

The Relational Database and Calibration Software for the Caltech Millimeter Array

N. Z. SCOVILLE, J. E. CARLSTROM, C. J. CHANDLER, J. A. PHILLIPS, S. L. SCOTT,
 R. P. J. TILANUS, AND Z. WANG

Owens Valley Radio Observatory, California Institute of Technology, Pasadena, California 91125
 Electronic mail: nzs@astro.caltech.edu, jc@astro.caltech.edu, cjc@astro.caltech.edu, jap@astro.caltech.edu,
 sls@ovro.caltech.edu, rpt@jach.hawaii.edu, zw@ipac.caltech.edu

Received 1993 July 9; accepted 1993 September 15

ABSTRACT. A relational database management system has been implemented on the Caltech millimeter-wave array for both real-time astronomical and engineering data and post-processing calibration and analysis. This system provides high storage efficiency for the data and on-line access to data from multiple observing seasons. The ability to access easily the full database enables more accurate calibration of the raw data and greatly facilitates the calibration process. In this article we describe both the structure of the millimeter-array database and the implementation of a data analysis program, both of which make extensive use of Sybase, a commercial database management system with application development software. This use of relational database technology in real-time astronomical data storage and calibration may serve as a prototype for similar systems at other observatories.

1. INTRODUCTION

The major steps in the processing of astronomical data typically include: the inspection, archiving, and calibration of the raw data, and some form of image processing or measurement. In the last two decades considerable effort has been expended towards the last area—that is image-processing software, measurement algorithms, and automatic search routines. On the other hand, relatively little effort has been expended on the archiving of astronomical data with emphasis towards making that data permanently on-line and immediately accessible. Although exceptions do exist for the fully calibrated final data products of recent NASA missions, the access to raw datasets is extremely limited and usually quite cumbersome.

We have recently implemented a commercial database management system (DBMS) on the Caltech millimeter-wave array. The system provides a number of major advantages in efficiency, user accessibility, and calibration accuracy which might stimulate the development of similar systems at other astronomical observatories. In this paper we start with a brief description of the physical characteristics of the millimeter-wave array, its users' needs, and our implementation of the database management system for data analysis (Sec. 2). In Sec. 3 the specific DBMS and the structure of the millimeter-array database are described. In Sec. 4 we describe the data acquisition, and in Sec. 5 the analysis and calibration program which accesses the database is briefly summarized. In Sec. 6 we review our experience with respect to manpower, costs, and long-term database management needs in order to provide realistic expectations for other observatories considering similar projects.

2. OVERVIEW OF THE MILLIMETER-WAVE ARRAY

2.1 The Instrument

The Owens Valley Radio Observatory millimeter-wave array is operated by the California Institute of Technology with support from the National Science Foundation. The array is located 250 miles north of Los Angeles near Bishop, California. It presently consists of five 10.4-m, millimeter-wave telescopes which can be positioned at stations separated by up to 220 m (cf. Scoville et al. 1993). The array is presently undergoing an expansion to six (and possibly seven) telescopes.

The array is used for astronomical aperture synthesis interferometry of molecular line and continuum emission in the $\lambda = 1.3$ and 2.6 mm bands. The signals received by the separate radio telescopes are cross correlated to produce interference fringes which sample spatial frequencies of the sky brightness distribution. The data are then Fourier transformed to yield an image with angular resolution equivalent to that obtained by a diffraction-limited aperture similar in size to the largest spacings between telescopes.

In order to obtain sufficient sampling of the brightness distribution in astronomical sources, two techniques are used to vary the interference fringe spacing on the sky: (1) semicontinuous tracking of sources as they transit from east to west over a fixed pattern of telescope positions (i.e., configuration) results in a continuous variation of the projected baselines, and (2) observations in multiple configurations with different relative telescope spacings further improve the sampling of the spatial frequencies. Since there is considerable overhead in moving the telescopes to a new configuration, both in transporting them and in determining their physical separations, array configurations are usu-

ally not changed more often than every few weeks. To achieve adequate sampling of the spatial frequencies with an array of five telescopes typically entails observations over a period of several months.

Calibration for the raw data requires access to information on instrument stability, telescope positions, atmospheric conditions, and source variability for the extended time period over which the observations were obtained. In the past, most data management for interferometer arrays has been sequentially organized in time rather than by astronomical source or observing project, and accessible after the fact only through cumbersome printed logs and searches of archive tapes. Furthermore, we note that virtually all aperture synthesis arrays have relied upon site-specific fixed format files for real-time data storage, providing little flexibility for change. All of these factors have been a serious impediment and frustration to observers using aperture synthesis arrays.

2.2 Typical Datasets

A typical dataset for aperture synthesis with the five-element millimeter-array involves earth-rotation tracks in 2–3 separate configurations, each of which has ten baselines, or telescope pairs. During each track interferometric data are obtained on: the program source, a suitable point-source phase calibrator, a spectral passband calibrator, and flux calibration sources. For each of these measurements 5-min integrations are obtained simultaneously on all baselines. The passband calibrator (usually a strong continuum source such as 3C 273 or 3C 84) is observed for one to three integrations during the track. For absolute flux calibration two alternatives are possible: (1) measurement of planets (primary standards) or (2) measurement of quasars (secondary standards). Since the fluxes of the quasars are slowly varying on time scales of weeks to months, their fluxes may be assumed constant during a track. However, the quasar flux for each epoch must be determined from nearby tracks when both the quasar and a planetary standard were measured. Most of the track is spent in repetition of half-hour cycles which typically include five integrations on the astronomical source being imaged followed by two integrations on the phase calibrator.

Within each 5-min integration, data are obtained for a broadband continuum channel and 256 spectroscopic channels on each baseline in both frequency sidebands of the double sideband receiver systems. The total quantities of data involved are approximately 4 Mbyte per track and 1.5–2 Gbyte per year. Although data acquisition occurs at the Owens Valley Observatory, most calibration and analysis is done either on campus in Pasadena or at guest investigators' home institutions. Two copies of the millimeter-array data are therefore maintained, one at the observatory and the other in Pasadena.

2.3 Data Storage and Analysis

Real-time data taking at the OVRO array is managed by a VAXstation 3100 and a number of μ Vax satellite computers (see Sec. 3.1). At the end of each 5-min integration,

data are transferred from the VAX to the database residing on a SPARC 10/41 running under the UNIX operating system. Data are then periodically transferred to the copy database in Pasadena.

To display, edit, and calibrate the raw data we have developed a program that can be run either at Owens Valley or in Pasadena, called "mma." It uses structured query language (SQL) to access the database with Fortran and C routines to calibrate and plot the data. The user interface for mma is based on menus and "forms." Each form is a page containing "fields" or parameters which are initialized to reasonable values by the program and then can be modified by the user. An example form in mma is displayed in Fig. 1. Along the top of the "OVRO MMA" window are listed the menu options, each of which activates a different SQL, Fortran, or C routine. The form shown in Fig. 1 plots average spectra, either as a standard spectrum (shown with the solid line being the amplitudes and the crosses, the phases), as dynamic spectrum (multiple spectra displayed in time sequence), or the Fourier transform of the average spectrum (lags).

A complete list of the routines and functions available in mma is given in Table 1. At present mma includes software for gain calibration, passband corrections, and flux calibration in addition to numerous utilities. The calibrated data can be output to a FITS format file compatible with one of the standard interferometric imaging packages, such as AIPS, GIPSY, or MIRIAD.

3. DATABASE MANAGEMENT SYSTEM

In establishing a database system for the millimeter-wave array, the major goals were:

- (1) Immediate access to all data covering a minimum of five years.
- (2) Minimum redundancy of information. For example, parameters which are common to all baselines during a single integration should be stored only once.
- (3) Access to engineering and systems monitoring information (for example, telescope focussing and weather data) to help evaluate the quality of the astronomical data.
- (4) Open access to the complete astronomical dataset for calibration purposes. One user may therefore make use of other users' flux calibration observations.
- (5) Flexibility in the selection of data to enable a variety of search criteria (e.g., source name, frequency, angular resolution, observing conditions, etc.).
- (6) Flexibility to modify and expand the database and to correct errors easily. This is especially important in a system such as ours where the data acquisition and storage programs are constantly under development and where the hardware is frequently upgraded.
- (7) The ability to access the data through user-friendly software.

At present there exist two major categories of DBMS: relational and object oriented. Because of the greater ma-

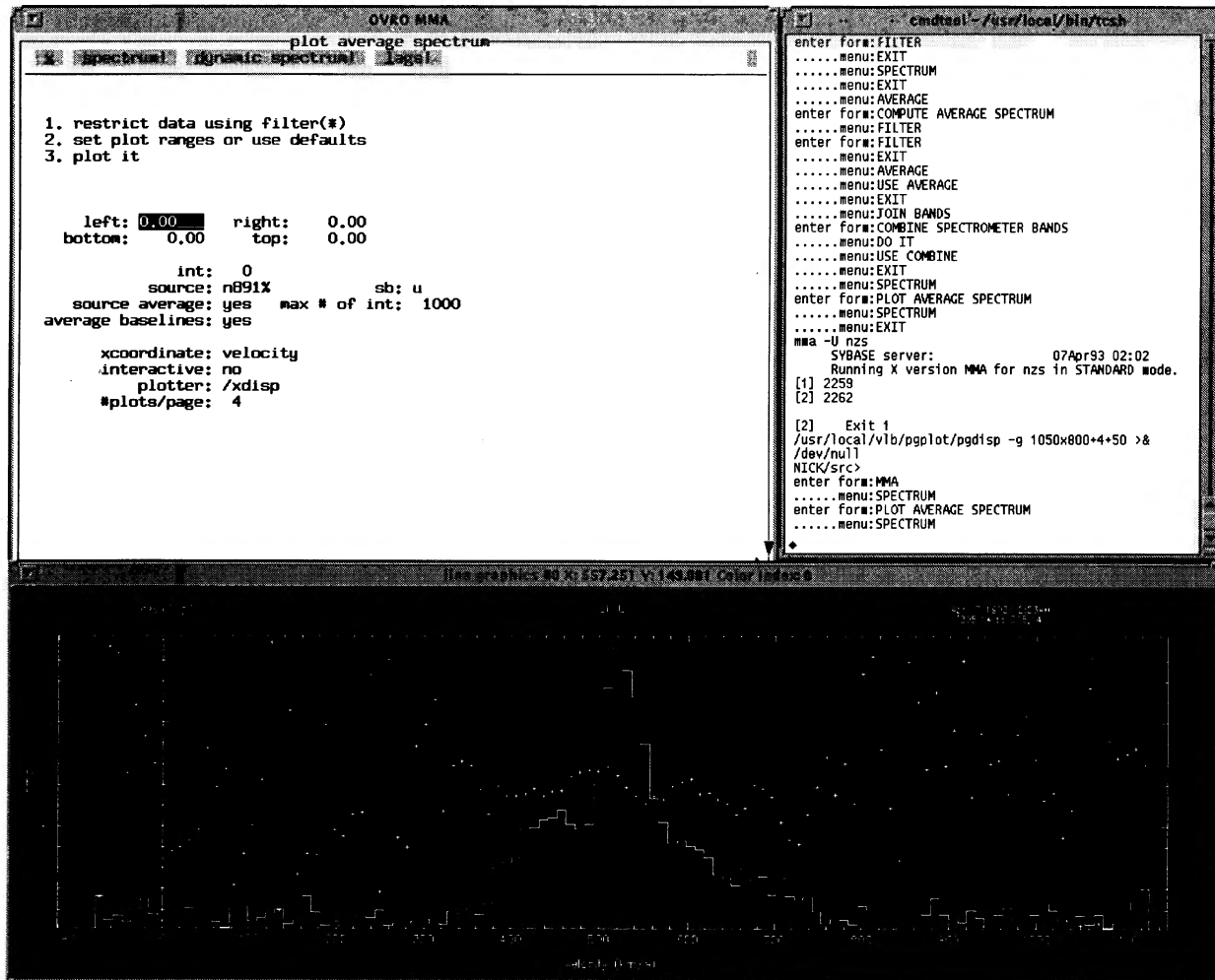


FIG. 1—Sample screen image of windows used by the mma program to retrieve and calibrate data from the millimeter-array database. In the upper left, the control form for spectrum plotting is shown; at the bottom, the PGLOT X-window shows the resulting CO spectrum on the spiral galaxy NGC 891; and at the right, the menu selections are echoed to the operating system.

turity of current relational DBMSs we considered only the former for our application. A decade from now the decision might go in the opposite direction in view of the anticipated greater speed of object-oriented databases (Hurson et al. 1993).

In relational databases the data are stored in tables with each variable represented as a column. Successive rows refer to sequential time records, or to different system components (e.g., the different telescopes or baselines). In most cases, the links between corresponding rows in the separate tables are established by “key” variables such as the universal time or integration number. In a well-designed relational database there are few, if any, repetitions of the same physical variable in different tables. This feature ensures minimum data storage requirements and removes the possibility of inconsistent or contradictory values for variables.

Rather than develop a specialized database management system specifically for the millimeter array, it was clear

from the start that a commercial system should be used. For a modest investment the commercial systems provide enormous flexibility and power, a high level of data integrity and protection against hardware failure, ongoing software upgrades, portability to a broad variety of hardware, and adaptability to changing hardware capabilities. We selected Sybase because of its support of “client-server” architecture with mixed VMS and UNIX operating systems, the broad range of available data types, and its tools for developing a user interface for the database.

3.1 Computing Environment

For the Caltech millimeter array, overall control of the observing is managed by an astronomical control program (ACP) running on a VAXstation 3100 which in turn communicates with separate satellite computers, each dedicated to controlling one of the telescopes or the continuum and spectral line cross-correlator systems (cf., Scott and

TABLE 1
Routines in the Millimeter-Array Analysis Program "mma"

Table management	
Create table/view	Create a "temporary" database table containing a copy of a user-selected range of data, or a "view" (see Sec. 3.2). The raw data are read only for most users, while data in temporary tables can be modified by the calibration routines.
Comment data	Add a comment to data row; can be used to flag data.
Delete data	Delete user-selected data from a temporary table.
Utilities	
Baseline fit	Compute the baseline vectors for the array from phase measurements of compact sources.
Change qualifiers	Allow the user to modify the qualifier associated with a source. Example qualifiers are "1" (primary flux calibrator), "2" (secondary flux calibrator) "g" (gain calibrator), "p" (passband calibrator).
Data entry	Allow the user to enter baseline solutions, antenna locations, project and investigator information, molecular line frequencies, and observing information into the database.
DB management	Find database sizes, do checks, and load databases.
DB transfer	Update the database in Pasadena by transferring recently acquired data from the Owens Valley database. New flux measurements entered by users in Pasadena are transferred to Owens Valley.
Defaults	Reset the data selection criteria and all the forms to their initial, default settings.
Errors	Call routines that correct data for known problems.
Logs	Produce a summary of projects, sources, frequencies and integrations observed during a single track, or during an entire configuration.
Tau0	Plot system temperature vs. elevation and solves for the atmospheric opacity.
Variables	Produce a listing of database table names, with a description of the variables they contain.
Weather etc.	Construct tables with weather and systems monitoring data.
Calibration	
Baseline correction	Apply new baseline solutions to the phases of a user-selected data.
Gains	Fit a polynomial to baseline- or antenna-based amplitudes and phases. The calibration polynomial may be fitted as a function of time or elevation. The polynomial coefficients are stored in the database.
Passband	Average the spectra of passband calibrators and gives the user the option of storing the resulting spectrum point-by-point, or fitting a polynomial to the spectrum and storing the coefficients.
Apply gains and passband	Apply the calibration corrections derived by "gains" and "passband" to data.
Flux calibration	Use the observed fluxes of primary or secondary calibrators to establish the scaling from instrumental counts to Janskys for the data.
Plotting	
Spectrum	Plot spectrum, a dynamic spectrum, or the Fourier transform of a spectrum.
Join bands	Concatenate adjacent correlator modules into a single contiguous spectrum.
Average	Compute and store the average source spectra.
Convolve	Hanning smooth spectra.
Continuum	Plot continuum amplitude and phases as a function of time.
Coherence	Plot baseline-based amplitudes, phases, and coherence as a function of time.
Variables	Plot any database variable as a function of any other database variable. For example, system temperature as a function of source elevation.
Flux	
History	Retrieve and display fluxes that have been entered into the database by users for any source.
Table management	
Measurement	Compute the scalar- and vector-averaged flux of any source within a track and store the result in the database.
Report	Generate a listing of all sources observed during a track, and their fluxes.

Table 1
(Continued)

Output	
Backup	Copy the entire database to tape.
Fits	Write calibrated data to a FITS format output file.
Table-to-File	Copy data from a temporary table to a UNIX file. The program can also read such files from disk back into the database.
Common tools (on most forms)	
Clear	Reset form entries to their default values.
Filter	Activate a form where data selection criteria may be entered. These criteria are used to select a subset of data contained in the current table or view.
Keys	Display a keyboard map of "fast" mma commands.
Global	Display a form with the current values of variables in the GLOBAL table (see Sec. 5).
Help	Activate an extensive tutorial and cookbook describing the use of mma.
Jobs	List currently active Sybase processes. Users can kill their processes.
List	Summarize the contents of the current table and lists any calibration procedures which have been applied to the data.
Print	Make a hardcopy of any form.
Reduction log	List all program activity for a specified mma session.
Tables	Activate a form used to manage user-created tables. The user may list the available tables, delete selected tables, or combine multiple tables into a single table.
Views	Activate a form used to manage views, similar to "Tables" above.

Finch 1992). The satellite computers are μ VAXs running the VAX ELN operating system. The Sybase database management system runs on a SPARC 10/41 (UNIX) workstation connected by ethernet to the ACP workstation and the satellite computers. In Pasadena a second SPARC 10/41 workstation is equipped to handle a duplicate copy of the millimeter-array database. Additional SPARC workstations communicate with both database servers using Sybase client software.

The software purchased from Sybase is summarized in Table 2. It includes a database server license for two SPARC 10/41 workstations, the VAX-based software used to write raw data on the server machine at the observatory, a database inspection tool called Data Workbench, the application development tool which provides a menu-driven, forms-fill-in interface (APT), and C and Fortran database libraries. The Sybase client-server products are used to bridge the gap between VMS and UNIX and handle details of byte swapping and different floating-point formats. The total capital investment in the commercial software is comparable in cost with 1/2 year of programming support.

3.2 The Millimeter-Array Database

Within the millimeter-array database, there are currently 55 permanent tables. Approximately 2/3 of these tables contain astronomical, engineering, and monitoring data from the millimeter array; the remainder contain information used internally by the analysis and calibration program (see Sec. 5).

3.2.1 Raw Data Tables

The observational data may be subdivided into: observing sessions (for example, Winter 1992), array configurations, tracks (a 6–12 hr period during which earth-rotation synthesis data are accumulated for a single observing project), 5-min integrations, baselines within each integration, and the individual spectral bands for each baseline. In Fig. 2 the tables required for the raw astronomical data are shown. For each of the "entities" described above, there exists a corresponding table (specifically called SES, CON, TRA, INH, BLH, and SPH). As one moves from top to bottom on the right side of Fig. 2, the number of rows in

TABLE 2
Sybase Products and Application and Development Tools

Product	Sun/UNIX	VAX/VMS	Use
SQL server	×		Database server
Data workbench	×		Database inspection and maintenance
APT workbench	×		Tool for building user interface for the database
Open client DB-Library/Fortran	×		mma data reduction program
Open client DB-Library/C	×	×	mma and real-time data collection programs

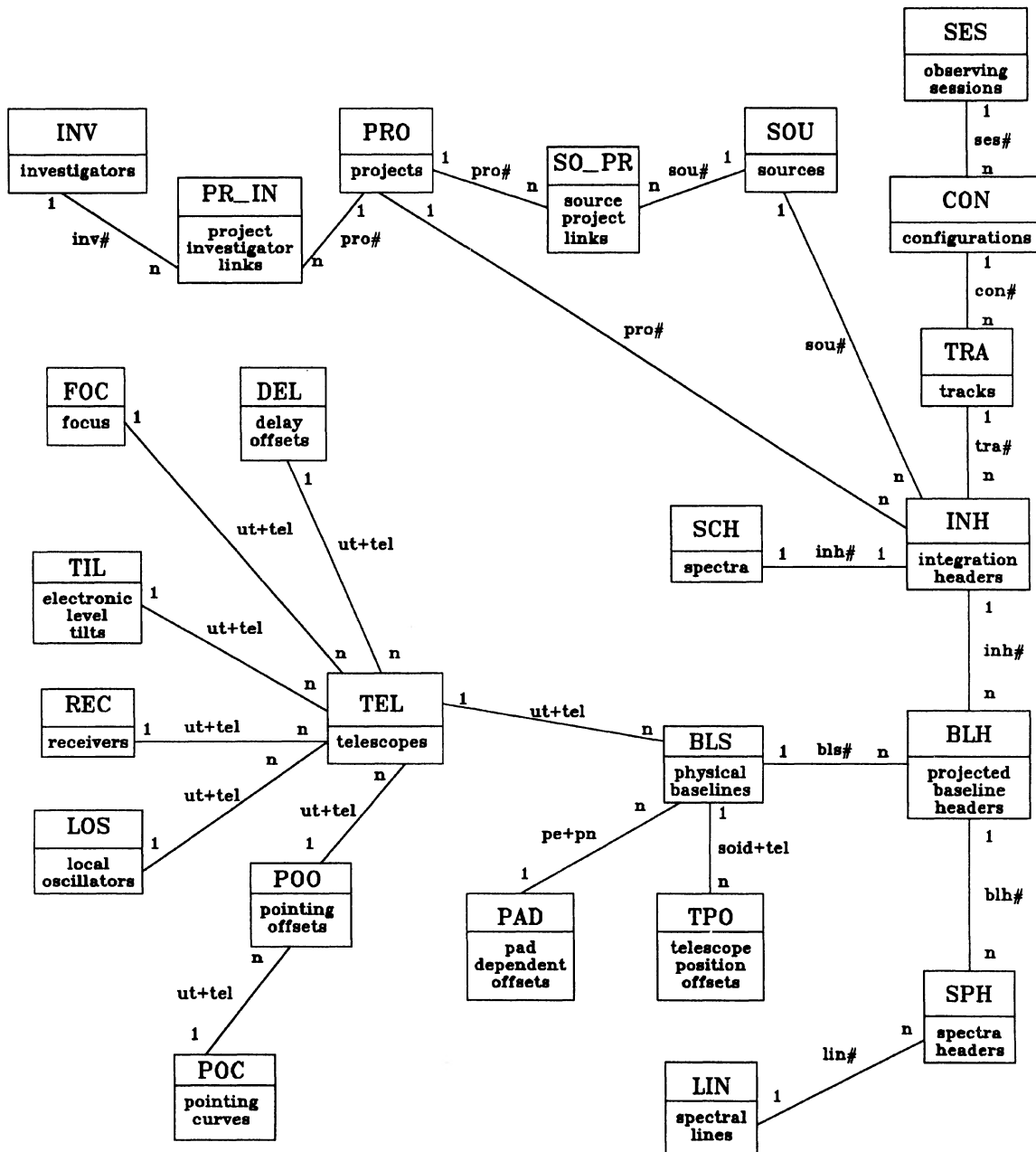


FIG. 2—A schematic diagram of tables containing raw astronomical and engineering data. In the upper portion of each box in bold letters is the table name; in the lower portion a description of the table of contents. The astronomical observations are stored in the tables on the lower right (INH, SCH, BLH, and SPH). The engineering data are contained mostly in the tables branching off the TEL table. Lines between tables indicate where links can be accomplished with the key variable noted along the line. For most tables which can be so linked, there is a 1 to n relationship indicating that a single row in the first table may be linked to many rows in the latter table.

successive tables increases. Thus, for example, corresponding to each row in the track table (TRA), there might be 150 rows in the integration header table (INH), one for each integration, and for each integration there would be 20 baseline header rows (BLH) corresponding to each telescope pair for five telescopes with two sidebands.

The complete set of data (header variables and channel values) corresponding to a single spectrometer for a single

baseline relies upon forming the proper links between the appropriate rows in the tables. The key variables which provide links between adjacent tables are also shown in Fig. 2 along the line between table pairs. This linking process, which is called “joining” in database terminology, is done rapidly by the DBMS provided the key columns have been indexed. In the millimeter-array database this joining of the tables to form a complete observation is accom-

plished through a logical construct called a “view.” The view may be thought of as a virtual table with many columns from which one may select data just as though the data were actually contiguous on disk.

3.2.2 Data Types

The Sybase data types used in the millimeter-array database include fixed and variable length character variables (up to 255 characters), 2 and 4 byte integers, 4 and 8 byte floating point numbers, 8 byte “datetime” variables (year, month, day, and time specification to an accuracy of 1 ms), and “image” variables consisting of packed byte strings (up to 2 Gbyte). For each of the former data types default values may be defined and one specifies whether or not “null” values are permitted.

3.2.3 Indexing

In addition to the indexing established on key variables which link adjacent tables, indexes exist on a significant number of variables or combinations of variables which might be used as search criteria during database retrievals (e.g., the source name or the center velocity and frequency of the spectrometer). In the raw data tables, approximately 1/5 of the columns have indexes. The use of indexes to optimize queries is handled automatically by the DBMS. Indexes use disk space but are essential for fast data access.

3.2.4 Documentation

In view of the extensive list of tables and columns (currently 55 and 720, respectively), we have developed an automated self-documentation procedure for the database. The procedure generates a complete list of all tables, the columns within each table, the data type and range of values for each variable, a list of which variables are indexed, and the present size of each table.

3.2.5 User-Owned Tables and Views

Most of the discussion above has referred to archival astronomical and engineering data stored in the millimeter-array database. As a general policy these raw data are never changed and when observers wish to calibrate their data, they first must make a table with a copy of the raw data. The construction of user-owned tables, usually referring to a single track, is accomplished by “bulk copy” from the view joining the raw data tables into a new, single table. These user-owned tables contain all of the columns which exist in the elementary tables (e.g., TRA, INH, BLH...), with the exception that key variables duplicated the raw tables to provide links occur only once in the user-owned tables. Within the user tables there exists one row for each spectrometer/sideband/baseline/integration combination, resulting in considerable duplication of header variables in the vertical direction. However, these tables are only created for individual tracks, and reside in the database only temporarily while the astronomer is calibrating his or her data, so the overall storage requirement is not a major factor.

When a user-owned table is made there are in fact two tables which are created: the first contains the header variables; the latter contains the continuum time samples and spectrometer data channel values in packed binary (image) format. When the tables are created the user specifies both a *name* and *version* for labeling. In the database, the names of the two tables are prefixed with “t” and “SCH” prefixes; the full names are therefore: “t_name_version” and “SCH_name_version.” Since the DBMS addresses tables with prefixes for the database name and the user id, two users may have tables with identical names and versions.

Once the data have been calibrated, the user-owned tables can be bulk copied to host operating system binary or ASCII files or output in FITS format files. Within the database the tables are tagged with the owner’s name, the creation date, and source of the data (i.e., whether it came directly from raw data tables, from another user-created table, or was read in from host files). Multiple user-created tables may also be unioned to form a single table.

A second construct, in addition to the user-created table, is the view. These are logical constructs which apply selection criteria to data in existing tables to produce a virtual table which appears to the user as a physical table on disk. The advantage of creating a view as opposed to creating a new table is that it is fast, since the logic is stored in the DBMS rather than a fresh copy of the data being created. The view is also dynamic in the sense that if the tables addressed by the view are changed after the view has been created, it will reflect those changes automatically.

In general the astronomer calibrates data on a track by track basis, creating a user-owned table for each track and then editing, displaying, and calibrating these data separately. The number of columns in user-created tables is 97. The number of rows depends on the number of integrations within the track, the number of baselines, and the number of spectrometers on each baseline. For example, a track with 100 integrations, two useful sidebands, five telescopes (ten baselines), and one continuum channel plus four spectroscopic correlators on each baseline would contain 10,000 rows ($100 \times 2 \times 10 \times 5$). When the user-owned table is created indexes are established on only two key variables (inh# and blh#). However, as the data are analyzed, additional indexes are created on the relevant columns appropriate for selecting data used by the particular analysis process. This “dynamic” indexing is extremely effective inasmuch as the indexes are tailored precisely to the needs of the moment. Once created, the index is retained (unless it becomes a burden in a subsequent operation) in case the user wishes to repeat the analysis step with slightly different selection criteria.

The DBMS also keeps track of permissions for the four basic table operations: select (i.e. read), insert (add new rows), update (change existing rows), and delete (remove rows). For the raw data tables, general users are permitted only to select data from these tables and not update or delete rows. For user-created tables, the owner has all permissions while other users are permitted only to select data. This allows one user to copy tables generated by

others but not to change the other user's table. This permission system enables both raw and user-created tables to exist in a single database without fear of corruption. This, in turn, allows easy access to all tables and provides a high efficiency in the use of disk storage since the overhead that would be required for separate databases for each user is unnecessary.

Although the above discussion implies that the entire millimeter-array dataset is contained in a single database, this is not in fact the case. A slight modification was adopted in order to facilitate and shorten the time for daily database backups and to allow for the possibility that several years after the data were taken it might no longer be necessary to keep the old data continuously on-line. We have one major database (called *mmdb*) which is permanently on-line containing all tables except the three raw data tables with largest storage requirements. Essentially all variables used for general searches are contained in the smaller tables permanently resident in *mmdb*. For the three largest tables (BLH, SPH, and SCH), data are accumulated in a new database for each observing season (called, e.g., *mm9192*, or *mm9293*). Daily database backups are then required only for *mmdb* and the millimeter database for the current year. The millimeter databases for earlier years do not need to be backed up since they are not changing. All user-owned tables are in the *mmdb* database and thus are subject to the daily backup.

4. DATA ACQUISITION

The data acquisition programs run on the VAXstation. At the end of each integration information such as telescope pointing, system temperatures, weather data, projected baselines and raw amplitudes, and phases from the analog correlator is incorporated into an ASCII SQL command on the VAX. The SQL command is then sent over the ethernet to the database server on the SPARC station and executed as an atomic transaction by calling a routine in the client VMS 'C' DB-Library. Spectral channel data from the digital correlator are packed into a binary packet corresponding to the Sybase "image" data type. Another routine is called to write the channel data to the database server. The use of atomic SQL transactions prevents the corruption or loss of data in the event of hardware or software failure. It currently takes 3 s to write all the data for a single integration across the network.

Along with the raw data we also allow observers the option to associate a number of qualifiers with a particular integration. The qualifiers are used by the calibration routines to identify integrations suitable for, e.g., gain, flux, and passband corrections. The qualifiers we use most frequently are listed below:

- aq—flux calibrator (aq=1 indicates a primary calibrator, aq=2 a secondary calibrator)
- bq—baseline calibrator
- cq—atmospheric coherence calibrator
- gq—complex gain calibrator
- pq—passband calibrator
- tq—identifies data taken with similar velocity or tuning.

These qualifiers are not mutually exclusive, and a single observation may have several qualifiers associated with it. The goal is to enable the original schedule file for the data acquisition to set up some of the reduction operations. (A side benefit of this is that observers are forced to consider more seriously the observations needed for calibrations). These qualifiers may be reset later in the data calibration process.

5. ANALYSIS AND CALIBRATION PROGRAM

An extensive program called "mma" has been developed to access the millimeter-array database for data reduction and database administration (Sec. 2.3 and Table 1). It operates on SPARC stations running the UNIX operating system and may be run either locally on the database server machine, in client mode from a network SPARC station, or as a remote login on INTERNET. It is designed for X-Windows or OpenWindows environments, but a wide variety of terminals can be used including simple ASCII devices.

5.1 Program Components

The program is menu driven from forms built using the Sybase APT development tool. The Sybase APT forms and the associated procedure routines written in APT-SQL provide an extremely flexible programming environment. The forms can have simple menus, cascading menus, numeric and rotating choice entry fields, and scrollable group fields (for listing multiple results returned by database queries). The *mma* program consists of a linked set of C, Fortran, Sybase "DB-Library," and APT routines. All graphics are handled by the PG PLOT package (Pearson 1991).

Figure 3 illustrates schematically the structure and the variety of software links used in the *mma* program. The program is started as a C program (*main.c*) which sets up structures used by the APT forms, initializes the database library routines, installs general error and message handling routines, and then loads the primary, *mma* form. Menu options on this form then trigger other forms and routines (C, Fortran, APT-SQL, and graphics). This program structure follows examples provided by Sybase with the exceptions of the graphics interface and our extensive use of the APT "system" command, which is used to start database tasks as independent processes in the host operating system. The use of C and Fortran is necessary for graphics and heavy computations such as curve fitting and for analyzing the spectral channel binary data.

5.1.1 Form Operation

While *mma* is running, the screen is divided into two sections. Along the top are menu options, while the main part of the screen is the form containing entry fields to be filled in by the program or the user. The normal way of navigating around the forms is with a mouse, which is used for the selection of menu options, and for positioning the cursor into entry fields. However, each menu option can also be triggered by a single character or function key.

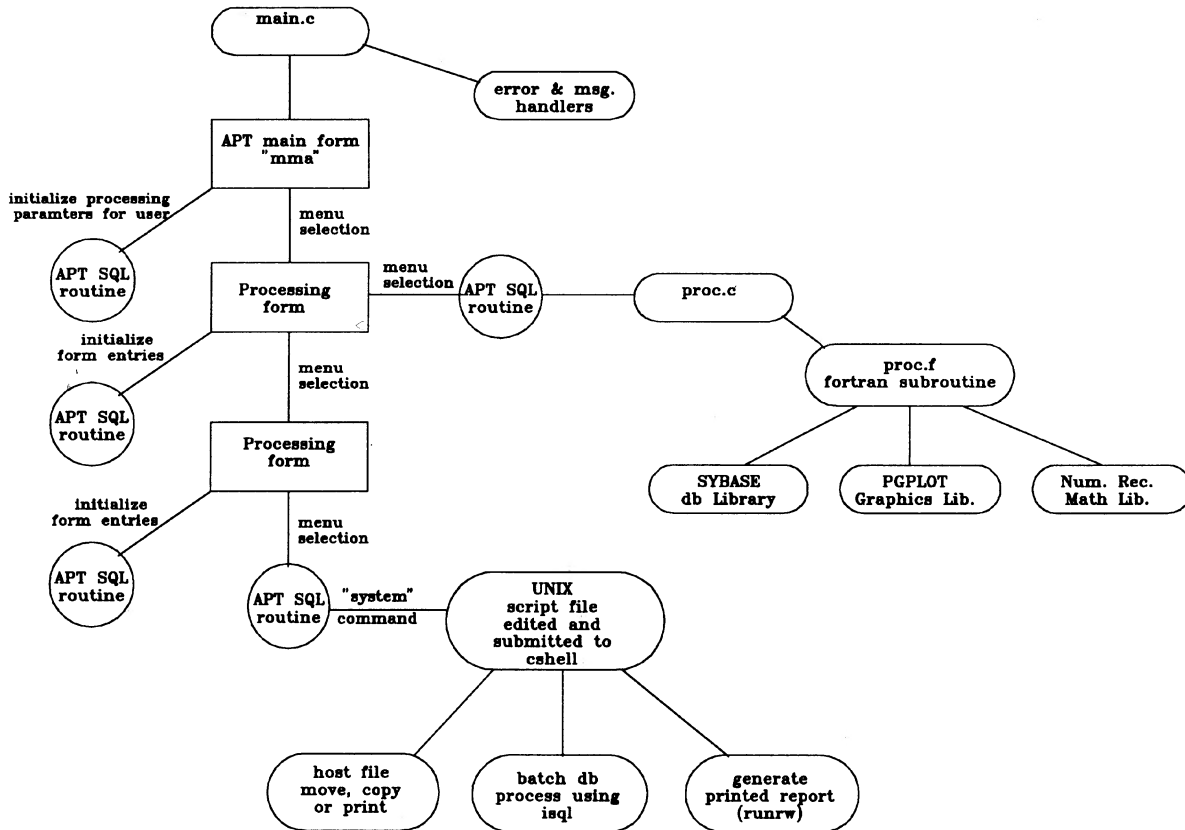


FIG. 3—A schematic diagram of the structures and program links occurring within the mma program. Fortran, C, and UNIX processes are indicated by the oval symbols. User-interface forms are indicated by rectangles and APT procedure routines called by these forms are indicated as circles.

Using APT to create a form, the properties of its entry fields are defined. For example, one may specify the variable type allowed in the field, whether the field has rotating choices, or whether any processing occurs when the cursor enters or exits from the field. Some fields display information obtained by SQL processes and are not modified by the user.

An example of processing on entry or exit from a field, and of displaying information obtained by SQL routines, is provided by the baseline correction routine in mma. The user enters the identification number of a particular baseline solution into a field on the form. On exit from the field the baseline solution is recalled from the database by an SQL command, and displayed on the form. Another case where this kind of processing can be useful is in checking whether a value input by the user is within an acceptable range. If not, the entry can be reset or an error message displayed to notify the user. All of the options and variables which appear on forms and govern the operation of specific processing routines is stored in a database table (GLOBAL) with one row corresponding to each user. Upon entry to each form, an APT-SQL processing routine reads the relevant operational parameters from GLOBAL and enters them in the form fields; upon exiting the form the same fields are saved back in the GLOBAL table. The parameters for each function are therefore “sticky”—when

the user exits and restarts mma, these operational parameters are restored.

Many of the mma forms have “group” fields which are used to print multiple results returned by database queries in APT-SQL. These groups can be set up with multiple columns for different variables, the number of visible rows being defined to fit within the space available on the form. If a database query returns more rows than can fit on the form, the invisible rows can be scrolled into the visible area using a scrollbar. An example is the flux calibration form in mma, where the user can list all the individual observations of primary or secondary flux calibrators in the active table. Individual observations can then be excluded from the computation a scale factors to convert instrumental counts to Janskys. In some tracks there may only be one flux calibrator observation, while in others there could be 30 or 40 measurements and the group fields provide this flexibility.

5.1.2 Data Filters

At the heart of any database analysis program is the issue of how the user can select a restricted subset of the total data. In the mma program the construction of user-owned tables (typically containing the data from a single track) is one means to select a subset of the database.

However, once this table has been created, the user may wish to focus on different subsets of the data within the table at different stages of the analysis process. In the *mma* program, this is done by constructing a “data filter” which can be changed at the different processing steps. This filtering is accomplished by having the program construct an SQL “where” clause to achieve the desired selection criteria. This character string is then inserted by the program in all subsequent queries to the active table. To speed up subsequent searches new indexes are constructed whenever the data filter is changed. In this way the indexes which exist on an active table are always optimized for the search conditions being used.

It is important to note that the user of *mma* need not be familiar with SQL to access the database. All SQL queries are constructed and executed within *mma* and are therefore transparent to the user.

5.1.3 Help and Logging

Two additional features to facilitate use of the program are an extensive “help” facility and logging capability for all data reduction operations. Help on the current form can be selected using the F1 function key. This loads a new form with tutorial information about the operation of *mma*. Many of these help forms are linked to additional help forms by picking highlighted words within the help text.

We have also built into the *mma* program the capability for complete logging of all program operations, and the values of all entry variables on each activated form. Every time a user starts up *mma*, enters or exits a new form, selects a menu option, or changes an entry field, this information is recorded in a database table (RLO—reduction log) in which each row is identified by the user id and time. The contents of this table may then be examined from one of the *mma* forms. An additional form of logging occurs whenever a form is entered or a menu item is selected. A message is written to the UNIX window indicating the event which just occurred. This is useful for long processes to reassure the user that the operation is initiated.

5.1.4 Fortran Routines

Heavy computations in *mma* (curve fitting, data scaling, and graphics) are done in Fortran routines (see Fig. 3). Access to the database is easiest using the Sybase DB-Library/C, but we chose to use their Fortran library in most instances since Fortran was more familiar to those of us who would be writing the software. Since a number of people have worked on the software, care was taken to ensure that the routines follow a standard structure, so that they were readable by others. Most of the database process initialization and database retrievals are carried out within the primary Fortran routines. When the Fortran routines require variables from the database, the SQL query is sent to the server as an ASCII string and rows of data are returned sequentially and bound to program variables. Communication of parameters and variables between the APT forms and the C and Fortran routines is accomplished

through the database by writing the variables into the database table GLOBAL before calling the C and Fortran routines and then reading the same variables out at the beginning of those routines.

For most of the numerical computations such as curve fitting and interpolation we have used subroutines from the Numerical Recipes Math Library (Press et al. 1989) and all graphics are done using PGPLOT (Pearson 1991). In both instances these libraries were selected because of their high reliability and their familiarity to astronomers.

5.1.5 Program Outputs

The primary goal of the *mma* program is to calibrate raw interferometric data and output calibrated amplitudes and phases. The standard output from *mma* is presently in FITS format files (Wells et al. 1981), with one source per file. These files are immediately readable by either AIPS, GIPSY, or MIRIAD. Alternatively, the tables may be saved as UNIX files using the Sybase bulk copy utility.

Another form of output from the *mma* program is printed reports generated by the Sybase report writer. In *mma*, menu selections are available to generate summary reports for individual tracks, configurations, source flux measurements, spectral line lists, investigator and project information, and an automatic self-documentation for the database tables. The most generally useful of these reports is the “track log” which summarizes all information on an individual track. At the end of each track the observer enters comments into the database concerning the quality of the track using *mma*. This information is valuable later for evaluating the data without searching through handwritten logs.

5.1.6 Database Management Functions

Within *mma* there are also several routines which perform management functions: DB backups, DB integrity checks and size estimates, maintenance of table index statistics, and automatic transfer of new data between the observatory and campus. Full backup of the changing databases at the observatory is done approximately every two days and transfer of data between the observatory and campus is done on a daily basis. The database backup onto digital audio tape requires approximately 2 hr. The daily transfer of new data typically requires about 1/2 hr using the 56 kbyte INTERNET link.

The transfer of data between campus and the observatory can be initiated from either site. The first step in the process is to query the local and remote database servers to find the maximum values for the key columns in the relevant tables at each location. Once the ranges of data existing at each location have been determined, the second step is to bulk copy to UNIX files the rows which do not exist at the other location, and then to FTP these UNIX files to the other computer. The last step in the process involves bulk copy of the transferred files into the local databases and insertion of the new rows into the local tables. During this process checks are done to ensure that the row range read into the database matches output from the other database.

In general, most of the data transfer is in the direction from the observatory to campus; two exceptions are the tables containing flux measurements and baseline solutions which are useful to users at both locations. In the few tables for which this situation arises the two database servers are assigned different allowed ranges for the key column variable in order to distinguish rows entered at the two locations.

5.2 Typical Calibration Sequence

To illustrate better the operation of the database and the program capabilities we now briefly outline the sequence of operations involved in calibrating a track of raw data. The process may be broken down into several discrete steps: (1) determining which track contains the desired data, (2) generation of user-owned tables with a copy of the raw data from that track, (3) derivation of the relative gain functions for the interferometer amplitudes and phases, (4) derivation of the spectral passband response on each baseline, (5) application of the derived gain and passband functions to the data, (6) derivation of flux scale factors and their application, (7) measurement and archiving of flux measurements for sources in this calibrated dataset, and (8) output of a FITS format file.

To locate the track of interest the user can activate forms which list the tracks including a specified source or combination of sources, or which include data for a specified observing project. Once the filter has been set for the appropriate track range of interest, the user then selects a menu option which will create the two user-owned tables containing header and channel data. Creating the two tables typically requires a couple of minutes. In the event that some of the data within the track are unwanted, the user can delete rows from the two tables, or change the calibration qualifiers which were set during the original observations.

The second step of the calibration is to derive the gain and passband corrections for data within the track. The gain as a function of time (or elevation) is derived by fitting a polynomial to the continuum measurements of a calibrator source. Upon entering the gain form the user selects menu options to list either the default gain calibrators, or to enter the name of a specified calibrator source. The user then selects whether gain functions are to be derived as a function of time or elevation for the amplitude and phase, and the order of the polynomial fits. Another menu selection will then initiate a call to a Fortran routine which plots the continuum amplitude and phase as a function of time or elevation and then fits polynomials to these data. At the end of the process the polynomial function is stored in a database table (GFN) and parameters describing the range of applicability of this gain function are stored in a second table (GAI). For derivation of the spectral channel passbands, a different form is called but the general procedures are similar to those for gain calibration.

The gain and passband functions are then applied to data in the user-created tables as the fourth step of the calibration process. Before this application is initiated, the

program first checks that these data have not already been calibrated. If they have the program will print a warning and query the user for permission to continue. At the end of the gain and passband application the fact that the calibration has been applied is recorded in the database. The application of the gain and passband calibration to a single track of data typically requires about 3 min. The separate tables associated with the calibration process and the connection between these tables are shown schematically in Fig. 4.

The last step in the calibration process is the derivation of flux scale factors to convert the raw amplitude counts into Janskys. Within the track there should exist observations of planets (primary calibrators) and quasars (secondary calibrators) which can be used to derive these scale factors. The flux scaling form provides the user with the options of using either primary or secondary calibrators. Once the calibrator sources have been specified, the program will search for all observations of the sources, list them on the screen, and permit the user to turn undesired observations "off." In deriving the flux scale factor for the secondary calibrator sources, the program searches the flux archive table (FLU) to find observations of the same source at a similar frequency and within a specified time range of the observations being calibrated. After the user is satisfied with the list of observations to be used, a menu option is selected to compute average scale factors for each sideband and baseline combination. The derived scale factors are displayed, and a final menu option may be selected to apply the flux scaling.

At this point the raw data have been gain, passband, and flux calibrated. The user might then wish to plot the average spectra for each source on each baseline or to display amplitude, phase, and coherence as a function of time for all "bright" sources. In any case, satisfied that the calibration has been derived and applied correctly, the user can measure the average fluxes for sources within this track and store those flux measurements in the database flux archive table (FLU). When the flux measurements are entered in the archive table the user name is also recorded. Thus an individual user can restrict searches in the archive table when deriving flux scale factors to just their own archived measurements or make use of other observers' measurements.

In most millimeter-array projects, the final astronomical image will be made up from data obtained during several tracks. If this is the case, the individual tracks might be calibrated as described above and then the separate calibrated track tables combined into a single table containing all of the tracks. The last step is then output of a FITS file containing the data for the desired source(s).

5.3 Batch "Auto" Processing

Although the *mma* program was designed to be run interactively, there is a facility to enable batch processing. This mode is effective in instances where one wishes to process a number of similar datasets in the same way. This auto processing is accomplished in three steps. First, the

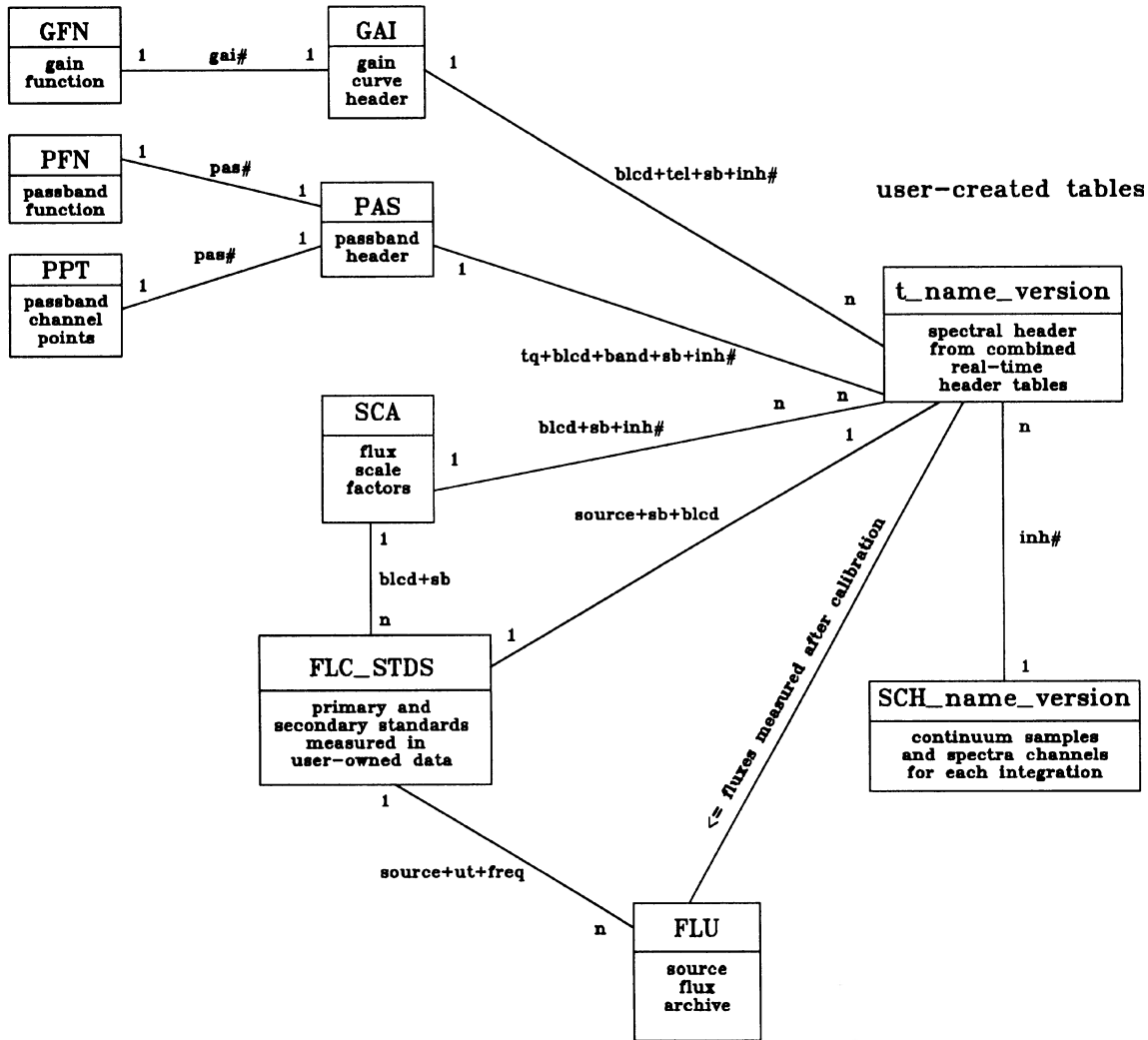


FIG. 4—A schematic diagram of tables associated with the calibration sequence of interferometric data. The table links and key columns are shown as in Fig. 2.

mma program is started up in “teach” mode in which a sample data set is treated and all mouse movements, menu selections, and keyboard strokes are recorded in a file. Second, this file is edited using the UNIX stream editor in order to substitute new values for specified parameters entered during the teaching run. Last, the mma program is restarted in “playback” mode where all instructions are taken from the edited file rather than the keyboard or mouse. If additional processing runs are desired, the second and third steps can be repeated.

One situation in which this auto processing is frequently used is in the reduction of multiple tracks of data obtained on a single observing project. Usually these tracks will consist of data taken in different telescope configurations but the general sequence of observations will be similar in each track. Once the data calibration options have been optimized based upon experience with reducing the first track, these instructions can be recorded in an auto-

processing teaching run and then applied in edited-playback mode to the later tracks.

6. SUMMARY

In the foregoing sections we have related our experience with setting up a commercial relational database management system and menu-driven database access program for archiving, calibration, and inspection of real-time astronomical data. Our general experience using commercial software has been extremely positive—providing much greater power and flexibility than would have been possible with a home-grown system. The flexibility to change is an extremely valuable characteristic in state-of-the-art scientific systems where the equipment and computer systems are constantly being upgraded. Perhaps most important in the long run is the level of integrity and accessibility provided by the commercial software. Over the last two years

while the software has been installed and under development no astronomical data have been lost or corrupted, and in fact the database software was used on several occasions to correct errors resulting from faults in the data acquisition system.

The total effort invested in programming the database software, setting up the millimeter-array database, and testing the *mma* program is approximately two person-years. Although this is a considerable level of effort for a small group, probably a much greater effort would have been required to maintain the old-style file archiving system and calibration program. At the same time the end result would have been considerably less powerful and probably require much greater long-term maintenance.

The Owens Valley Millimeter array is supported in part by NSF grant AST 90-16404.

REFERENCES

- Hurson, A. R., Pakzad, S. H., and Cheng, J. 1993, *Computer*, 26, No. 2, 48
- Pearson, T. J. 1991, "Caltech VLBI Analysis Program," *BAAS*, 23, 991
- Press, W. H., Flannery, B. P., Tenkolsky, S. A., and Vetterling, W. T. 1989, *Numerical Recipes* (Cambridge, Cambridge University Press)
- Scott, S. L., and Finch, R. P. 1992, in *Proceedings of Workshop for Remote Observing*, in press
- Scoville, N. Z., Carlstrom, J. E., Padin, S., Sargent, A. I., Scott, S. L., and Woody, D. P. 1993, *IAU Symposium 140—Millimeter and Submillimeter Interferometry*, ed. Ishiguro (in press)
- Wells, D. C., Greisen, E. W., and Harten, R. H. 1981, *A&AS*, 44, 363