Geophysical Data Base Variable Resolution (GDBV):
An Object-Oriented Database for Dynamic Geo-Acoustic Data Storage

Mr. Chad A. Steed, U. S. Naval Research Laboratory
1005 Balch Blvd.
Stennis Space Center, MS, 39529 USA
csteed@nrlssc.navy.mil

Mr. David W. Harvey, Naval Oceanographic Office
1002 Balch Blvd.
Stennis Space Center, MS 39529 USA
harveyd@navo.navy.mil

Mr. Kim A. Koehler, Neptune Sciences, Inc.
4250 Pacific Highway, Suite 219
San Diego, CA 92110 USA
kim.koehler@navy.mil

Mr. Bruce Northridge, OAML Chairman
The Naval Meteorology and Oceanography Command
1100 Balch Blvd.
Stennis Space Center, MS 39529 USA
northridgeb@cnmoc.navy.mil

Abstract-The Geophysical Data Base – Variable resolution (GDBV) is a modern, object-oriented data store that is designed to accommodate the dynamically derived parameters of the Geo-Acoustic Inversion Toolkit (GAIT). Sponsored by the Oceanographer of the Navy (CNO NO96) via PEO (C4I and Space) PMW-155, GAIT/GDBV is a Through-the-Sensor (TTS) program that includes a flexible data model for the assimilation of data at local, regional, and global levels of operation. In addition to its dynamic capabilities, GDBV also includes support for historical database roles similar to the Naval Oceanographic Office's (NAVOCEANO) Low Frequency Bottom Loss (LFBL) database.

In order to demonstrate its highly extendable design, this paper explores GDBV's database format and data model. In both the historical and dynamic capacity, GDBV must be capable of evolving with new system specifications. GDBV's multiple levels of organization and object-oriented implementation provide an efficient solution for these requirements.

In addition to its dynamic operational requirements, the GDBV database will accommodate the parameter definitions from each of the following Oceanographic and Atmospheric Master Library (OAML) databases: High Frequency Bottom Loss (HFBL), LFBL, MIW Sediments and Roughness, LFBL'S N-Layer dataset, and the Applied Physics Laboratory of the University of Washington's (APL-UW) GeoAcoustic Bottom Interaction Model (GABIM) Bottom Back Scatter (BBS) database. Support for these static databases enables the future evolution of each individual database into a single broad database that provides a complete description of the ocean bottom. The complete parameter set of GDBV is presented along with a physical representation of the parameters.

The overall data flow is very similar to a preceding TTS system, PUMA-TEDS. This working example of GDBV in GAIT reinforces the portability of the database design and the benefits of generic TTS software design.

GDBV is a modern database that provides a lightweight, highly portable data store with sophisticated features traditionally offered in large scale database management systems. In addition to fulfilling the requirements of the expanding GAIT algorithms, the generic implementation of GDBV will provide a valuable tool for a wide range of environmental data types, including those of future TTS programs.

I. INTRODUCTION

To support the algorithms of the Geo-Acoustic Inversion Toolkit (GAIT), a standardized, worldwide database for geo-acoustical information must be established. Currently, the Naval Oceanographic Office's (NAVOCEANO) Low Frequency Bottom Loss (LFBL) database is the Navy's only operational, geo-acoustic database with worldwide coverage. Since LFBL version 9.0, a high resolution, n-layer data model has been employed for certain areas of the world ocean. The starting point for this effort is to combine LFBL, n-layer LFBL, and the Applied Physics Laboratory of the University of Washington (APL-UW) bottom backscatter parameters into a common database called the Geophysical Data Base Variable resolution (GDBV) [5].

A. GAIT Background
The PMW-155 sponsored Shallow Water Issues for METOC Support (SWIM) Report recommends that the Oceanographer of the Navy (NO96) approve a program to develop and transition inversion techniques to the Fleet. In general, SWIM recommends the development of algorithms that will feed a three-dimensional, variable-resolution worldwide geo-acoustic database that supports shallow water full wave propagation loss models [5, 8]. To address this directive, the GAIT development team has been assembled with representatives from government agencies, private industry, and academia. Within the GAIT team, the GDBV Data Architecture Working Group has been created to specifically address the formulation and implementation of the GAIT database, GDBV [9].

B. PUMA Background
GDBV follows the integration of a similar TTS system that facilitates the assimilation of bathymetric data collected by the Precision Underwater Mapping (PUMA) system on Fleet attack submarines. The PUMA system defines three levels of near real-time decision support (local, regional, and global) through the NAVOCEANO Digital Bathymetric Data Base – Variable resolution (DBDB-V) supplemental database as well as periodic

Oceanographic and Atmospheric Master Library (OAML) DBDB-V product upgrades [11]. This supplemental database co-exists with the historical (OAML) DBDB-V database.

The historical and supplemental databases utilize the PUMA/DBDB-V common format, Variable resolution GRID (VGRID). VGRID is a generic, geographic format for storing grid data that was developed primarily to meet the needs of PUMA grid data storage [10]. Due to the numerous high level similarities between the PUMA and GAIT/GDBV systems, GDBV development efforts can utilize much of the technological advances and lessons learned from the PUMA project [9].

C. LFBL Version 11.0 Data Model

Since NAVOCEANO's LFBL Version 11.0 is the Navy's sole operational, global database for geo-acoustic information, it is an appropriate starting point for the design of GDBV. It is necessary to achieve a conceptual understanding of the LFBL data model to facilitate the following description of the GDBV data model.

The LFBL data model utilizes the concept of a geographic coverage entity to define the bounds of individual datasets that are stored in the database. In LFBL Version 11.0, there are four such entities: world, high resolution Barents and Kara Sea, high resolution Yellow Sea – East China Sea, and high resolution South China Sea. The world coverage is a one layer dataset with a spatial resolution of 5 minutes of arc while the other datasets are n-layer (see Fig. 1) where the spatial resolution is 12 seconds of latitude and longitude [2].
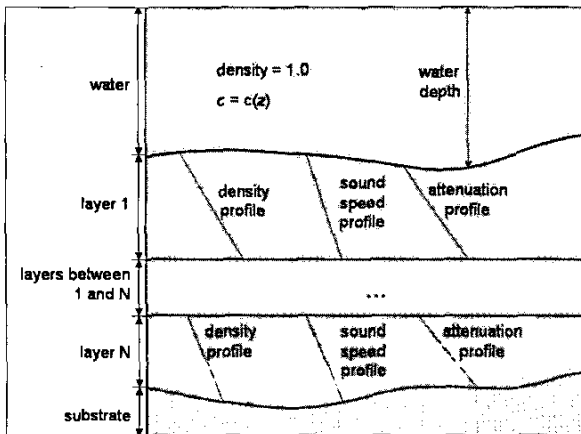

Fig. 1 N-Layer Parameter Structure

For each coverage entity stored in LFBL, the database stores a corresponding grid structure, called a province map, where each grid cell contains a province identifier. These province identifiers are used to retrieve specific geo-acoustic parameters for the cell via a province lookup table. At NAVOCEANO, scientists use a Geographic Information System (GIS) to develop the LFBL provinces from survey data. When a new version of LFBL is built, the GIS data stores are exported to a grid structure for inclusion in the LFBL database.

Each province map has an associated province lookup table that stores the defining parameters for each layer in each of the provinces. The province identifier stored in the province map grid is used to locate all the table records that describe the province's layers.

The LFBL database also stores a layer thickness map for each layer in a coverage entity. This map is stored as a regularly spaced grid where each grid cell corresponds to the thickness of the layer at the grid cell's geographic location. In some coverage entities, this parameter is stored in seconds to indicate two-way travel time (TWTT) and in others it is stored in meters [2].

D. GDBV Physical Description

Like NAVOCEANO's LFBL database, GDBV will store geo-acoustic parameters for the world ocean bottom. The bottom description in GDBV will include information about the water-sediment interface and the sediment layers that lie beneath this interface. The sediment layers represent a high resolution view of the ocean bottom and will be numbered sequentially from the top to the nth layer.

GDBV will also store non-layered parameters that apply to the water-sediment interface, the entire sediment volume (the sediment volume regarded as a single layer), and the sediment basement. When compared to the n-layer model (see Fig. 2), the non-layered parameters represent a low-resolution description of the ocean bottom that is included, for the most part, to support legacy databases [9].
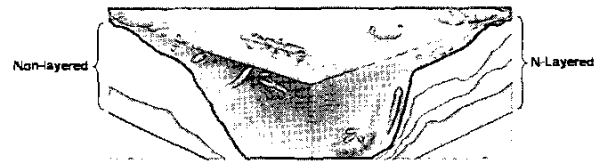

Fig. 2 GDBV Parameter Layers

II. GDBV Architecture

In this section, the GDBV architecture is presented in terms of the database format, data model, and parameters. In general, the GDBV data model is intended to accommodate the n-layer data model from NAVOCEANO's current version of LFBL for both in-situ (dynamic) and static (historic) data storage. Single layer parameters, such as those of the worldwide LFBL coverage, are stored in the database as singular objects while layered parameters are stored by a series of layer objects where each layer may provide different values for a constant set of parameters.

A. Database Format

The database format is the database or file system used to make the data persistent on the local or remote file system. The GDBV database format will build on the layered-API approach of the VGRID format developed for PUMA. VGRID can be thought of as an interface between low level data storage and the product API [10]. The

133

product specific API provides a means for providing customized algorithms that access the data through the VGRID API. For example, LFBL's "pinch out" algorithm is a product specific function that makes sense in extractions from a LFBL n-layer dataset but not for extractions from a DBDB-V bathymetric dataset. Similarly, the DBDB-V APIs include a feathering algorithm to smooth data along the boundaries of different resolution datasets which is not directly applicable to LFBL datasets. The layering of APIs gives a data production center the ability to utilize a common geographic file format and still provide customized extraction and ingest algorithms.

Designed to store multiple-resolution, geographic grids from the PUMA system, the PUMA version of VGRID is implemented in the Hierarchical Data Format version 5 (HDF5). HDF5 provides a high level interface for efficiently storing scientific data and allows standardization through the definition of a file specification [4]. VGRID is an example of a file specification that uses the HDF5 API to organize and store geographic data. Some of the main features offered by the HDF5 format are: built-in compression options, custom data filters, internal caching, and data attribution [3].

Since HDF5 is a file format and not a database management system, it does not provide multiple user features such as file-locking, transaction-based processing, commit levels, or rollbacks. Although the actual HDF5 file is considered platform independent, a platform specific HDF5 API must be available to the external application to access the file's data. Because the API is implemented in the C and FORTRAN programming languages, the HDF5 API must be built for each of the platforms supported by the HDF5 developer, the National Center for Supercomputing Applications (NCSA). Although NCSA also distributes a Java™ HDF5 API, it is built around the platform-dependent, C-based HDF5 library via the Java™ Native Interface (JNI) [4].

GDBV demands a highly flexible database system with more platform independence than the HDF5 file format can currently offer. Although the HDF5 format is well suited to static datasets (such as OAML approved database products), it does not provide enough extensibility for dynamic systems such as GAIT.

A viable alternative to the HDF5-based VGRID file format is the . Ozone Object-Oriented Data Base Management System (ODBMS). Since Ozone is an ODBMS, it includes several sophisticated database access features such as safe multiple-user access, transactions, import and export functionality, remote database connectivity, commit levels and rollbacks, and internal clustering. Ozone is written entirely in the Java™ programming language so both the database format and software are completely platform independent [7]. That is, the Ozone API can be compiled once on a particular platform and executed on any other platform that has a functional Java™ Virtual Machine (JVM). Furthermore, Ozone is freely available (open source license) from the project website (http://www.ozone-db.org) and has been successfully utilized by the NRL-SSC developed Geospatial Information Data Base (GIDB) for the past three years. The use of Ozone in GIDB and the fact that it will also be used in the much anticipated Tactical Environmental Data Services (TEDServices) development provide substantial evidence of its robustness and suitability for use with the GDBV system.

By organizing the data around the actual entities as opposed to the functions being processed, object-oriented database systems, like Ozone, combine the speed of hierarchical and network approaches with the flexibility of a relational approach. In the relational structure, each entity is defined in terms of its data records and the logical relations that can be interpreted between the attributes and their values. In an object-oriented database, data are defined in terms of a series of unique objects which are organized into groups of similar occurrence (known as classes) according to a natural structuring [1]. This natural structuring provides a more logical basis for the storage of geospatial data than hierarchical, relational, or network oriented database systems alone.

The Ozone ODBMS represents a modern database management system implemented in the Java™ programming language. Because it stores objects as Java™ classes, an Ozone database can store the data and the methods for accessing and manipulating the data. Consequently, data stored in the database are no longer passive but form an active object with its own algorithms for utilizing its instance variables. The ability to directly associate data with the algorithms for manipulating them can potentially revolutionize the process of database distribution which is often a complex process with flat-file based databases and platform-specific API binaries.

The plastic nature of GDBV is more efficiently addressed through the utilization of an extendable database system like Ozone. The HDF5 file format in VGRID simply does not provide the dynamic extensibility or platform independence needed for the evolving GDBV definition.

B. Data Model

The database model is the conceptual representation of how the data is organized within the database format. The formulation of the GDBV database model is the result of merging the LFBL n-layer data model with the PUMA VGRID data model and some aspects of NIMA's Vector Product Format (VPF) standard. The result is more levels of organization than VGRID's current offering and a more generic design than the LFBL legacy file format by incorporation of some of the high-level hierarchy of the VPF.

To support dynamic data ingest of non-homogeneous data points, the new data model will include the capability to store irregularly or regularly spaced data points in a dynamic data structure. This structure will support the in-situ role of GDBV and a regular grid structure will be included to provide historic database capabilities. The dynamic structure will also be available as an alternative means for storing historic data in GDBV. The dynamic

structure will be stored as tiled point collections in the database in a manner similar to the tiling scheme developed for the rectangular grids of the VGRID data model (see Fig. 3).
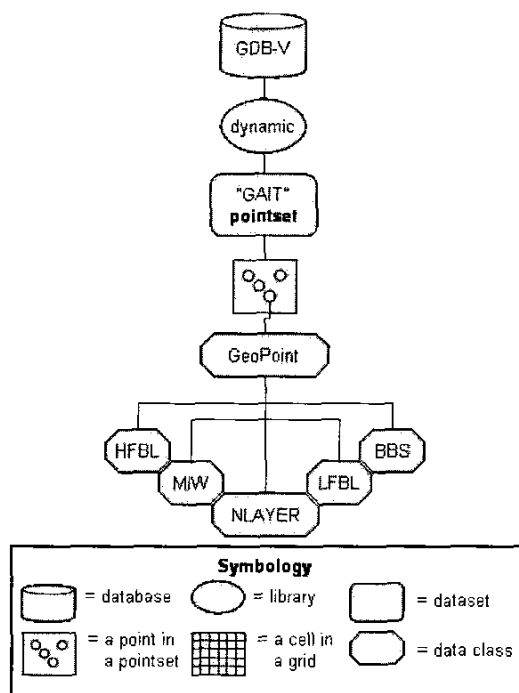

Fig. 3 Proposed Dynamic GDBV Model

In addition to topological organization changes, the GDBV data model will also provide more levels of organization than the VGIRD design in PUMA. The new data model will provide three levels of organization within each database. The database is the top-level object that represents a single Ozone database directory on disk. The database will include a general set of metadata fields that are attributed to the contents of the entire database. The data model will also include the ability to define, populate, and extract specific metadata for each level of organization in its hierarchy.

A database object will store library entities at the next level of the hierarchy. The GDBV library is an entity that allows a related set of parameters to be grouped independent of other parameter sets stored in the database. For example, the GDBV data model will have a dynamic library to store the GAIT derived data. If NAVOCEANO chooses to utilize the GDBV format for consolidation of its geo-acoustic databases, a historic library will be used to store static OAML datasets in order to ensure separation of the historic and dynamic data within the same GDBV database.

The new model also allows tiled datasets to be created within a library object. A tiled dataset stores homogeneous or non-homogeneous data points and will be available for use in either the historic or dynamic libraries. For most datasets in the historic library, a tiled grid structure should be used for database storage. The dynamic GDBV datasets

will store non-homogeneous (scattered) points in a point set data structure. A point set or grid's storage properties, such as tile size and areal extent are defined when it is created using the GDBV API. In the historic library, a separate grid can be created to store each of the legacy OAML databases and the historic BBS database (see Fig. 4). If future evolution of GDBV requires the storage of a multiple resolution database, such as OAML DBDB-V, a separate grid object can be created for each of the resolution sets stored in the original database. Tiles may be written to or extracted from the database via the database's ingest and extraction interfaces, respectively.
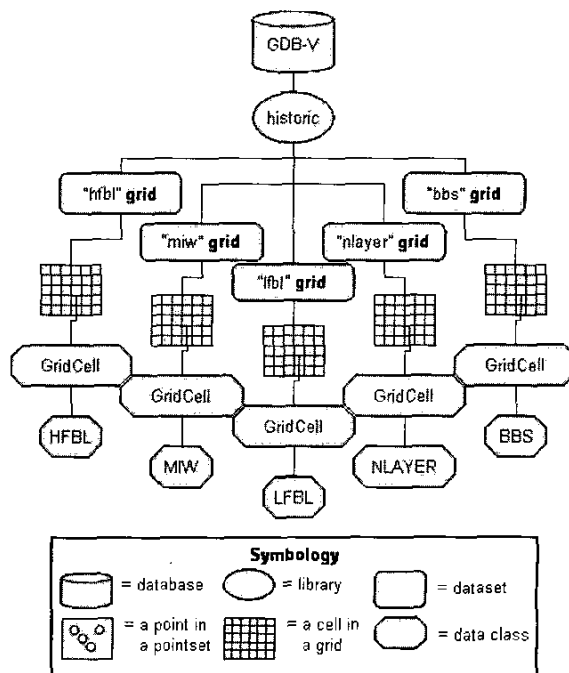

Fig. 4 Possible Historical GDBV Model

III. GDBV Parameters

By defining the GDBV database architecture, development of applicable Tactical Environmental Data Services (TEDServices) segments and the historical and dynamic data merging algorithms can commence in parallel to the inversion technology algorithm research. In addition, the GAIT developers (including BQN-17 and AQS-20 inversion efforts) can work towards a common set of universal output parameters that will parameterize both bottom loss and bottom backscatter. Furthermore, the early establishment of this database will allow Transmission Loss model developers to begin modifying their code to accommodate these parameters [5].

A. Object Oriented Definitions

The atomic objects of the GDBV grid and point set data structures are the GridCell and GeoPoint, respectively. As shown in Fig. 5, a GeoPoint represents a point in a dynamic point set and carries a coordinate pair to denote its location

in the world. Similarly, a GridCell is associated with the intersection of a particular row and column in a grid dataset.
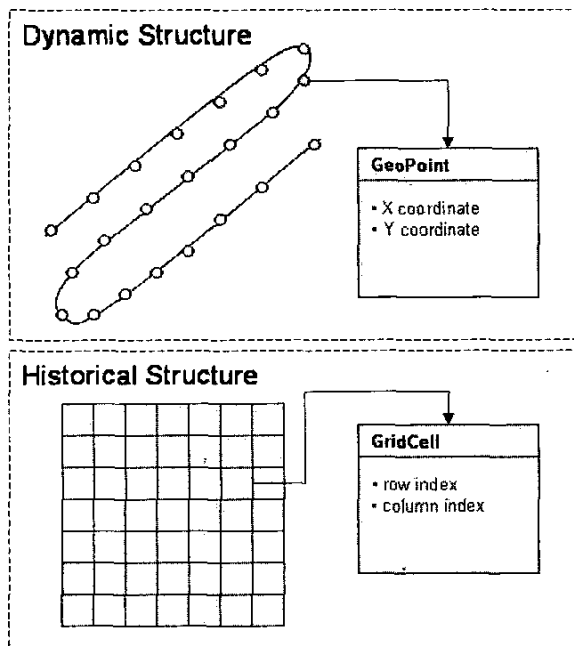

Fig. 5 GDBV Feature Objects

Both of these objects extend the GeoData class that is included in the GDBV API. By extending the GeoData object, GridCell and GeoPoint inherit its instance variables and methods. The GeoData object is designed to support an n-layer description of the ocean bottom. The GeoData object's first layer can be used to store the non-layered data parameters that are also a part of the GDBV definition.

As shown in Fig. 6, the GeoData class contains a standard Java™ Vector (java.lang.Vector) object which may contain zero to N number of Layer objects. Representing a layer of sediment in the ocean bottom, a Layer object is referenced by a layer identifier and contains a Vector of Field objects. The Field objects are defined via a name-value pair where the name can be used to reference a FieldDefinition object that is defined within the top level database object.

The standard GDBV database fields are defined automatically whenever a new database is created. These fields are listed in the following section. Fields that are not included in the GDBV parameter specification can be accommodated by defining a new FieldDefinition with the GDBV administration software or the GDBV API. Once the new field definition is created, the GDBV ingest software can write the new field types to the database. The ability to extend the type of data stored in the database supports the evolving GDBV definition and re-use with future TTS systems.
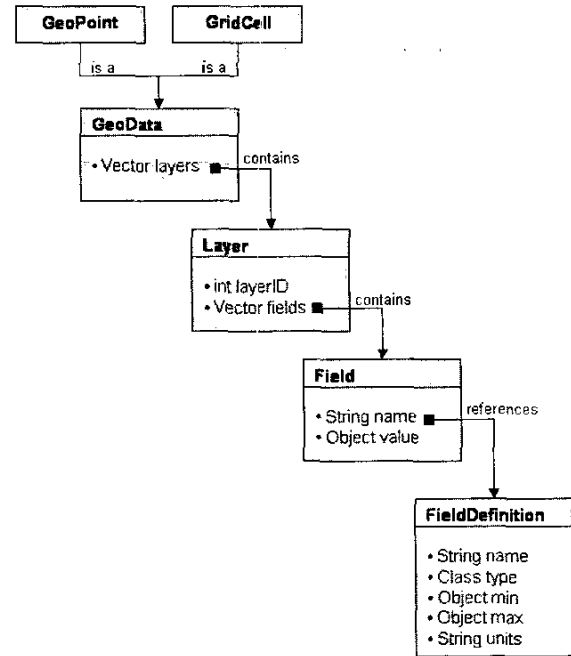

Fig. 6 Data Object Relationships

B. GDBV Parameters

In this section, the GDBV parameters are listed along with the units for each parameter in Table 1. The parameters are organized according to their corresponding legacy OAML database.

Table 1 GDBV Parameter Definitions [9]

Legacy OAML Database	Parameter Name	Units
HFBL	HFBL code	N/A
LFBL	ratio of sediment compressional wave speed to water sound speed	N/A
LFBL	thin-layer thickness	m
LFBL	thin-layer density	kg/m³
LFBL	sediment-surface density	kg/m³
LFBL	sediment compressional wave speed gradient	s⁻¹
LFBL	sediment compressional wave speed profile curvature parameter	N/A
LFBL	sediment-surface compressional wave attenuation factor	dB/m/kHz
LFBL	sediment compressional wave attenuation gradient	dB/m²/kHz

136

LFBL	basement reflection coefficient	N/A
LFBL	compressional wave attenuation frequency exponent	N/A
LFBL	thickness of stochastic layers	m
LFBL	compressional wave attenuation of stochastic layers	dB/m/kHz
LFBL	density of stochastic layer #2B	kg/m^3
LFBL	basement critical angle	deg.
LFBL	stochastic layers' two-way travel time to reflecting horizon	s
LFBL	frequency range	Hz
MIW	bottom roughness	N/A
MIW	surface sediment type (enhanced)	N/A
GABIM BBS	water-sediment interface roughness spectral strength	$m^{(4-\gamma)}$
GABIM BBS	water-sediment interface roughness spectral exponent (γ)	N/A
GABIM BBS	sediment-basement interface roughness spectral strength	$m^{(4-\gamma_b)}$
GABIM BBS	sediment-basement interface roughness spectral exponent (γ_b)	N/A
GABIM BBS	sediment volume parameter	N/A
GABIM BBS	sediment volume spectral strength	$m^{(4-\gamma_3)}$
GABIM BBS	sediment volume spectral exponent (γ_3)	N/A
GABIM BBS Layered	sediment volume spectral strength	$m^{(4-\gamma_3)}$
GABIM BBS Layered	sediment volume spectral exponent (γ_3)	N/A
GABIM BBS Layered	sediment volume aspect ratio	N/A
GABIM BBS Layered	sediment volume fluctuation ratio	N/A
GABIM BBS Layered	sediment volume cross spectrum fluctuation ratio	N/A
GABIM BBS Layered	sediment-sediment interface roughness spectral strength	$m^{(4-\gamma)}$
GABIM BBS Layered	sediment-sediment interface roughness spectral exponent	N/A

	(γ)	
N-Layer	total number of layers	N/A
N-Layer	sequential layer numbering scheme	N/A
N-Layer	layer thickness	s (TWTT)
N-Layer	compressional wave speed at the top of the layer	m/s
N-Layer	density at the top of the layer	kg/m^3
N-Layer	compressional wave attenuation at the top of the layer	dB/m/kHz
N-Layer	shear wave attenuation at the top of the layer	dB/m/kHz
N-Layer	shear wave speed at the top of the layer	m/s
N-Layer	compressional wave speed at the bottom of the layer	m/s
N-Layer	density at the bottom of the layer	kg/m^3
N-Layer	compressional wave attenuation at the bottom of the layer	dB/m/kHz
N-Layer	shear wave attenuation at the bottom of the layer	dB/m/kHz
N-Layer	shear wave speed at the bottom of the layer	m/s
N-Layer	compressional wave attenuation factor	$dB/m/(kHz)^n$
N-Layer	compressional wave attenuation gradient	dB/m/kHz/m
N-Layer	compressional wave attenuation frequency exponent (n)	N/A
N-Layer	conductivity	S/m (siemens/meter)
N-Layer	frequency range	Hz

C. Metadata

GDBV will also provide the ability to attach metadata records to entities. Currently, two metadata objects are defined for use in GDBV: SurveyMetadata and PointMetadata. At the time of this writing, the definition of these metadata objects as well as the level at which they are attributed in the database hierarchy are under discussion and are subject to modification. In Table 2, the metadata parameters are listed along with the units of each parameter.

137

Table 2 Metadata Object Definitions

Object Name	Parameter Name	Units
SurveyMetadata	collection ending date	N/A
SurveyMetadata	collection starting date	N/A
SurveyMetadata	distance surveyed quantity	nautical miles
SurveyMetadata	geophysical measuring device description text	N/A
SurveyMetadata	geophysical measuring device model number	N/A
SurveyMetadata	geophysical measuring device manufacturer	N/A
PointMetadata	bearing angle	deg.
PointMetadata	variance	N/A

IV. Concept of Operation

During normal operation, GAIT derived data parameters are written to a local GDBV database that will store in-situ ocean bottom parameters. Either TEDServices or an external transport mechanism can be used to transmit the GDBV database or subsets of this database to a remote TEDServices installation (e.g. domain authorities, regional centers, global data repositories) or standalone GDBV database platform [6]. After a deployment, the GDBV database and the system specific SONAR Data Archive (SDA) are sent to NAVOCEANO for quality assurance and official integration into the relevant OAML database(s).

During a deployment, the GDBV dynamic library will be populated with updates from the GAIT algorithms via the GDBV ingest interface. The official GAIT CONOPS suggests the onboard storage of system specific reverberation in a SDA structure. To derive non-system specific GDBV data, the system parameters must first be deconvolved from the SDA through the use of system specific transfer functions, TF_{system}. This process generates non-system specific reverberation levels that are then processed by the GAIT algorithms to derive system independent GDBV data [5].

Then, the GAIT derived dataset is passed to the GDBV ingest interface as point updates for single layer or n-layer parameters. The data are stored in the GDBV dynamic library of the local GDBV database installation. If an update is processed that overlaps a previous update, the operator may choose to replace, preserve, or add to the existing data records. In the future, a fourth option will be developed for resolving overlapping data records that will utilize the GDBV/GAIT "merge" algorithm.

In the local GDBV system, the in-situ GDBV database can be queried via the GDBV extraction interface. Historical database information will be accessible via the

OAML APIs that are provided for each of the relevant geo-acoustic databases in OAML (MIW, LFBL, HFBL, and GABIM BBS). In the future, NAVOCEANO should consider the sole distribution of an OAML GDBV that evolves the separate geo-acoustic OAML databases into one database that utilizes the GDBV format. A historical OAML GDBV would provide historical and dynamic data accessibility through the same GDBV interface and a vastly simplified method for quickly evaluating historical and dynamic ocean bottom characteristics.

A local GDBV database or portions thereof may be transmitted offboard to a remote GDBV installation via a standard file transmission mechanism (e.g. File Transfer Protocol, Electronic Mail). If TEDServices is available, remote connectivity and data sharing may be satisfied by integrating GDBV into this portal based system [6].

Upon successful receipt of a GDBV dataset, the GDBV ingest interface will be utilized to incorporate the received update into the dynamic GDBV library. Once the update is written to the GDBV dynamic library, the remote location can extract, ingest, and administer the database in the same manner as the local installation from which the update was received. The ability to transmit GDBV datasets to remote locations will provide an improved common operation picture through the sharing of recently observed datasets.

The concept of a regional GDBV center can be extended to establish a global GDBV data repository that receives updates from and transmits updates to regional centers worldwide. In this scenario, a global GDBV database is maintained to provide an "unofficial" GDBV database for the world ocean at a central facility, such as NAVOCEANO. As regional centers receive updates from remote vessels or other centers, the regional center transmits updates to the global repository for ingest into the global dynamic GDBV database. Eventually, the dynamic database and corresponding SDA datasets will be evaluated, integrated into a future release of relevant NAVOCEANO historical databases, and purged from the dynamic global database.

Upon return to port, the SDA data and the GDBV dynamic database will be sent to NAVOCEANO for quality assurance and then used to officially update the static OAML databases. Updates to the OAML databases will be made according to the normal NAVOCEANO procedures and schedule. The historical update process will be made to the separate LFBL, HFBL, MIW, and BBS database formats unless NAVOCEANO evolves these databases into one database product. Since GAIT algorithms and external TDAs will utilize the OAML APIs for accessing historical OAML data, the underlying format of the historical databases is not required to be that of GDBV. However, NAVOCEANO can contribute to GDBV development and evaluate the GDBV format for performance and suitability as a future format for distributing its database products.

IV. OAML Considerations

The GAIT algorithms will be submitted for OAML approval. The dynamic GDBV dataset that is populated in-situ by the GAIT algorithms will not contain OAML approved data. The dynamic GAIT data will eventually be OAML approved if it is incorporated into future OAML database(s) but only after it is subject to adequate quality assurance and migration procedures performed by NAVOCEANO.

As the landing pad for the GAIT algorithms, the dynamic GDBV database is a crucial component for enhancing historical products with TTS datasets. Therefore, the GDBV API and utilities may be submitted to the OAML Software Review Board (SRB) as a component of GAIT. That is, GDBV would be a set of algorithms that GAIT employs to make dynamically inverted datasets persistent in a file system for future transmission to NAVOCEANO. In this dynamic operational role, the GDBV database contains no data for validation and verification according to normal OAML database approval procedures. However, the dynamic GDBV does contain routines that can be OAML approval via the procedures executed for OAML certified algorithms. These OAML approval procedures would be similar to those that will be applied to the GAIT algorithms.

In the future, NAVOCEANO may choose to evolve its historical databases to the GDBV format. NAVOCEANO may execute this conversion one database product at a time with phased OAML updates. The conversion would involve reformatting the current OAML geo-acoustic database, providing a product specific API that utilizes the GDBV API and/or utilities, modifying database production tools to write to the GDBV format, and updating database documentation.

NAVOCEANO should consider evolving its separate geo-acoustical databases to a single database format with a common access API. A single broad database would eliminate the maintenance and validation of multiple independently formatted databases and ease user integration of multiple OAML products.

It is important to note that any effort to maintain a consistent OAML API for accessing OAML databases makes the actual format of the databases a non-issue for users. As long as external applications (in this case the GAIT algorithms) access OAML databases through the official OAML APIs, they are not subject to major modifications when an OAML database format is changed. GDBV is suggested as a future geo-acoustic OAML format because of its expandability and utilization of modern technology, but OAML adoption of any single standard database format (e.g. HDF5, netCDF) for its historical databases would be of significant valuable to OAML end users.

Currently, each OAML database has a separate API that provides access to its data. A common geo-acoustic API that pulls data from all the geo-acoustic databases (or the single geo-acoustic database with separate libraries for each of the original OAML datasets) via one set of functions would improve utilization of the OAML products by providing a common mechanism for accessing OAML data. Indeed, this API might be expanded to include access to all the OAML database products by providing a common middleware component thereby reducing the level-of-effort required for the integration of multiple OAML databases. Efforts to preserve the API signatures (the API routine's inputs and outputs) would provide a robust mechanism that would be adaptable to future OAML database updates, even those that involve format changes. Certain aspects of the GDBV system would provide valuable methodologies for fielding a common OAML access API.

V. Conclusion

GDBV will provide the dynamic landing pad for GAIT and ASCS derived parameters through an enabling set of file-based data transfer utilities and a traditional API. GDBV draws from the advances of the PUMA TTS project whereby dynamic PUMA derived bathymetry is stored as a supplemental database to the OAML DBDB-V product. In its historic role of operation, GDBV will support the storage of four OAML databases: OAML MIW (surface sediments and roughness), OAML HFBL, OAML LFBL (including n-layer LFBL), and APL-UW's GABIM BBS.

Although GDBV has been developed to store ocean bottom information, its generic implementation and extensibility makes it a suitable solution for a wide variety of environmental data types. Future integration with networked systems such as TEDServices would provide substantial data sharing capabilities for GAIT/GDBV users. GDBV is a crucial component to the overall GAIT system in its quest to fulfill the original SWIM requirement to develop algorithms that feed a three-dimensional, variable-resolution worldwide geo-acoustic database that supports shallow water full wave propagation loss models.

REFERENCES

[1] P.A. Burrough and R.A. McDonnell, *Principles of Geographic Information Systems*, New York: Oxford University Press, 1998, pp.48-50.

[2] "Data Base Description for Low Frequency Bottom Loss (LFBL)(U)," *The Naval Oceanographic Office (NAVOCEANO)*, July 2002.

[3] "HDF5 User's Guide," Retrieved October 8, 2002, from the *National Center for Supercomputing Applications (NCSA)*, Hierarchical Data Format Web Site: http://hdf.ncsa.uiuc.edu/HDF5/doc/H5.user.html.

[4] "Introduction to HDF5 Release 1.4," Retrieved October 8, 2002, from the *National Center for Supercomputing Applications (NCSA)*, Hierarchical Data Format Web Site: http://hdf.ncsa.uiuc.edu/HDF5/doc/H5.intro.html.

[5] K.A. Koehler, "Acoustic Research and Development: Execution Plan," *Space and Naval Warfare Systems Center (PMW-155)*, DRAFT (Rev 3), February 28, 2003.

[6] K.A. Koehler, J.G. Kusters, "Gaining Tactical Advantage through Undersea Oceanographic Data Collection and Distribution from Navy Platforms by Exploiting Through the Sensor (TTS) Technology and Net-Centric Warfare (NCW) to Populate the Virtual Natural Environment (VNE) as Part of the Navy Four Dimensional Cube (4DCube)," *Proc. MTS/IEEE Oceans 2002*, October 2002.

[7] "ozone – The Open Source Java ODBMS," Retrieved October 8, 2002, from the ozone Database Project Web Site: http://www.ozone-db.org.

[8] "Shallow Water Issues for Meteorology and Oceanography (SWIM)," *Space and Naval Warfare Systems Center (PMW-155)*, February 22, 1999.

[9] C.A. Steed, "Geophysical Data Base Variable Resolution (GDBV): Database Definition Document," *The Naval Research Laboratory – Stennis Space Center Detachment*, DRAFT (Rev 6), May 14, 2003, in press.

[10] C.A. Steed, K.A. Koehler, and J.E. Braud, "VGRID: A Generic, Dynamic HDF5 Storage Model for Georeferenced, Grid Data," *Proc. MTS/IEEE Oceans 2002*, October 2002.

[11] C.A. Steed, J. Landrum, and C. Moreau, "PUMA-TEDS Technical Execution Plan," *The Naval Research Laboratory, Stennis Space Center Detachment*, NRL/FR/7440-02-10,003, June 2002.