VLB ARRAY MEMO No. 469

The VLBA Database

JONATHAN D. ROMNEY
National Radio Astronomy Observatory
Charlottesville, Virginia

1985 July 23

INTRODUCTION

In this memorandum I present the case for maintaining all the VLBA's operational data in an integrated database management system, beginning with a very abbreviated tutorial introduction to the concept of a database system, and an outline of the salient advantages of such a system for the VLBA in particular. On a less abstract level, I then develop a conceptual structure for a VLBA database, and consider the access required through the several interfaces to the Array's operational subsystems. Still more concretely, two final sections review some of the unique technical aspects of our application which may influence the design of a VLBA database system. In most of the present discussion I leave the implementation and procurement of the database system entirely unspecified; if we can agree on the concept and the interface specifications, a subsequent memorandum will consider the options available for acquiring the capabilities we require.

DATABASE MANAGEMENT SYSTEMS

Two complementary fundamental elements constitute the essence of a database system: an integrated, structured database, and the management software which mediates access to the database. Both the hardware which supports the storage and processing, and the application software which provides input and requests output data, are explicitly external to the database system. The database appears to the "user", typically an application programmer, as a thoroughly abstract entity, containing only those logical elements — and their relationships — of immediate interest. The details of the data organization within the database are determined by the management software, with some guidance from a human database administrator who may suggest appropriate directions for optimization. For the user, however, the primary point of contact to the database management system is through the "data-manipulation" interfaces it provides to his application code, typically in the form of callable subprograms. These interfaces allow the user to specify insertion or deletion of data and retrieval of potentially extremely complex logical combinations of stored data, all in a completely "data-independent" manner — *i.e.*, the application software is not required (or permitted) to encode a knowledge of the detailed organization of the database. Throughout these operations, the management system retains control over the integrity and validity of the database by performing authorization and validation checks and journaling its activities as specified by the database administrator.

The essential advantages achievable through this approach are detailed in any elementary work in the voluminous and rapidly-growing database literature. (Available at NRAO in both the Charlottesville and VLA libraries is C. J. Date's *Introduction to Database Systems.*) In the present discussion, I concentrate on those areas particularly relevant in the VLBA context:

> *Data Independence.* This point, alluded to above, cannot be overemphasized, and contributes heavily to many of the more specific features mentioned below. Data independence is vital to the VLBA, where the development effort as well as the completed configuration is geographically distributed. It enforces the specification of clean interfaces among project subsystems; at the same time, the interfaces need only be specified at a simple conceptual level, so that subsystem development is not impeded by the need to negotiate detailed interface formats. Perhaps most importantly, the programming staff is relieved of the burden of explicit data-retrieval and -replacement coding. I would estimate, conservatively I believe, that we could expect at least a 20% reduction in volume of code and programming costs from this aspect alone.
>
> *Integrity of Operational Data.* The scope of observations with the VLBA — continuous routine operation on a scale achieved in current VLBI practice only with extraordinary effort — raises automated scheduling and record-keeping from a convenience (which we have never found compelling in Mark 2 VLBI work) to a necessity. While facilitating operational control, this concentration of the Array's working data in computer storage media also introduces a vulnerability to loss or distortion of that information through hardware or software failures, human error or even malicious action. A database management system provides facilities to ensure the continued integrity and security of the stored data, including efficient backup and restore operations, a journaling mechanism which allows partially-completed manipulations to be "rolled back" in the event of a failure, validation checks to preclude entry of erroneous inputs, and passwords and other security features.
>
> *Minimization of Redundant Data.* Connections and relationships among the stored data elements are integrated into the database structure, reducing the need to store multiple copies of the same information. This minimizes as well the possibilities for internal inconsistency.
>
> *Facilitation of New Applications.* The threshold of effort for introducing new applications is lowered, because the necessary data are either already present or easily incorporated, and the volume of new code is reduced. And it is not imperative to foresee and provide for all eventual applications in designing the initial database.
>
> *Global Optimization.* Both the storage structure and the access methods can be changed as the Array develops to maintain peak overall operating efficiency and to accommodate new applications — without requiring any modification of existing application software.

Beyond these essential points, several other major advantages are worth mentioning. While the former must be realized by any database system, including an in-house development, in practice I doubt that any of the following "conveniences" would ever actually be implemented except in a commercial system. These powerful features must not be dismissed as mere frills. They offer both significant economies in programming costs for the overall development of the Array, and important contributions to flexibility of operations management:

> *Reports and Graphics.* Many commercial database systems embody a rich assortment of tools which facilitate the generation of numeric reports and in some cases even graphic displays. Some areas where this capability is particularly promising include: management (summaries of the Array's scientific activities according to various criteria, assistance in scheduling); operations (work force planning, tape logistics, dynamic scheduling via meteorological data); and engineering (near-real-time remote diagnostic information, maintenance and spare-parts control).
>
> *Interactive Query Language.* An interactive facility for retrieving and updating information in the database brings users in all areas of Array support closer to the basic operational data. Special-purpose outputs and modifications are quickly accomplished, frequently without the need for diverting the attention of the programming staff. The query language is often oriented toward use by non-specialists, and may provide a menu-driven mode for novice users. Frequently-used procedures can usually be stored as a relatively clear and easily modified sequence of commands in the query language. In some systems the query language supports interactive terminal-screen formats for data entry.

Concluding this brief descriptive overview of database management systems, I suggest that we can gain much in efficiency, simplicity, and reliability — both in development and operation of the VLBA — by integrating all the Array's operational data into a central database management system. Despite some initial investment in establishing the system, this general approach offers an eventual global economy of both cost and effort for the entire VLBA project.

I would argue further that this is an area where we should forsake our historic propensity to re-invent the wheel, and seriously consider buying one of the many commercially available systems. Both by definition and as a fundamental objective, a database system represents a separable body of code, with cleanly defined interfaces to the rest of the operation in which it functions; thus it is an ideal candidate for purchasing software whose development, debugging, and maintenance costs can be shared with a large number of other users. Certainly many organizations and projects similar to ours have made this choice, and indeed there is a large and competitive market in such systems.

This recommendation is in fact more compelling in view of the recent and/or anticipated budget stretchouts. Especially with a restricted programming staff we will be reluctant to commit the resources needed for an in-house development. Yet the procurement of the system itself could be deferred somewhat if necessary without critically impacting development of the individual subsystems once the conceptual interfaces are determined.

VLBA Database Structure

How would a VLBA database be organized? I address this question from two different perspectives: this section can be regarded as a "top view", outlining the major conceptual entities which constitute the database, and the primary relationships among them; in the following section, a series of "side views" describe the aspect these entities present to the user at the interfaces with the Array's operational subsystems. In both cases, it should be emphasized, a relatively abstract conceptual view is maintained, removed from any specific file or record structures. The discussion is also, inevitably, skewed toward those areas to which I (and others who have commented on the subject) have given the most thought.

Both this and the subsequent section must be supplemented eventually by detailed quantitative specifications of the storage volume required for the various structural entities and the data rates to be supported through the interfaces. It does not appear feasible to devise such specifications in the current state of planning for most VLBA subsystems. A rough estimate for the potentially dominant volume of monitor data, however, appears in a later specialized discussion.

In considering the structure of a VLBA database, it may be valuable to consult some earlier specialized suggestions along these lines in the VLBA "literature". Clark introduced the concept of a database for station monitor measurements, and discussed some of the difficulties related to the volume of these values, in VLBA Memos 278 and 396. And in VLBA Correlator Memo VC041, the correlator group suggested a more central database, and sketched the conceptual elements relevant to operation of the correlator.

Readers unaware of the distinction should note that the word "archive" is used consistently to refer to the permanent copy of the Array's astronomical measurements (plus some supporting information). The VLBA *archive* represents a much larger volume, of more simply organized data, than the proposed VLBA *database*, and the two structures require only a loose coupling. I do not propose to implement the archive using a general-purpose database system.

We can group the major database entities conveniently in several categories:

> **Fundamental Parameter Catalogs.** These catalogs tabulate the fundamental geodetic, astrometric, and geodynamic parameters relevant to all Array observations. Each contains the current best values available, with good estimates of the precision of each entry, and should be maintained and updated as necessary by a single responsible staff member, although provision will also be necessary for provisional initial entries.
>
> > **Station Catalog.** Earth-centered coordinates for each station involved in VLBA observations; also such static secondary information as mounting type, axis offsets, horizon profile, and perhaps axis limit stops.
> >
> > **Source Catalog.** Reference positions of all known compact sources. I suggest that flux, polarization, structure, *etc.* all be omitted from this list.
> >
> > **Geodynamic Catalog.** All "constants" — *i.e.*, slowly-varying parameters — of geodynamic origin, including such effects as precession, nutation, solid earth tides, perhaps even known tectonic plate motions.

Time Catalog. The rapidly-varying geodynamic parameters: UT1 and polar motion. Unlike the other catalogs, this will require frequent new entries as soon as available (preferably from a single time-service agency); lapsed data can be purged.

Array Logs and Schedules. These data constitute together a growing permanent record of past and planned Array activity. Entries are basically at observing-program level (and will likely be keyed to the program name), with some indication of the subarraying involved in transitions between programs. There will probably be specialized programs named "maintenance" and "calibration" with perhaps a sequential qualifier.

Program Log. Assorted basic information on each observing program, including program name, current status, name(s) of observer(s), dates of proposal, observation, correlation, and distribution, *etc.* Each entry points to an associated **Program History** either currently in the database or in the archive; in this sense the **Program Log** serves as an index to the VLBA archive.

Observation Log. A chronologically-oriented record of programs observed by the Array.

Observation Schedule. Currently planned sequence of future programs to be observed, structured similarly to the **Observation Log.**

Alternate Schedules. Pre-planned substitutes for the **Observation Schedule**, to be used in the event of, *e.g.*, unusual weather or interference conditions.

Correlation Log. A record of programs correlated, structured similarly to the **Observation Log.**

Correlation Schedule. Planned correlation sequence, derived from the **Observation Log** but modified for efficiency of correlation.

Program Histories. A detailed history is maintained for each currently active program, from the time a detailed observing plan is filed until correlation is complete; the completed history is then unloaded to and retained in the VLBA archive with the associated data and purged from the database's on-line storage. In the event that re-correlation is required, this history can then be reloaded and extended. *Each* history includes the following substructures.

Observation Plan. Detailed chronological plan of observations, including sources to be observed, frequency, polarization, channelization, and recording configuration. This may serve as a skeleton for the **Observation History.**

Fundamental Parameters. Entries from the **Fundamental Parameter Catalogs** used for observation and correlation. Two sets of entries are necessary: entries current when an observation begins are used for the entire observation; and similarly those current when correlation begins govern the entire correlation process.

Observation History. Detailed chronological record of observations as actually performed, including most entries from the **Observation Plan** (showing any deviations from plan), plus, *e.g.*, video-tape serial numbers.

Correlation History. Similarly detailed record of correlation, including playback and correlator configurations, and disposition of output data.

Station Histories. A history of station-based data is maintained for each VLBA station, and as far as possible for participating "foreign" stations as well. The histories probably should be retained on-line for several years before being unloaded and purged from the database, and available indefinitely from off-line storage. *Each* history includes the following substructures.

Diagnostic Data. A record of monitor points indicative of all aspects of station performance. These data dominate the station history, and may dominate the entire database as well unless some compression is achieved. This important point is discussed separately below.

Clock History. A compilation of clock and LO offsets derived both from local measurements and from interferometer calibration observations.

Calibration History. Complex gain measurements, either locally derived or from interferometer calibration observations.

Weather History. A record of local meteorological measurements.

Tape Inventory. This entity tabulates a variety of information pertaining to the Array's entire inventory of video tapes. Entries, presumably keyed to the serial numbers, indicate each tape's current location and status, history of manufacture and use, and record of data quality.

Parts Inventory. Similarly, this inventory maintains a current itemization of the location, status, and repair history of each module.

VLBA Database Interfaces

This section emphasizes the manipulations of database entities, which represent the interfaces between the database system and the many Array subsystems involved in management, operations, and technical support. As a corollary to the data independence and flexibility of applications achievable with a database management system, it is neither necessary nor desirable — from the perspective of the database — to restrict the number or complexity of these interfaces as we have done with the interfaces *among* the Array subsystems. Rather the database interfaces follow naturally the substructuring in the Array's operational tasks:

Array Operations. Under "operations" I include both program-level and detailed control of observations using the Array, and also the handling of monitor data returned from the stations.

Normal Program Control. Basically follows the **Observation Schedule.** Activating a new program requires manipulating the **Array Logs and Schedules** and the appropriate **Program History** to show the status as "currently being observed" in the **Program Log**, update the **Observation Log**, and obtain the necessary values from the **Fundamental Parameter Catalogs**. Similarly, termination (or suspension) of a program changes the status to "observations (partially) complete".

Alternate Program Control. May be invoked at any time to supersede the **Observation Schedule** and substitute a program from the **Alternate Schedule**, in order to escape low-frequency interference problems or take advantage of good weather conditions.

Observation Control. Follows the detailed **Observation Plan** in the currently-active **Program History**, making reference to the stored observation-time **Fundamental Parameters** as necessary. Some entries will be made in the **Observation History** where confirmation via monitor data is inappropriate.

Status Monitoring. Enters station monitor data according to type, primarily into the appropriate **Station Histories**. Measurements intended for diagnostic and/or engineering purposes are inserted in the **Diagnostic Data** substructure; as discussed in the next major section of this memorandum, these may well be buffered, averaged, or otherwise processed before entry. Other monitor information, however, must be inserted immediately — notably, entries in the **Clock**, **Calibration**, and **Weather Histories**, and video-tape serial numbers and status indications required for the **Observation History**. The **Tape Inventory** must also be updated immediately when recording begins on a new tape.

Correlator Operations. While generally paralleling *Array Operations*, these tasks do not bear the burden of the massive monitor data flux, but on the other hand must support the collection of data for the VLBA archive.

Correlator Program Control. Basically follows the **Correlation Schedule**. Activating a new program requires manipulating the **Array Logs and Schedules** and the appropriate **Program History** to show the status as "currently being correlated" in the **Program Log**, update the **Correlation Log**, and obtain the necessary values, again, from the **Fundamental Parameter Catalogs**. Similarly, termination (or suspension) of a program changes the status to "correlation (partially) complete". A completed **Program History** is transferred to the archive and purged from the database, and the **Program Log** updated to point to the archive entry.

Correlation Control. Follows the detailed **Observation History** in the currently-active **Program History**, making reference to the stored correlation-time **Fundamental Parameters** as necessary. References to the **Clock** and **Weather Histories** are crucial secondary inputs. The **Correlation History** records details of the correlator's activity. In addition, entries in the **Tape Inventory** are necessary, and "gain" and "flag" tables must be constructed from the information in the **Station Histories** for inclusion in the VLBA archive.

Calibration Reduction. Asynchronous, but prompt, post-correlation fringe-fitting on observations of specified calibration sources will be used to track variations in the independent atomic clocks at all stations, and may also provide additional gain-calibration information. The feedback of these results to the correlator is accomplished by entries in the **Clock** and **Calibration Histories.**

Tape Logistics. Besides the automated **Tape Inventory** entries mentioned above as part of *Array* and *Correlator Operations*, "manual" updates will be required in conjunction with all shipping and receiving activity at both the Array Operations Center and all the Array sites. The bar-code system introduced by Haystack Observatory for Mark III tapes can probably be used to great advantage for this purpose, as well as to simplify the entries and checks when mounting a new tape for recording or playback. A vital function exercised through this interface is the provision for periodic checks to verify timely arrival of recorded tapes at the Operations Center prior to processing, and a sufficient supply of tapes available for recording at the stations.

Engineering. Since the primary technical expertise will be centered at the Array Operations Center, we must rely on remote diagnosis of problems and a modular spares strategy.

Monitor Reports. Must be available over a wide range of timescales. Both tabular and graphic displays of the **Diagnostic Data** in one or more **Station Histories** will be essential. (The discussion in the next major section is again very relevant here.)

Repair Control. Similar to *Tape Logistics*, requires entries in the **Parts Inventory** whenever modules are shipped to or from the remote sites, and as repairs occur.

Management. I have grouped primarily non-routine activities in this final, admittedly catch-all category.

Schedule Entry. Inserts new **Program Log** entries to extend the **Observation Schedule** into the future; an associated **Program History** is also created at this point.

Fundamental Parameter Maintenance. Included as a management function because of the necessary careful control and accountability. This is the only path by which the **Fundamental Parameter Catalogs** may be modified.

Performance Review. Must be possible at all levels of all areas of Array operation. Tabular and graphic reports will be needed from almost any part of the VLBA database. A brief enumeration to demonstrate the diversity of possible requirements would include: weather history at a particular station; overall reliability statistics for cryogenic equipment; recording quality as a function of age for a particular production run of video tape; correlator down time, sorted by cause; and an up-to-date index to the VLBA archive.

Strategies for Monitor Data

In VLBA Memos 278 and 396, Clark has called attention to some of the problems inherent in the volume and characteristics of monitor data which will be returned to the Array Operations Center from the remote antenna sites. An informal estimate of the anticipated volume is 60 numeric values per second for all ten Array elements. Most of these data will never be needed as individual values; for a long-term record, averages over a period of some hours will probably suffice. But for occasional remote diagnosis, access to particular subsets (by station and monitor point) of recent densely-sampled data, extending back from near real-time to as much as several days, as well as to the long-term record, will be essential. Access to both the short- and long-term monitor values must be achieved using a single set of software tools.

In view of these requirements, the brute-force approach of simply entering all monitor values into the database as received (and subsequently purging all but the long-term averages) seems impractical. One of the prices paid for all the advantages enumerated earlier in this memorandum is an inefficiency in insertion and deletion of data relative to that achievable in a straightforward sequential organization. Thus a database system specified to support this mode would have to devote all but a small fraction of its resources simply to the insertion and deletion of the data otherwise referenced only in forming averages. On the other hand, we do want to root the diagnostic tools within the database system to take advantage of the report-generating and graphic-formatting facilities available.

I propose as a resolution of this dilemma the following sorting and buffering scheme. All status data from the remote sites are processed as received by a sorting interface which routes the critical status information directly to the database system for immediate entry, but in normal operation transmits purely diagnostic values to a series of simple sequential buffer files which are discarded when they reach several days in age. Note that this sorting function is necessary anyway; only the outputs differ here from what we might have if all data were entered directly. Other software then periodically passes through the buffered data and forms the appropriate averages for insertion into the database.

When diagnostic readouts are requested, a filler routine becomes active, transferring only the required subset from the buffer files to special-purpose structures in the database. And the sorting routine is instructed to route that same subset of received monitor data directly to the database too, facilitating near-real-time analysis. The specialized database entities can easily be purged when no longer needed. This scheme reduces the burden of filling monitor data to the minimum necessary, and imposes negligible overhead in accessing specific monitor data as received.

Clark's suggestion (VLBA Memo 396) that we might only log monitor values when the change from the previous value exceeds a threshold offers a potential compression of incoming data to the point that we could actually insert all these entries directly into the database. (I must admit to some skepticism that we can get the thresholds and integration times right for all the various measurements without a major headache. Nevertheless, this idea is sufficiently promising that I think we should consider it further.) I would combine this approach with the scheme suggested in the preceding paragraphs by modifying the "sorting interface" to include both the threshold testing and averaging operations in real time, so that the compression can be accomplished while retaining proper boxcar averages for the long-term record.

Monitor Data from Foreign Stations

Another problem in handling monitor data, of a very different sort, arises when "foreign" — *i.e.*, non-VLBA — stations are included in Array observations. For European and other distant stations at least, it is likely that real-time communications will be unreasonably expensive, and the presence at most foreign stations of full-time operating and on-site maintenance staffs removes the necessity for a remote diagnostic capability. For both these reasons, it seems probable that the critical status information from these stations — that necessary as input to the correlation process — will be transmitted to the Array Operations Center on standard magnetic tapes or floppy disks along with the video tapes.

Hence a delayed "array operations" interface will be required, through which this recorded status information can be entered, and properly sequenced, as if it had been transmitted in real time.