Discussed is an interactive graphic system for compiling air navigational data on aeronautical charts.

Eliminating many tedious manual updating operations, chart manuscripts can be interactively created and revised at a graphic console. Emphasized are design details of the graphic files, which are specific digital compilations of aeronautical data for individual charts.

Interactive aeronautical charting

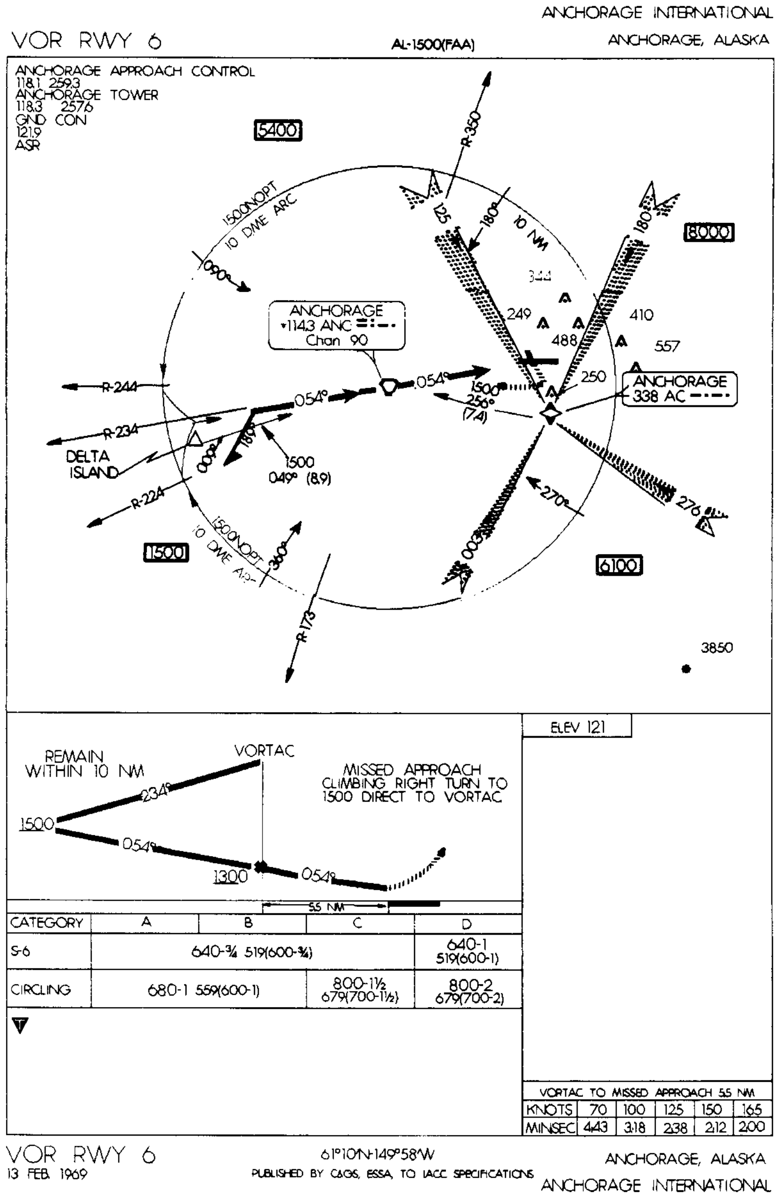
by J. H. Luetje

Clear, accurate, up-to-date aeronautical charts aid modern complex aircraft in traveling safely and efficiently at high speed over precisely defined airways. It is becoming increasingly difficult on a rigid production schedule to incorporate by hand the many changes required for these charts with their widely ranging detail and geographical area requirements. This paper discusses a solution to the problem whereby a computer-assisted cartographic system maintains a base of aeronautical data, allows editors (called chart compilers) to interactively create and revise chart manuscripts at a graphic console, and produces transparencies suitable for printing. A sample output of the system is shown in Figure 1.

The overall system, described first, is a computer-aided version of manual techniques. Included are the organization and maintenance of the nongraphic data base, and methods for translating individual chart data to visual outputs. The products of the system are color-separated transparencies for 2900 different aeronautical charts, organized into 14 series. The system handles about one-hundred daily changes, each one of which may result in changes to several charts. These changes report such factors as the opening or closing of airports, changes in radio frequencies, construction of new runways, and the establishment of new airways.

The author emphasizes design details of the graphic files, which are used for collecting specific aeronautical information for the individual charts. This information is in the form of specialized digital compilations derived from the basic data base. Here he

Figure 1 Example of plotted output



describes the structure, storage, display, and output drawings based on data in the graphic files.

Currently several skills are required for aeronautical chart production by the U. S. Coast and Geodetic Survey for the Federal Aviation Administration. Using the primary information sources—Airspace Dockets and the National Flight Digest—changes are posted as written notations on copies of aeronautical chart manuscripts. On a scheduled basis, these changes are manually translated into instructions for draftsmen who produce the color separations required for printing. Each step involves tedious hand work, human judgment, and is subject to human error.

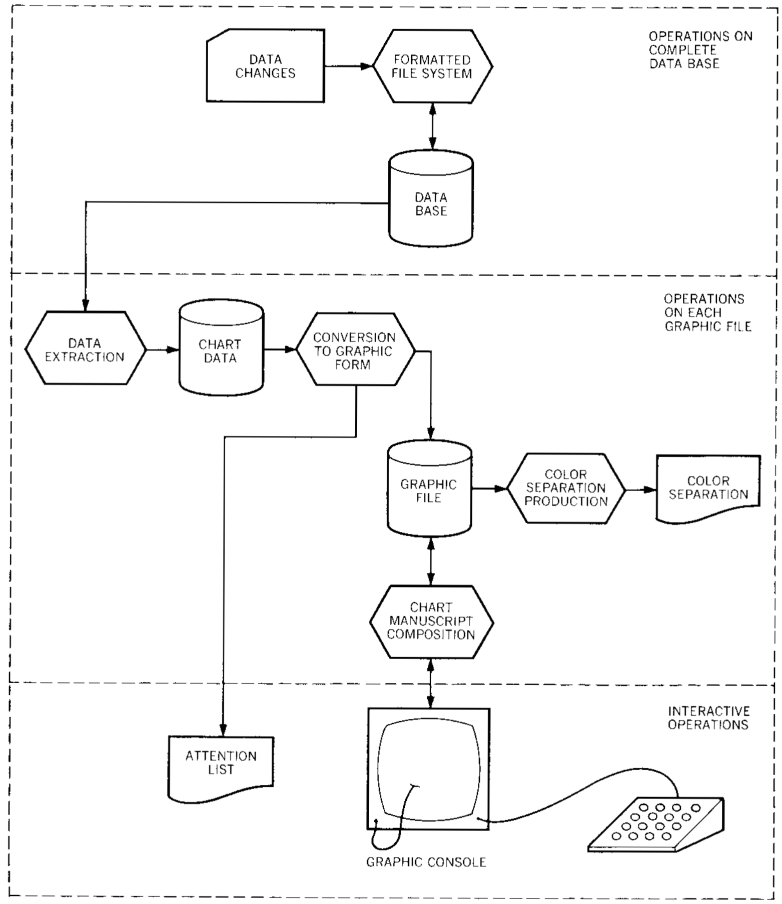
The interactive cartographic system is designed to correspond with these manual methods. The system utilizes the editor's and the draftsman's aesthetic skill and judgment in laying out changes on the charts at a graphic console. The results of this interaction are then drawn automatically with high resolution and precision. However, the laborious data base maintenance, posting changes, and chart content are automatically handled by the system. Human interaction with the system is accomplished by a simple display language (not discussed here in detail) and a light pen at an IBM 2250 display console.

The system under discussion does not produce the relatively stable natural topological features such as coastlines, rivers, and mountains; man-made aeronautical navigation information is the object being manipulated. However, the system is adaptable to the production of aeronautical charts that are now in the research stage to meet the growing aviation requirements. The work discussed in this paper was done under contract for the U. S. Coast and Geodetic Survey and might be regarded as complementary to other work in the retrieval of geographic information.[1]

Overall system design

Aeronautical charts can be conveniently and economically described by a nongraphic data base and a derivative graphic file structure. More important to those responsible for publishing the charts is the fact that the data base and graphic files are created and main-
tained without a detailed knowledge of the data base and graphic

Figure 2 Overall system design



and coastal waters consist of more than 36,000 aeronautical features organized into six files requiring a total of 25 million bytes of storage. Updates occur at a rate of about 3000 per month. The data base maintenance function is performed by the System/360 Formatted File System (System/360 FFS),[2] which operates on card-update records to maintain the six files: navigation aids, reporting points, airways, controlled airspace, airports, and flight hazards. Since reporting points are positionally defined with respect to navigation aids, and airways are defined between navigation aids and/or reporting points, the data base files are interrelated. Special exits from the System/360 FFS handle the dependencies so that, for example, a position change to a primary navigation aid is propagated through the airway network predicated on that aid.

Because the data base supports all chart series produced for the Federal Aviation Administration, and each series of charts has unique formatting requirements, the data base contains no graphic data or chart-dependent parameters. Typically, each aeronautical feature is described by a record that contains a uniqueness key, geodetic coordinates (latitude and longitude), control dates, and fields of descriptive data.

data extraction

Data extraction is the function of retrieving records from the data base, which contains information on updated features to be applied to a chart manuscript. System/360 FFS also performs this retrieval function. Control parameters, i.e., statements that specify conditions for retrieval (or "hits"), and report instruction statements that define the descriptive data fields to be returned on hits are provided for each chart manuscript being updated.

In general, the retrieval statements establish three criteria that must be met to qualify as an update to a particular manuscript:

- Information is germane to this type of chart.
- Information has been updated in the data base since the last chart manuscript publication.
- Information is located within the area covered by the chart.

Queries are thus executed by directing System/360 FFS to examine specific fields of descriptive data, specific control data, and the geodetic coordinates, respectively, of each feature.

conversion to graphic form

When converting to graphic form, the updated data fields of the features retrieved for a chart are isolated, and the established format for the chart series is applied prior to posting changes to the chart manuscript. In the computer-assisted system, the chart manuscripts are stored as digital data structures, called *graphic files*, especially designed for this application. There is a graphic file for each chart. The storage requirements for the graphic files vary according to chart size and data density; the graphic file for a small chart as shown in Figure 1 requires 50K bytes, whereas the largest, most dense chart requires 750K bytes. Graphic files are discussed in detail later in this paper.

In converting to graphic form, the system formats the data. This requires an algorithm (discussed later) for anticipating how an item of aeronautical data will be portrayed on the final chart. After appropriate formatting, the data is entered into its particular graphic file. In doing this, charting rules and standards are considered. For example, the symbol for a given type of a navigation aid feature is inflexibly specified for a given chart series, including its color and its positioning on the chart. On the other hand, the positioning and symbolization of annotation data to the feature, although governed by standards, is less critical and is subject to review and reassignment for improving chart clarity.

As updates are posted to the graphic file, an *attention list* is created and routed to the responsible editor for his review and guidance in

Figure 3 Detail from attention list

FEATURE:	ANCHORAGE VORTAC	BLUE	62D38MN156D43MW		
LINKED TO:	DELTA ISLAND RPT.				
ACTION	NAME	SYMBOL	FONT	SIZE	TEXT
OLD	VORTAC SYMBOL	A43			
OLD	COMPASS ROSE	A21			
OLD	RESTRICTION	C01			
OLD	COMMUNICATION BOX	C11			
OLD	FEATURE NAME		E01	8	ANCHORAGE
OLD	FREQUENCY		E01	8	116.3
CHG	FREQUENCY		E01	8	*114.3
OLD	IDENT		E01	8	ANC
OLD	IDENT IN CODE		E03	8	ANC
OLD	CHANNEL		E01	7	CHAN 90
OLD	LEADER LINE	B09			

completing the portrayal at the graphic console. Details of such an attention list, which relate to the instrument landing procedure chart in Figure 1, are shown in Figure 3. In this figure, features being changed are identified by name (Anchorage), type (VORTAC), color (blue), and location in latitude and longitude (62D38MN156D-43MW). All symbols and annotation associated with a change are summarized on the attention list for the editor so that he can quickly determine the graphic impact of the change.

chart manuscript composition

Interactive chart manuscript composition and revision completes the updating of chart manuscripts, a functional diagram of which is shown in Figure 4. By using an interactive editing language to operate the display console, the editor accesses a library of graphic files to find and display areas of a manuscript requiring his attention. He then completes the revision of the chart by interactively modifying the displays at the console, and requests the plotting of reproducibles for the completed manuscript.

Within the editing language there are three broad categories of commands that are called by the syntax processor. Data structure management commands give the editor access to individual graphic files for retrieval and updating in the graphic file library. Display management commands give him control of the area and content of each display. Display modification commands allow the chart editor to add, delete, or change any displayed information, i.e., alphanumeric data, symbol, color, position, or angle.

A 2250 display of chart data retrieved from a graphic file, as requested by the editor, is created as follows. A program examines the symbols imbedded in the retrieved data. This program then looks up the symbols in the 2250 *symbol dictionary*, and applies dynamically determined scaling, rotation, and translation parameters to the symbol models found. The 160 predefined dictionary symbols require 8200 bytes of storage in the 2250 symbol dictionary.

Figure 4 Manuscript composition

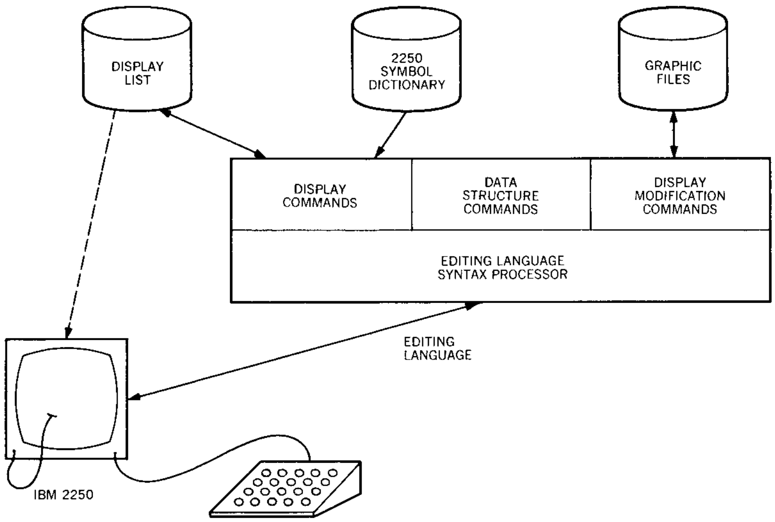


Figure 5 Example of displayed output

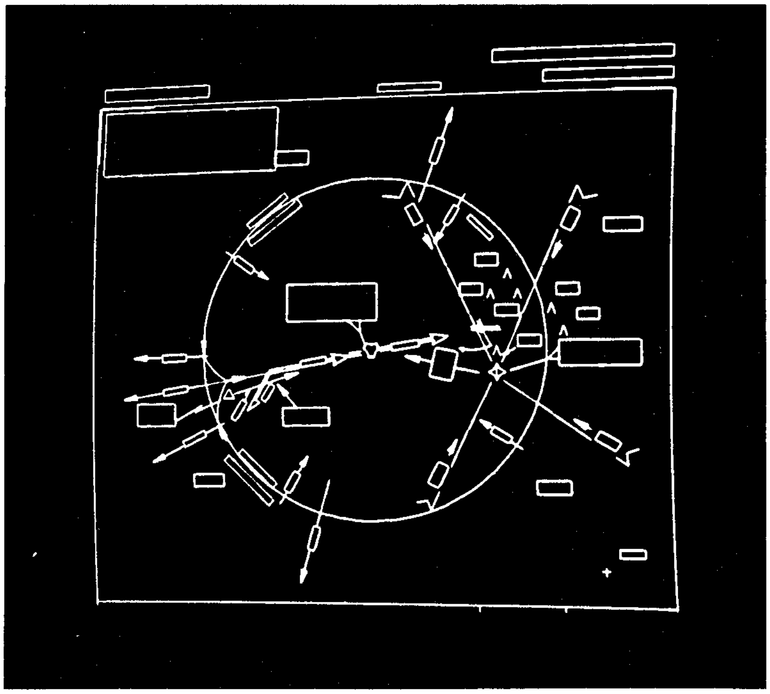
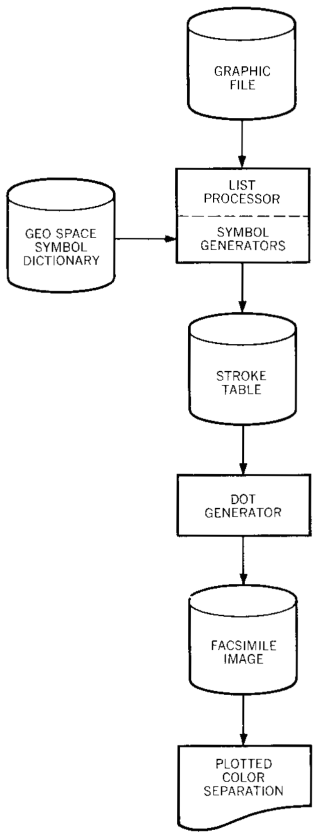


Figure 5 is a 2250 display of the upper part of the manuscript for the chart shown in Figure 1. Note that the 2250 uses stylized versions of the actual chart symbols to conserve symbol dictionary space and to improve system response time. Because alphanumeric annotations on the charts are made in variably spaced type and presented at any angle, they are not displayed on the graphic console directly. Alphanumeric information can, however, be displayed on request at the bottom of the screen.

color separations

The final function performed by the computer-assisted system is that of color separation production, illustrated functionally in Figure 6. In this phase, the reproducibles of a chart required for the lithographic printing process are plotted on successive color passes of the chart's graphic file. A Geo Space[3] high-speed optical plotter has been used for making true-scale, reproducible mechanical drawings as photographic positives on transparent film base.

Figure 6 Color separation production



The plotting of color separations is similar to the 2250 display procedure in that symbols and their color codes are retrieved from the graphic file by the list processor. These symbols are looked up in the *Geo Space symbol dictionary*, and appropriate scaling, rotation, and translation parameters are applied to the models by means of symbol generators. The symbol models are resolved into strokes. These are the inputs to the dot generator, which produces the facsimile image in the form of finely spaced dots as required by the plotter. In performing this function, the plotter automatically draws the symbols on the basis of the format that was derived interactively at the display console. In the Geo Space symbol dictionary, the 160 predefined symbols require 23,000 bytes of storage, reflecting the increased resolution of the plotter symbols over those interactively displayed. The difference in resolution may be seen by comparing Figures 1 and 5, showing the plotted output and the display output, respectively.

Graphic files

The key elements in the computer-assisted cartographic system are the chart manuscripts, which are digital data structures called "graphic files." Collectively, they make up the library of graphic files. The author now discusses his graphic file and library design principles.

A graphic file contains a digital description of all the attributes that make up the features depicted on a specific chart with the associated format information for their display. The basic system design principle is that the content of charts is under program control, whereas the manner of portraying these contents is under the editor's control. The design of the graphic file supports the following three major functions performed by the system in fulfilling its basic objectives:

- Updating the manuscript
- Creating displays for output devices
- Interactive editing of charts

updating the manuscript

Since a graphic file contains all the data for a specific chart, it is designed to permit ease of maintaining the manuscript from data base revisions. In other words, the data base is updated by passing on the revision to the graphic files and incorporating them into the manuscripts that cover the affected aeronautical facilities. The capability of program control of the contents of charts also permits many kinds of updating to be effected without the editor's attention. For example, the correction of a radio frequency can be done simply by the exchange of a data field in the graphic file. In the design of the file, duplicated data appearing on a chart is not redundantly stored in the graphic file. Hence, the replacement of the single field containing the radio frequency results in that information's being replaced wherever it occurs on the chart.

output devices

Another function of the graphic file is to permit the rapid and selective synthesis of the orders that can drive a variety of graphic output devices for the editing, display, proofreading, and production of color separations. Graphic symbols described in the graphic file are color coded, but not color separated. This distinction not only permits the extraction and display of data by color, which is required for the plotting of the reproducibles, but also allows for the composite color display of the manuscript, which is useful during the compilation process. The graphic file is the source of data to any graphic output device, because the graphic data is described independently of specific display devices. The symbol dictionaries contain the linkages to graphic display devices, such as the graphic console and the plotter.

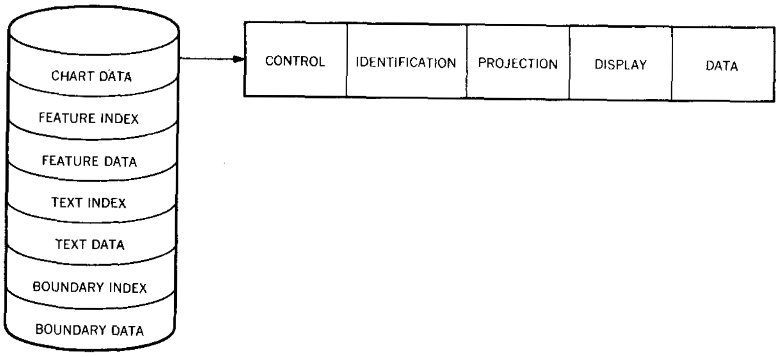
interactive editing

The third function that the graphic file is designed to support is the interactive editing provided to the editor working at his graphic console. A special data structure is required for this support in order to assure positive identification, retrieval, and updating of the data that the compiler is operating on. High speed is required to avoid degrading the editor's performance. The graphic file uses threaded list structures and subordinate lists that link the data comprising those graphic entities that are of interest to the editor.[4] Thus all data required for any manipulation is quickly coordinated by following the appropriate list.

Each graphic file is a direct-access data set that, for reasons of fast access and economy of storage, is divided by application programs into the following seven direct access data subsets:

- Chart data
- Feature index
- Feature data
- Text index
- Text data
- Boundary index
- Boundary data

Figure 7 Graphic file organization



The chart data subset contains data set control information for the graphic file as a whole, as well as a concise description of the chart represented by the graphic file. By carrying such parameters as chart series, geographic area of coverage, projection, and scale with the graphic file, the programs that operate on this file can be generalized to be independent of individual charts and series.

Chart data consists of five fixed records as shown in Figure 7. The control record, shown in the strip in the figure, contains the allocation of tracks to the other six data subsets and the number of unused tracks in each. Their allocation depends on the characteristics of the chart described by the graphic file and may be changed as required.

Chart name (and the series to which a chart belongs) and the dates of last data base revision, editor attention, and color separation plotting (reproducible) are carried in the identification record.

The projection record contains parameters required for projection conversion, namely, type of projection (charts use the Lambert conformal conic projection with two standard parallels), central meridian, scale, and geodetic reference. These parameters are used to transform the geodetic cordinates (latitude and longitude) carried in the data base to the Cartesian coordinates required in the graphic file. Geodetic positions are mapped onto the charts to an accuracy of 1/256 inch.

The display record carries the information needed for positioning the chart display as a whole on either the graphic console or on the plotter. Parameters for controlling translation and rotation of axes as well as unit scaling are included for both types of output devices.

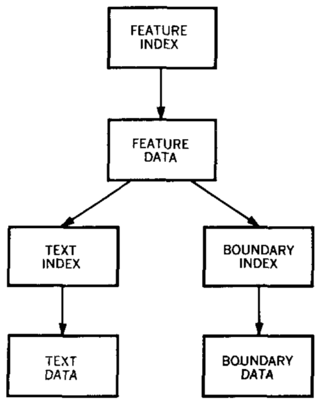
Last in the chart data subset is the data record, which contains the list of points defining the boundary of the chart as well as the predefined size of major subdivisions (or skeletons),[5] and the overlapping pattern of these skeletons on the chart. This information is necessary in assigning features to areas smaller than the chart itself. (A chart may be as large as 48 by 60 inches.)

The remaining six data subsets of the graphic file in Figure 7 are related by internal references as shown in Figure 8. This relationship is based on the fact that the items to be charted are aeronautical features. Thus, just as these features are the units of data in the data base, the aeronautical feature is the data unit of the graphic file, and therefore its integrity must be preserved. The six data subsets are bound by a threaded list structure. More precisely, there are as many lists as there are features.

The feature index consists of as many records as there are features. Thus each index record serves as the head of a list, and contains the feature name, type, data base key, graphic file key (i.e., the list pointer), and a set of flags. The flags indicate in which of the major chart sections (or skeletons) the feature appears. In retrieval, for a feature to qualify as a hit in on-line editing, or for the location of a feature to cause a data base revision, only a sequential scan of the feature index is required. Since the most dense chart depicts no more than two thousand features, this process is quite fast.

Feature data is the graphic description of the attributes that comprise each aeronautical feature. Each attribute of an aeronautical feature is reflected in a record, and all such records for a feature are bound in the list structure for that feature. Aeronautical feature data is organized into fixed-length records so that the list can be readily built. For this reason, variable length text and boundary attributes, are segregated from the feature data subset. By following its list, the retrieval of all attributes for an aeronautical feature is also fast because the average length of a list is twenty-five records.

Figure 8 Graphic file data connections



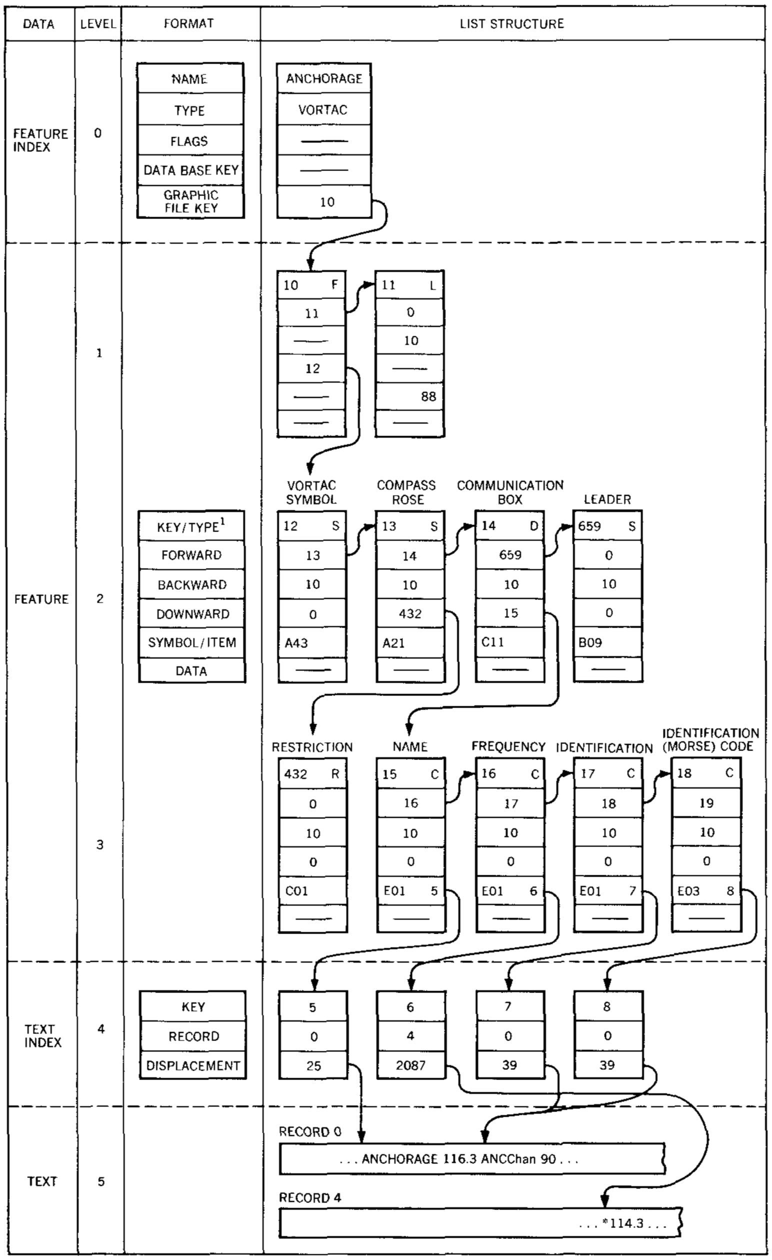
The separate storing of variable-length text and boundary items into text and boundary data, as indicated in Figure 8, speeds access to the graphic file. The intermediate indexing of these items in the text and boundary indices eases the data updating problem. Moreover, to conserve storage, this technique permits the removal of redundancies, which are common in the text.

graphic file list structure

A detail list structure for the graphic file is illustrated in Figure 9. As previously mentioned, the head of a list resides in the feature index. The graphic file key of a feature points to the feature data. This key, or pointer, is determined at the time the records of the feature are first introduced to the feature data subset, and like all other pointers referenced below, it is actually a relative record number within the data subset. Hence the retrieval of records during a list-following operation is straightforward.

There are six record formats defined for the feature data subset, one for each of the six types of defined entities. The feature index record

Figure 9 Example list structure



¹F FEATURE, L LINK, S SYMBOL, D DATA BOX, R RESTRICTION, C COMPONENT

points to a list headed by a feature entity that potentially contains link entities. At a subordinate level to a feature entity is a list of symbol- and/or data-box entities. Subordinate to a symbol entity may be a list of restriction entities, and subordinate to a data-box entity is a list of component entities. There are references in the records of symbol and component entities to boundary and text data.

All six record formats contain fields for the pointers that control the list structuring. Each record has a key field containing its relative record number. Three additional fields contain the forward, backward, and downward pointers. The forward pointer points to the next entity on a list at the same level in the hierarchy. The downward pointer points to the first entity on the subordinate list. The backward pointer always points to the parent feature entity, i.e., the aeronautical feature for which it was originally placed in the feature data. A zero value pointer indicates the end of a list.

The list structure is open-ended so that additional data can be added to a feature by the editor as he deems necessary to clarify the portrayal of information. Moreover, it is possible for the list of one feature to thread into the list of another, so that the editor may duplicate or combine data graphically without storing redundancies. The list structuring pointers in the feature data thus define relationships among entities that make up each feature. Data fields contain the parameters that specify the graphic characteristics of these entities.

Feature entity records contain absolute placement information for each feature. This information includes the charted position of the feature, the angular separation of the feature's longitude from the chart's central meridian, and the magnetic compass deviation at the feature's position. The contents of these fields cannot be altered by the editor because they represent prime data on which the graphic data for the feature is based.

Link records describe a predefined positional relationship between two features. The data fields in the record identify the linked feature as well as the direction and distance to it. This information can be retrieved and graphically utilized by the compiler at his discretion.

Symbols (not including individual characters of text) are described in a symbol entity record. In addition to identifying the symbol code, which is the link to the model in the symbol dictionary, the symbol entity record contains fields that describe each symbol's position, orientation, and color. The position is given in the chart's coordinate system and the orientation is given with respect to either the chart's coordinate system or the projection grid, depending on charting standards. If the symbol being defined is a boundary, the symbol entity record references the list of boundary points contained in the boundary data subset by item number, and identifies the symbol dictionary model to be applied between consecutive boundary points.

Data box records collect all fields of annotation that collectively form a graphic unit, and also describes the symbol that encloses the fields to indicate the bounds of their portrayal. Figures 1 and 5 show that all collections of annotation are bounded by data box symbols, but many of these data box symbols are assigned a color that is not printed as part of the finished chart. The data fields required for graphic parameters are similar to those described for the symbol record.

Restriction records specify a rectangular area where the parent symbol is to be erased or restricted from printing. Fields are defined to allow the specification of any size rectangle at any orientation. These records are created in response to the editor's command to restrict this symbol from over-printing that symbol.

Component records describe each item of annotation. Normally the annotations are fields of text, but symbolized annotations, such as underscores, are also permitted. For a string of text, which resides in the text data, the component record provides fields for type style and point size, item number, beginning character position, and number of characters. To provide for type style, the symbol dictionary contains a model of each alphanumeric character in each font. The position of an item of annotation is specified in a coordinate system within the parent data box. Thus, when the items are positioned within the box (i.e., when the box is formatted), the box may be manipulated as a unit without disturbing its format.

An example of the structure of the text index and text data are indicated in Figure 9. Each active string of text is represented by a record in the text index, which correlates item numbers with their virtual addresses in the text data. The text data thus consists of packed strings of characters that have been analyzed to remove redundancies.

The boundary index and boundary data are similar in structure to the text index and text data. Each active string of boundary elements is represented by a record in the boundary index, which correlates each element number with its virtual address in the boundary data. In the boundary data, however, the boundary elements are structured into three types of records—starting, line, and arc element records. A starting element record identifies the first point on a boundary or a discontinuity. A line element record identifies a point that is to be connected to a previous point by a straight line segment. An arc element record identifies a point to be connected to a previous point by a circular arc. In the case of the arc, coordinates of the center and direction of movement are also given.

Concluding remarks

The computer-assisted aeronautical charting system replaces many tedious operations in manually up-dating aeronautical charts. By providing interactive communication with the display, the system emphasizes the editor's (chart compiler's) real contribution—his ability to aesthetically and efficiently resolve conflicts in two-dimensional data presentations. Therefore, the concepts discussed here are potentially transferable to applications involving the two-dimensional mapping of structured data, e.g., newspaper page layout, public utility mapping, and chemical and other such processing plant design.

CITED REFERENCES AND FOOTNOTES

1. J. D. Jacobsen, "Geometric relationships for retrieval of geographic information," *IBM Systems Journal* 7, Nos. 3 and 4, 331–341 (1968).
2. D. W. Daetwyler, "National information processing system (NIPS) System/360 Formatted File System," *Proceedings of the Information Systems Symposium,* 41–52, International Business Machines, Washington, D. C. (1968) (May be obtained from the IBM Federal Systems Division, Gaithersburg, Maryland.)
3. Registered trade mark of the Geo Space Corporation, Houston, Texas.
4. L. G. Roberts, "Graphical communication and control languages," *Second Congress on the Information System Sciences,* Spartan Books, Washington, D. C., 211–217 (1964).
5. J. L. Pfaltz and A. Rosenfeld, "Computer representation of planar regions by their skeletons," *Communications of the Association for Computing Machinery* **10,** No. 2, 119–125 (February 1967).