Input Gridded Files		Size (Bytes)
Bathymetry (BA)	1/6 deg. squares	2,065,408
Watermass (WA)	"	2,065,408
Bottom Class (BC)	"	2,065,408
Historical Shipping (HS)	1 deg. squares	262,144
Synoptic Winds (WI)	"	437,248
Synoptic Sea Surface Temperature (ST)	"	62,464
Synoptic Mixed Layer Depth (LD)	"	62,464
Input ISAM Files		
Sound Velocity Profiles (SV)		132,096
Bottom Loss (BL)		19,456
WMO Ships (WM)		27,136
Source Characteristics (SC)		3,072
Systems Characteristics (SY)		7,168
Sample Output ISAM Files		
Transmission Loss (TL)		1,780,224
Ambient Noise (AN)		4,096
Beam Noise (BN)		5,120
Bearings (BG)		2,048
Computed Beam Pattern (BP)		6,144
Performance Prediction (PP)		5,059,584
Environmental Extraction (GC)		1,121,280

swapping of pages from core to disk. However, it is unlikely that typical use of SPARS for HGI purposes will require processing data from such a large ocean area.

2.3 Estimated Disk Accesses for Environmental Data

As was discussed in the report on HGI data base management requirements, statistics on SPARS data base accesses are not available, but they can be inferred from the file structures. For the gridded data bases, access should be at worst linear to the number of grid areas required to bring in al the data within the user's specified area of coverage. In cases where the relevant grid data are stored contiguously on the disk, the required accesses may be far fewer.

Access to the ISAM files will be dependent both on the value range specified for a given key by the user versus the key quantization in the file (the same ratio that applies to the gridded files) and to the number of keys specified. As a rule, the more keys the use specifies, the more accesses there will be to the stored indices, but once the location of the desired data is determined, it can be retrieved with one disk access. Conversely, specification of just one key will minimize index accesses, but may require a number of disk accesses to get all the data that has the specified single key value. A very rough worst case estimate is that the number of accesses required will be the number of keys specified plus the number of disk blocks containing the requested data.

2.4 Common Interface Requirements for Environmental Data

The important aspect of SPARS processing for HGI research purposes is maintaining a "trace" of the input data and parameters used in a given SPARS run so that results can be compared. In particular, any real-time or hand entered data, such as measured (as opposed to modeled) beam noise or synoptic data from FNOC, should be stored and its file name recorded in the header for environmental data that is transferred from the HP9000 or SUN to the general-purpose computer system that is being used for primary HGI processing. This file will also contain modeling descriptors, including version used and parameters. Finally, the file should contain the performance prediction results, or replicas, from a given SPARS run, so that they

can be used repeatedly in HGI processing without having to run SPARS again on the same data. File headers and contents will be further discussed in Section 6.0, HGI Common User Interface.

3.0 MANAGEMENT OF SYNTHESIZED AND RAW MEASUREMENT DATA

Synthesized and raw measurement data are being discussed together in this section, because, with the exception of the need for error-checking of raw measurement data, their storage and access requirements are the same.

3.1 Structure of Synthesized and Raw Measurement Data

Given the expected volume of raw data from the High Gain Arrays and current limitations on the density of magnetic storage media, it must be assumed that once error-checking and validation have been performed, the raw data will be stored for archival or temporary recall purposes only. The reason for this is that suitable tape storage, for example the Ampex high-density digital cassette recorder, supports sequential access only and is not fast enough for read/write signal processing that is essentially random access in nature. Current optical disk technology supports the required storage density and random read access, but cannot be over-written with the results of validity checking or intermediate signal processing outputs.

The structure of the measurement data is simple and basically hierarchical. As shown in Figure 3-1, organization is by time, stave, and phone. In the case of synthesized data, or real data that contained no errors, the block address of data from a phone on a given stave at a given time could be computed and accessed directly. Since validity checking will almost surely result in the modification or elimination of some of the phone data, the final contents of the measurement data file will have a fine-grained organization that varies from block to block and cannot be predicted in advance. Therefore, data from a given phone will be located via indices in the file header rather than via computation. A sample structure for these header indices is illustrated in Figure 3-2.

During HGI experiments, data will also be collected on actual phone positions by acoustic sources used for calibration. This is made necessary by the drift in phone positions on tethered

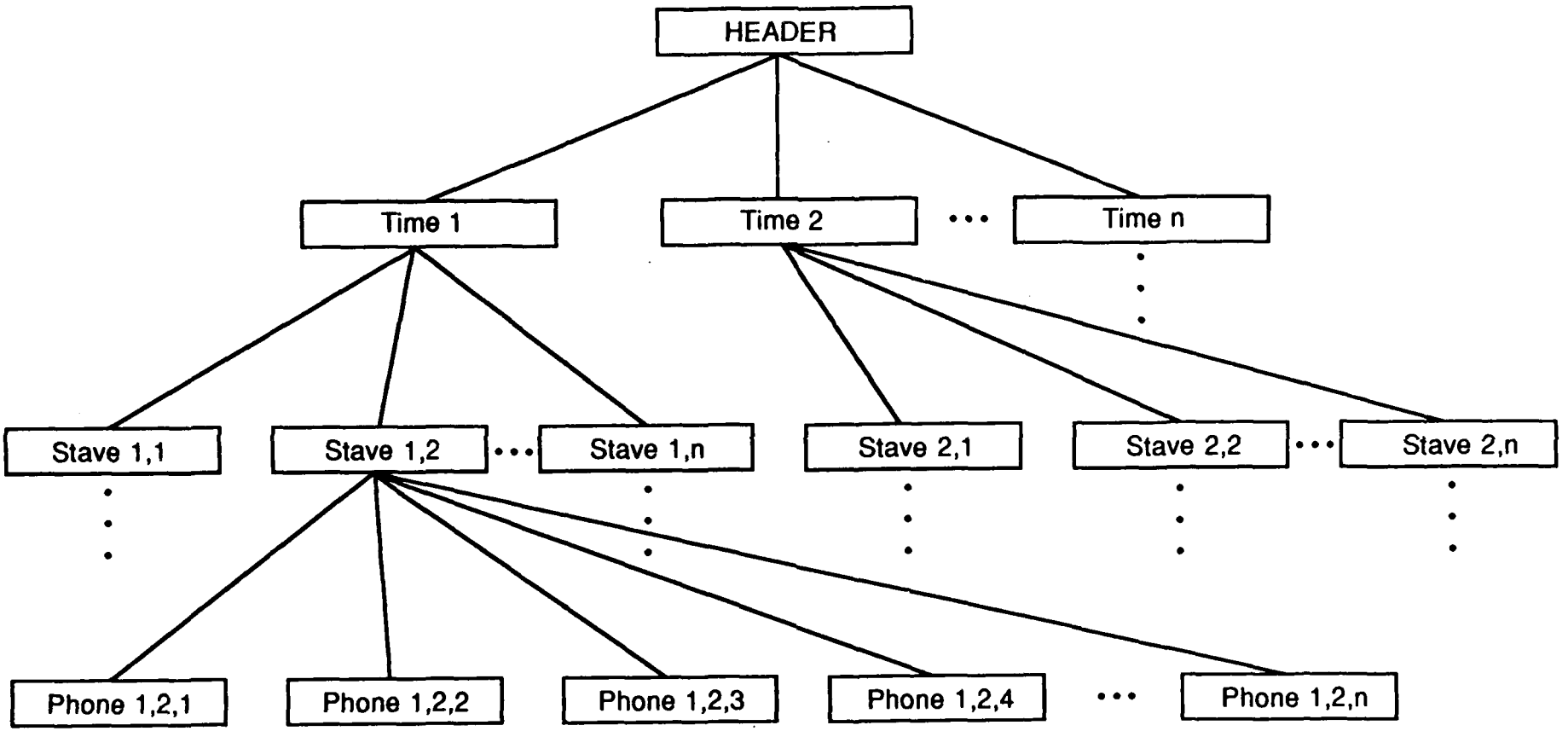

Figure 3-1. Hierarchical Structure of HGI Measurement Data

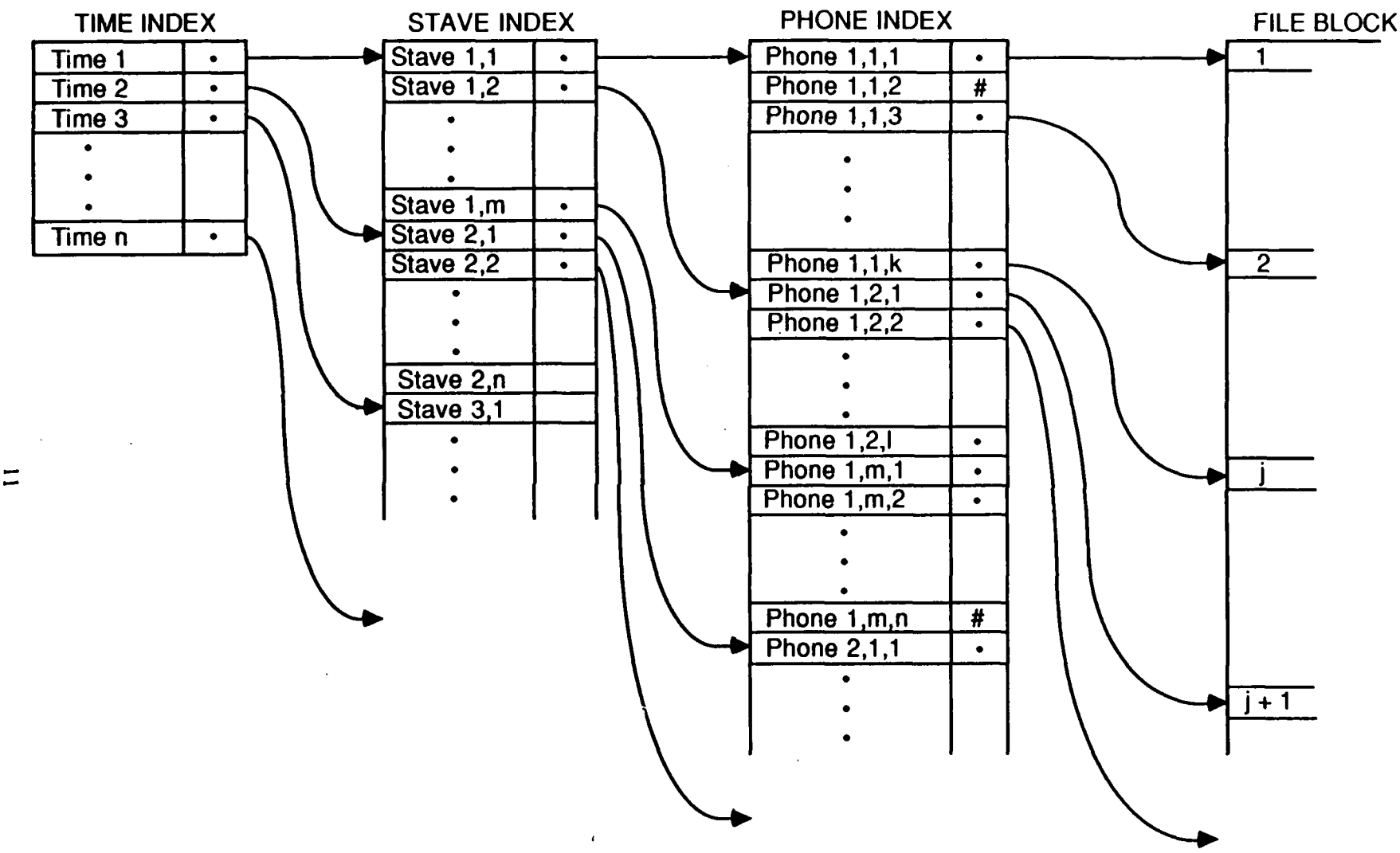

Figure 3-2. Index Structure of the HGI Measurement Data File

staves caused by ocean currents. Phone positions can be stored in the file header by time and stave, but for efficient data processing, they should also be stored in a separate index organized four-dimensionally by time, latitude, longitude, and depth. In this way, software can access the file to determine which phone was closest to a given position at a given time.

3.2 Size and Residence Requirements of Measurement Data

The size of the HGI measurement data bases is not precisely known at this time, but it can be estimated from the HGI experiments that will be conducted periodically from the present to 1992. Taking 200 phones as an example, and typical sampling parameters, generates

> 200 phones x 100 samples per second x 2 bytes per sample x 3600 seconds per hour x 24 hours per day = 3.46 gigabytes per day.

Later experiments may involve as many as 1,000 phones, resulting in 17.3 gigabytes of data per day.

On the assumption that most of this data will be spooled onto archival storage, a more realistic figure to estimate would be two hours of on-line data to support beamforming computations and data recall. This amounts to 288 megabytes for the 200 phone case and 1.44 gigabytes for the1,000 phone case. These figures are within the capacity of the largest magnetic disk drives currently available.

Validity checking done in the time domain will be basically sequential. If data is checked on a phone-by-phone basis, the amount that must be core-resident at a given time will not be large and will depend upon the time interval necessary to give meaningful results when the first four statistical moments are computed. As was discussed in Section 2.2.3 of the requirements report, use of alternate buffers for validity checking would double the space requirements but would also increase efficiency.

The result of the validity checking may be some rewriting or editing of the measurement data, and also the construction of the header indices. It is important that the header indices for a moving window of two hours (or whatever recall period is chosen) be stored in primary

memory, but the space requirements will not be large. Assuming three 2,000 sample validation sets combined to form time blocks of one minute, five staves, and 200 phones per stave, and two words per index entry, index storage requirements for two hours of data would be approximately 240K words, or 120K words if packed.

Residence requirements for the computation of fast Fourier transforms (FFTs), the initial processing on validated measurement data that converts it from the time to the frequency domain, will also depend on the time interval chosen for conversion. Since the FFTs are applied to one phone at a time, they can be performed sequentially. Also, they do not increase the dimensionality of the data. Assuming that the time series data is averaged over a 160 second interval, data sampled at 100 samples per second at two bytes per sample would require 32K bytes for input and 32K bytes for output.

3.3 Estimated Disk Accesses for Measurement Data

The basically sequential mode of processing that applies to HGI measurement data allows disk accesses to be kept to a minimum. For validity checking, read operations will be one per block, but since the data will be read from a contiguous stream, no radial disk arm movement or non-sequential tape read head positioning will be required, unless other processes are allowed to interrupt the validity checking. Since validated data will typically be written to a different, random access device, interruptions should not be necessary.

Accesses in support of FFT processing can take one of two patterns. The simpler, and no doubt better, for post analysis of experiment data bases is to write out the validated data and create the header indices in a separate step before FFT processing begins. FFT processing would then require navigation of the indices to locate the beginning of each block of data, but as long as the data was read sequentially, little or no radial arm movement would be required and access would be very efficient.

In a real-time system, a potential bottleneck would be contention for control of the disk arm between the process writing the validated data and the process reading the data to compute FFTs. To avoid this, the validated data could be passed directly on to FFT processing in primary memory. While the validated data would still be written to disk, the contending read

operations would be eliminated. This approach will require multiple buffers and careful attention to synchronization, but it is perfectly feasible. It would be wise to test and refine this approach during the HGI research phase, so that it will be well understood before a real-time system is required.

3.4 Common Interface Requirements for Measurement Data

The most important aspect of measurement data that should be incorporated in the common user interface is a complete description of its characteristics. This would include information on array spacing, position, and depth, phone parameters, data times and rates, etc. Data should also be recorded on the validity checking parameters. (Note that the header indices will indicate data that has been reduced or eliminated.) Finally, information should be recorded on associated data files, such as the file containing calibrated phone positions organized by time and position, and the file containing the associated truth track if one exists.

4.0 MANAGEMENT OF BEAMFORMED DATA

The beamforming process for HGI data is currently the subject of intensive research, and the best suit of algorithms for generating accurate results in real time is not yet defined. Given the very large amounts of data to be processed and the potentially very high number of computations that will be required, a data base manager that is flexible, efficient, and maintains a complete record of processing parameters that were used is very important for supporting productive research that avoids false starts and reprocessing of data and ensures comparability of results.

4.1 Structure of the Beamformed Data

The initial step in the beamforming process — generation of FFTs — will result in data in the frequency domain that is of the same dimensionality and is stored in the same time/stave/phone hierarchy as is used for the measurement data discussed in the previous section. If additional validity checking is done on the FFT outputs, for example, to detect phones that were wired

backwards, the index structures developed for the measurement data may have to be modified to reflect further editing. Otherwise, their structure will be the same as that shown in Figure 3-2, but within each data block, the data will be ordered from the beginning to the ending frequency, rather than from the beginning to the ending time.

The next step in the beamforming process is the computation of cross-spectral matrices (CSMs) or their equivalents. Since CSMs are computed by bringing in the FFT data at a given frequency from all phones at a selected time, their creation represents both an orthogonalization of the data descriptors and a major increase in the size of the data. Once the CSMs have been accumulated to the point that they can be inverted, they will be stored in a file organized by time and frequency.

Editing due to validity checking in the frequency domain or partitioning of CSMs due to processing constraints will result in CSMs of varying sizes. This will require creation and maintenance of an index structure for the CSM file, since starting addresses for CSMs will not be possible. While the basic index structure will consist of a hierarchy of pointers, information will have to be maintained on included phones. A sample index structure employing bit maps on included phones is illustrated in Figure 4-1.

Additional files that will be created during the beamforming process consist of beam and beam noise data. Beam data is typically organized by time, stave, and perhaps phone, and includes descriptors such as number of beams, total angle if uniform or angle by beam, number of non-unique orientations, delta time or delta frequency, depending on the domain, and integration time. Beam noise data is typically stored by time, stave, azimuth, and elevation angle for display and comparison with modeled results.

4.2 Size and Residence Requirements of Beamformed Data

The primary storage requirements of computing FFTs and validity checking in the frequency domain, as discussed in Section 3.2 above, are relatively small — roughly 32K bytes for input and 32K bytes for output. In contrast, the primary storage requirements of computing and accumulating CSMs until they are ready for inversion are extremely large, because the basic dimensionality of a CSM is the number of phones squared, and one is computed for each

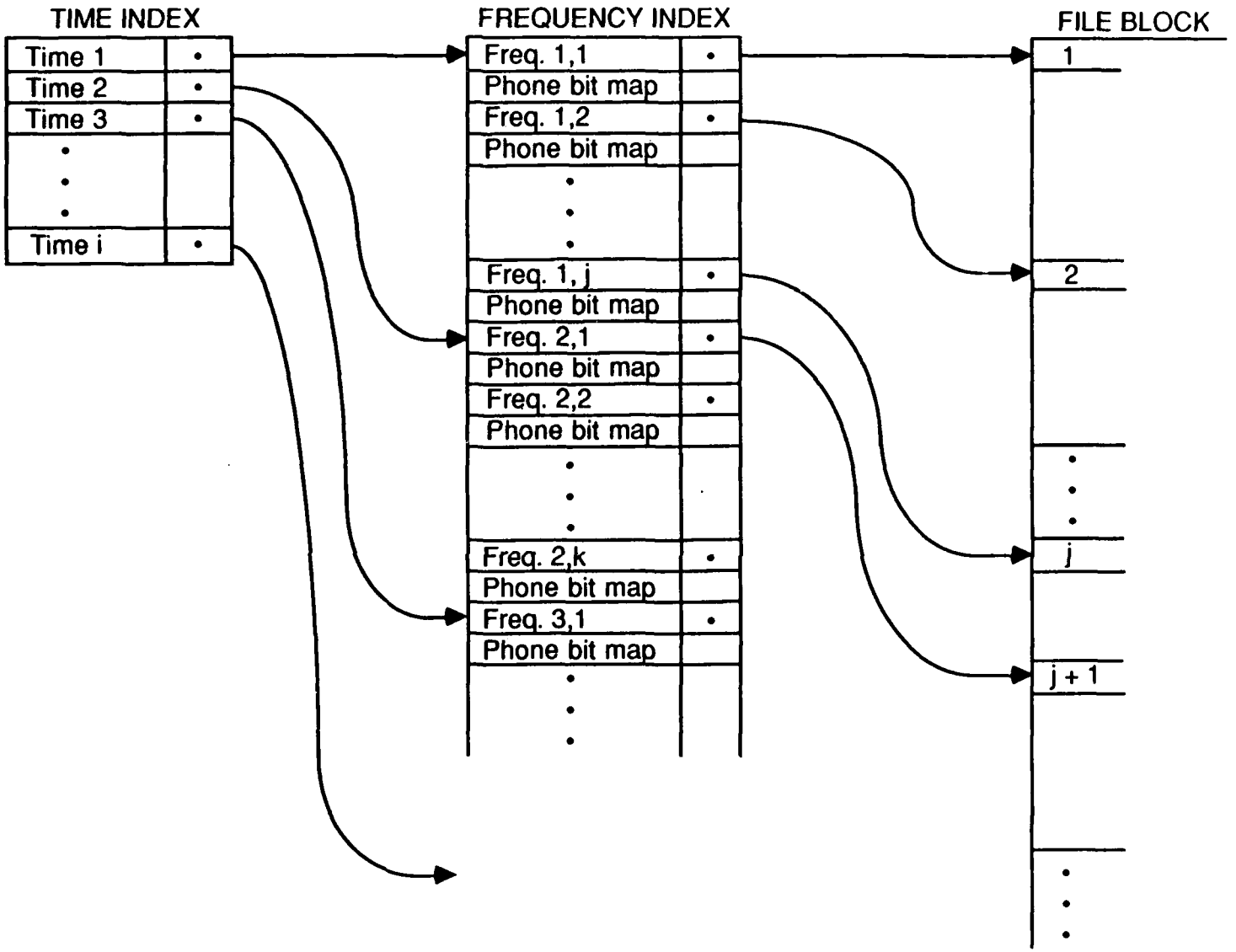

Figure 4-1. Index Structure of the HGI Cross-Spectral Matrix File

frequency bin. Since the CSMs are symmetric, only half of each matrix needs to be saved, yielding the following residency requirement:

$$X = 1/2n^2m$$

where:

X = the number of floating point words required
n = the number of phones, and
m = the number of frequency bins used.

If m is assumed to be approximately two-thirds of the FFT size divided by two, due to clipping, then the expression FFT size/3 can be substituted for m, giving

$$X = 1/6n^2 \text{ x Fft size.}$$

as the storage requirement for the CSMs from a given time interval. Using an FFT size of 32K words, the result is 5.33 billion floating point words, or 21.3 gigabytes, for an HGA suite containing 1,000 phones.

Requirements of this magnitude point up the importance of having adequate CPU, primary memory, and disk storage resources for HGI research. The effects of different configurations on disk access requirements, and therefore on overall processing efficiency, are described in Section 4.3 below.

4.3 Estimated Disk Accesses for Beamformed Data

The number of disk accesses that will be required to support generation of beamformed data will be heavily dependent on this computer configuration being used. The first problem is to fetch the FFT data that is used to construct and accumulate the CSMs. If the data is read from disk files, as will commonly be the case during the HGI research phase, the process will involve using the header indices to skip through the phone data at a given time, picking off data at the desired frequency. If primary memory space is available, it would be more efficient to fetch

data for several CSMs at the same time. For example, if space were available for five CSMs, then data for all five could be gathered in parallel by reading data from five frequency bins from each phone block and storing it in the appropriate slot in each of the five CSMs, before reading data from the next phone. In any case, the required number of disk reads will be the number of time intervals x the number of phones x the number of frequency bins/the batch size, if more than one CSM can be accumulated at a time.

In a real-time system the FFTs would ideally be maintained in primary memory after validity checking, thus obviating the need for disk reads. Using the estimates of FFT size and number of phones given above, this would require 10.7 million words of storage for each time interval in addition to that allocated to CSMs.

The second problem is to accumulate the CSMs until they can be inverted and stored. There are a number of alternative processing approaches. The most attractive is to employ parallelism, i.e., to compute CSMs at different frequency bins in parallel on distributed processors. Another is to take advantage of the virtual memory capability supported by most modern computers, including the VAX family of processors. Using virtual memory, the software to compute the CSMs will treat them as primary memory resident, but the actual data will be swapped to and from disk by the computer operating system. While these operating systems are designed to be as efficient as possible, extreme care will have to be taken to avoid excessive paging and unacceptably slow processing times. The solution lies in learning how the operating system stores and accesses matrix data; i.e., in row-major or column-major order, and designing the CSM formation software to respect that order and process all the data on one page at a time, rather than skipping from page to page.

The task of writing out the inverted CSMs will be relatively straightforward. Once the header indices illustrated in Figure 4-1 have been constructed, the data can be written sequentially in time/frequency order. By the same token, once the beam and beam noise data have been derived, and their file headers created, the files can be written sequentially by time, stave, and if necessary, phone.

4.4 Common Interface Requirements for Beamformed Data

The HGI common interface for beamformed data should serve two primary purposes: to link the data with its associated files and to provide a full description of the parameters used to generate the data. In the former case, associated data will include beam and beam noise data linked to a given set of CSM outputs, and also the names of the FFT and measurement files that served as inputs. In the latter case, parameter information will consist of the CSM processing parameters, including the names of the algorithms used if more than one set is being evaluated, filters used, times, and integration parameters, in addition to the parameter information stored in the associated file headers.

5.0 MANAGEMENT OF PROCESSED AND ANALYZED DATA

The processed and analyzed data that will be generated during HGI research and by an eventual fielded system consists of two basic categories: platform data, such as detections, localizations, tracks, and classifications, and displays of data at all levels of processing. They are being treated jointly in this section because they have a number of elements in common. Unlike the measurement, FFT, and CSM data, which is large in volume, predictable in organization and structure, and generated automatically, the processed and analyzed data tends to be smaller in volume (on a component by component basis), less easily organized by common sets of keys, and generated according to the interests and research methods of a variety of individual analysts. Therefore, they present a more complex data management problem in terms of organization, if not in terms of handling large volumes of intensively processed data.

5.1 Structure of Processed and Analyzed Data

Platform-related data varies in structure and volume depending on how many processing levels it represents. Cross- and auto-correlation surfaces are structured similarly to the CSMs and inverted CSMs from which they are generated, and would typically be indexed by time, array, beam, and frequency range. Thresholded peaks taken from these surfaces would have similar key attributes, with the addition of amplitude or coherence, and be significantly lower in volume. Detections, localizations, and tracks are based on sets of one or more peaks, and tracks

have associated attributes such as position, course, and speed in addition to the generic attributes carried by their constituent peaks. Classifications can include all of the previously mentioned descriptors, but also have an associated platform type, class, and/or name. Typically, they are the lowest of all in volume.

The data base managers developed for the Acoustic Data Base and the Multi-Target Tracker deal with all of the types of data described above via a hierarchically-linked system of peaks, clusters, and tracks. This hierarchy is illustrated in Figure 5-1. While this software contains features that were incorporated to manage very high volumes of data in real time, and may represent "overkill" relative to current HGI research requirements, it could easily be modified to be more interactive and maintain more analyst-specific data. In particular, the file headers should be expanded to include information on processing parameters at all levels, user-specified file names and suffixes, pointers to related displays that have been stored for later analysis, and free-form comments.

Since displays can be generated at all levels of HGI processing, they too tend to be organized more by an individual researcher's needs than by any fixed set of key attributes. The types of displays that were defined during the HGI requirements analysis and the source data they are generated from are given in Table 5-1. To support both review of a set of displays generated by a given HGI processing run and comparison of the same display types from different processing runs, displays should be indexed both by a user-specified set identifier and by type descriptors that are common throughout the HGI files. As shown in Figure 5-2, these descriptors can be linked via auxiliary indices to support rapid retrieval of all displays of a given type. This subject will be discussed further in Section 6, The Common User Interface.

Other information that should be stored in display headers includes user id and date, source data file names and selection parameters, time, stave, beam, phone, and frequency data as appropriate, processing parameters for FFT, CSM, and platform data as appropriate, modeling descriptors as appropriate, and free-form comment text.

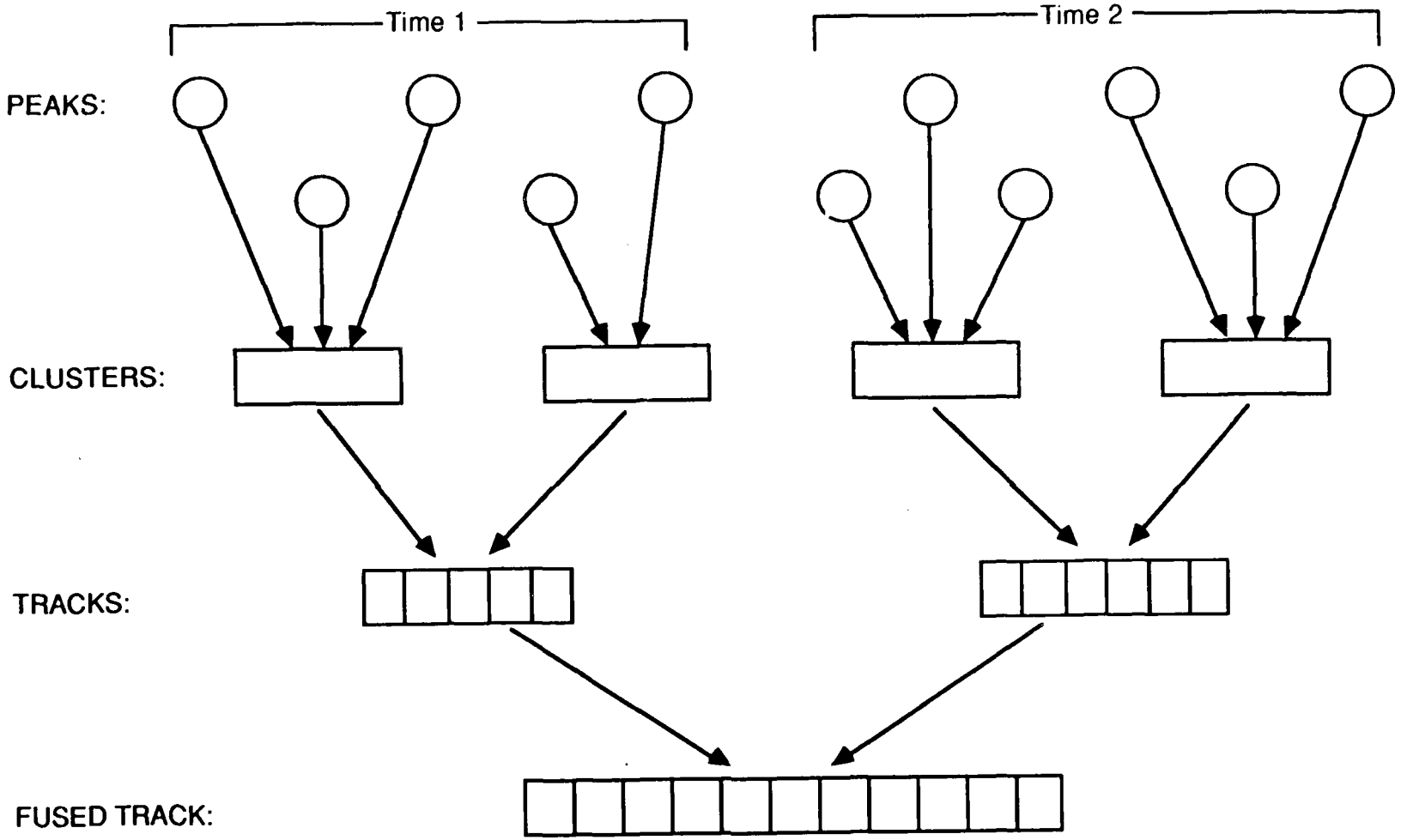

Figure 5-1. Hierarchy of Processed and Analyzed Platform Data Maintained by the Multi-Target Tracker

Table 5-1. Sources of Data for HGI Displays

Display Type	Source
Tracks versus time	Track data, correlated beams, geo or map
FFT frequency/phase vs. amplitude/power	FFTs
A-scans and 3-D frequency vs. time vs. amplitude plots	FFTs
CRT Grams	FFTs
Time series magnitude vs. time	FFTs or original time series
Depression/elevation angle vs. time or frequency	Beam data
Auto- and cross-correlations	CSMs, $CSMs^{-1}$, noise statistics, FFTs
Ray path displays	*Model outputs*
Ambiguity surfaces	FFTs, CSMs, $CSMs^{-1}$, noise statistics
Noise displays	FFTs, noise statistics, environmental data bases
Unwrapped beams	FFTs, beam data

CRT INDEX	
Item	Address
1.2	
2.3	
3.3	

AMBIGUITY INDEX	
Item	Address
1.4	
2.6	

TRACK INDEX	
Item	Address
2.1	
3.1	

FFT INDEX	
Item	Address
2.2	
3.2	

USER SET 1	
1.1	A-Scan
1.2	CRT Gram
1.3	Cross-Correlation
1.4	Ambiguity Surface
1.5	Ray Path

USER SET 2	
2.1	Track versus Time
2.2	FFT
2.3	CRT Gram
2.4	Ray Path
2.5	Noise Display
2.6	Ambiguity Surface

USER SET 3	
3.1	Track versus Time
3.2	FFT
3.3	CRT Gram
3.4	A-Scan
3.5	Ray Path
3.6	Cross-Correlation

A-SCAN INDEX	
Item	Address
1.1	
3.4	

CORRELATION INDEX	
Item	Address
1.3	
3.6	

RAY PATH INDEX	
Item	Address
1.5	
2.4	
3.5	

NOISE INDEX	
Item	Address
2.5	

Figure 5-2. Index Structures for Processed and Analyzed Data Display Types

5.2 Size and Residence Requirements of Processed and Analyzed Data

With a few exceptions, the sizes of display and platform-related data are not large, as is shown in Table 5-2. Thus, it is reasonable to store them in disk files for subsequent review and analysis, rather than regenerating them each time. Since these estimated sizes represent stored screens, their disk and primary memory residence sizes are the same; i.e., each display will require the same amount of primary memory when read in for display purposes.

A-scans, three-dimensional frequency versus time versus amplitude plots, and CRT grams represent exceptions to the approach put forward above. Although the sizes given in Table 5-2 are based on the best available data storage and compression techniques, they are still several orders of magnitude larger than the storage requirements of the other displays. While these displays might be saved temporarily during an analyst's working session, or in preparation for a demonstration, in general it would be better practice to regenerate them on the basis of need.

While it is beyond the scope of this report to discuss the sizes of HGI processing modules, as opposed to data bases, we do have reasonably accurate estimates of the primary memory storage requirements of display generation software that are derived from extensive previous experience. Since it makes sense to have this software be part of the HGI processing suite that is available to all users, just as the data bases will be available, we are including these estimates as part of this section. Display software for an application such as the HGI program falls into two basic categories: code to manipulate data in preparation for display, and code that actually generates graphics on the screen. The estimated lines of source code and kilobytes of stored object code for software in these categories are shown in Table 5-3.

A final element to be considered in the size and residence requirements of processed and analyzed data is platform-related data that is not generated for immediate display purposes. Such data would include thresholded peaks, clusters, localizations, and track data other than position, course, and associated uncertainties. Assuming eight words per peak and 20 words per track, storage of 1,000 peaks would require 32Kb, and storage of 50 tracks would require 4Kb on a VAX. These space requirements are not large enough to cause problems unless users fail to delete or archive old data on a fairly regular basis.

Table 5-2. Estimated Sizes of Stored Display Data

Display Type	Size (K bytes)
Tracks vs. time	1-4
FFT frequency/phase vs. amplitude/power	2-4
A-scans and 3-D frequency vs. time vs. amplitude plots	300
CRT Grams	1,000
Time series magnitude vs. time	2-8
Depression/elevation angle vs. time or frequency	2-4
Auto- and cross-correlations	2-8
Ray path displays	2-4
Ambiguity surfaces	4-8
Noise displays	1-2
Unwrapped beams	2-4

Table 5-3. Estimated Sizes of Display Generation Software

Display Pre-processing Code	Source Code (Lines)	Object Code (Kb)
FFTs (interpolation, etc.)	100	5
Correlations	600	20
Matrix Manipulations	600	30
Filtering	250	10
Miscellaneous	200	10
Total	1,750	75

Graphic Display Code	Source Code (Lines)	Object Code (Kb)
X,Y Plots	500	15
3-D Plots	700	30
Grams	500	15
Library Routines	200	10
Miscellaneous	500	20
Plotting Libraries	unknown	350
Total	2,400*	440

*Without plotting libraries

5.3 Estimated Disk Accesses for Processed and Analyzed Data

Disk accesses for processed and analyzed data are difficult to estimate because they are highly dependent on how the user chooses to store and annotate his data. If the hierarchical scheme used by the Multi-Target Tracker and illustrated in Figure 5-1 is employed for platform-related data, two or three accesses to an index stored in primary memory plus one direct disk access for each block of data will generally be required.

Access to stored displays will involve two processes: identifying and locating the data interactively, and then reading the data from disk. Identification might be as simple as entering the name of a user-defined set of displays, or it might involve browsing through file headers until the desired data is found. In the latter case, the index structure by display type described in Section 5.1 would speed up the process of showing the user what displays of a given type are available. Once the desired display has been identified, the stored screen can be read from disk with a single read operation.

5.4 Common Interface Requirements for Processed and Analyzed Data

The common interface requirements for processed and analyzed data can be met by user adherence to a set of file naming conventions that specify the types of display and platform-related data that are stored and the set relationships among them. This structure will be supplemented by information in the individual file headers, such as was described in Section 5.1, that can be scanned to identify displays generated from a particular processing run, modeling effort, or input measurement data base. Menu selection procedures should be developed to make it easy for the user to indicate the type of data he is seeking and examine the alternatives that are available to him.

6.0 COMMON USER INTERFACE FOR THE HGI DATA BASE MANAGER

In this section, a common user interface will be defined that can serve as an umbrella function that will integrate and manage the file types and structures that were presented in Sections 2 through 5. As was mentioned in the introduction to this document, the HGI data manager

should provide sufficient structure to maintain configuration control of HGI processing and data files and allow users to work efficiently and to compare results. At the same time, it cannot be overly rigid, because many HGI functions and algorithms are not fully defined and researchers need "room" to experiment with parameters and algorithms without having their selections and results frozen into the HGI system. In the sections that follow, we present a high-level design for the HGI data manager that is aimed at satisfying the above requirements.

6.1 HGI User Group Definition and File Management

A command level structure that takes advantage of the user group, directory, and file naming conventions of computers such as the VAX is a reasonable and cost effective starting point for defining the HGI data manager. This approach has been adopted because it allows users to experiment with HGI processes in "private" files and evaluate results before choosing to make them "public" files shared by the HGI user group. We envision that an HGI user group will be defined that permits access to software and data files in included directories, on a user-specified read only or read/write basis. A major advantage to this approach is that general-purpose software, such as display generators, standard FFT and other signal processing algorithms, etc., can be included in the group files as they mature, thus making it easy for users to share them rather than having to find, or write, their own code. By the same token, synthetic and raw measurement files, validated versions of these files, FFT outputs from these files, stored displays, etc., can exist as a common, read only library for group use. SPARS outputs, containing environmental and performance prediction information on the areas for which measurement data is available, will also be included. Use of the VAX (or other computer) group and file definition, file protection, and file copying utilities, with which the HGI research group is familiar, is a much more cost effective approach than developing a new system.

6.2 HGI Command Level Software

We propose that command level software tailored to HGI requirements be developed that would reside within the user group and serve as a means of managing and integrating HGI data. The software would function in much the same manner as user-defined command files, which support execution of a number of system level commands with a single execute instruction,

except that it would be considerably more flexible and powerful. The user would initiate HGI processing via the command level software whenever he wished to either use software and data files included in the user group or move newly created software or files into the group.

While the command level functions will undoubtedly be refined during development and in response to feedback from users, the basic set of commands can be assumed to be as follows:

- DIRECTORY <file type>
- DISPLAY <file>
- EDIT/CREATE <file>
- RUN <file>
- SCAN <file group>
- PRINT <file>
- QUIT

These functions are discussed in more detail in the paragraphs that follow.

The DIRECTORY <file type> option is already supported by standard operating systems, but it will be more convenient for HGI users to be able to examine and select files from a directory within the HGI command level software than to have to look through a directory and remember or write down the files of interest before entering the HGI command software. Inclusion of this function will also allow researchers to examine and select new files without having to exit and re-enter the HGI environment.

The DISPLAY <file> option will be selected when the user already knows the name of the file he wishes to examine. DISPLAY will apply to either data files or source code files, and will have the standard page forward and backward functions to support browsing.

The EDIT/CREATE <file> option will be chosen when the user wishes to edit an existing user group file, create a new file, or include a privately created file in the user group. Definition of the appropriate functions for this command is complex, and will require refinement as user feedback becomes available. In general, user group files should be read only; i.e., they should be protected from changes entered by individual users. However, there will always be exceptions. While the HGI processing suite is being developed, software may need to be

included that is not fully transitioned into the HGI protocols. For example, existing software may have embedded input and output file names that need to be changed when new names are desired. Eventually, all such file specifications should be part of the RUN option discussed below. Also, there will be cases when a change in source code or a data file is deemed desirable by a consensus of HGI users, and supporting this capability via the EDIT function is the only efficient option. A long-term HGI goal should be to restrict this function to occasional use by a single authorized user who has been designated as the Data Base Administrator.

Selection of the EDIT/CREATE option will always invoke the standard editor supplied by the computer system in use. This option can also be used when a researcher wishes to create a new file for HGI group use. After entering, debugging, and testing his source code or data file, the user will be presented with a series of standard menus when he elects to save his file as part of the user directory. These menus will interact with the file open statements, if the new file is source code, to ask the user to define the files according to the adopted HGI file naming conventions. For both source code and data files, the software will determine from the file names what indices, data structures, and header information needs to be created and will prompt the user with appropriate menus and selection options. In this way, consistency and configuration control will be maintained within the HGI user group, but it will be highly dependent on the willingness of HGI researchers to adhere to the conventions. It is the hope of the author that the menus will be sufficiently well designed, and the resulting suite of HGI software sufficiently easy to use, that the researchers will willingly cooperate.

In the case where the user wishes to move a privately-created file, or a SPARS file, into the HGI processing suite, he will go directly into the menu-driven file specification process described above.

The RUN <file> option will be selected when a user wishes to execute an existing file of object code. In the mature HGI processing suite, selection of this option will automatically generate menus asking for the names of input and output files used by the program. Default names will be given so that the user will not have to re-enter appropriate names that have been entered previously. After the run has been completed, the user will be given three options: to save his output files with appropriately filled in header information, to move his output to a private directory, or to delete it from the system. If the first option is chosen, menus will be presented with the current header contents as default values; in this way, the user will only have to enter

new information. Adherence to this protocol will ensure that all data files in the user group are properly documented.

The SCAN <files> option will be selected when the user wishes to browse through the headers of one or more files before choosing items to be displayed, printed, run, or used as input to a run. The user will be able to use standard wild-card notation on the file name or suffix in order to step through a sequence of files. The SCAN function will contain menus of sub-options, such as NEXT, MORE (detail), and SELECT, to enable the user to navigate through the selected files with ease.

File naming conventions will also be used in support of the scan function. Files will be given a base name that reflects the file type; i.e., RAWU (raw unvalidated), RAWV (raw validated), SYNT (synthetic), SPRS (SPARS output), FFTS (FFT output), BMFD (beamformed), ASCN (A-scan), CRTG (CRT Grams), and so forth. Users will assign suffixes that are meaningful to them. Under this scheme, a user could specify all the displays from a given run by entering *.MYSUFFIX, and all the displays of a given type by entering ASCA.* or CRTG.* .

The <PRINT> and <QUIT> functions are self explanatory.

6.3 HGI Data Manager Software

While development of the command level software will require careful attention to detail and the management of considerable complexity in support of what appears to be a clear and easy to use man/machine interface, the real sophistication will lie in the data manager software. On the basis of user-entered key specifications, again prompted by menus, the data manager will construct and maintain all file indices. Whenever a file is specified as input to a program run, the data manager will use the indices to make the program run efficiently and correctly. By the same token, when a file is used as output the data manager will structure the file correctly and update all indices as the file is being written. The embedded calls to the data manager will be designed to be as simple as possible, so users can incorporate them in their software without having to understand their underlying complexity. In addition to the basic read, write, and update functions, a data deletion capability should be implemented for selectively deleting old or invalid data.

The detailed design of the data manager functions are beyond the scope of this report, but the design is based on well-understood techniques for index construction and maintenance. As mentioned previously, features of existing ARC/STIC software, such as SPOOLMAN and FAMEAS, can be incorporated in the HGI data manager as appropriate. The Acoustic Data Base for thresholded peaks and the Multi-Target Tracker data management software can be modified for HGI use. Common display generation software will be a very important component of the HGI suite; GEOPKS and GEOPLOT are good examples of display utilities that have been used extensively, and a great deal of other display software already installed at the ARC/STIC can easily be modified for HGI use.

With respect to commercially available Data Base Management Systems (DBMS), almost any product could be used for high-level file management and configuration control, including INGRES, ORACLE, or assorted VAX products. However, none of these systems integrate well with the computationally intensive numeric processes that will be run on HGI data. They offer layers of man/machine interfaces that are not required for HGI research, which basically needs easy-to-use file manipulation commands, straightforward index structures, and efficient embedded read, write, update and delete functions. Use of these systems typically requires considerable training and experience in a specialized user language; in general, HGI researchers would not want to make such an investment of their time and would not feel that the result was cost effective.

The file designs, index structures, and command level software presented in this report are aimed at providing users with a working environment that is flexible enough to meet their needs and also creates a structure for research results that will be a benefit to the HGI community. Due to the legacy of existing software that can be drawn upon and a good understanding of HGI research issues and goals, the software can be developed at reasonable cost and in a relatively short period of time. We believe that implementation of the data manager design proposed in this report would provide long term benefits to the HGI research program.