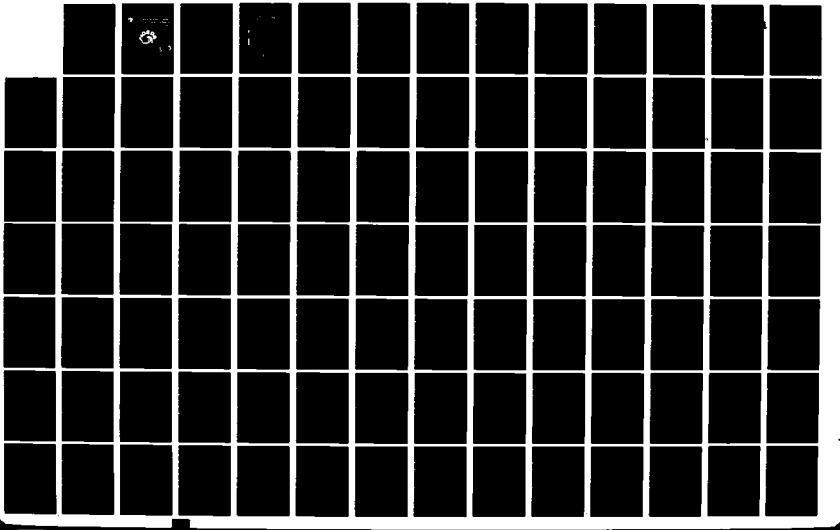
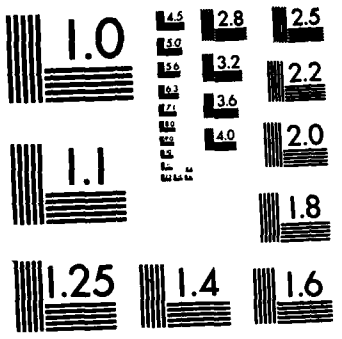
AD-A126 241 ADVANCED TYPE PLACEMENT AND GEONAMES DATABASE: 1/2
COMPREHENSIVE COORDINATION PLAN(U) PLANNING SYSTEMS INC
SLIDELL LA A E BARNES ET AL. JAN 83 NORDA-TN-189
UNCLASSIFIED N00014-82-C-0726 F/G 5/2 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

(12)

NORDA Technical Note 189

Naval Ocean Research and
Development Activity
NSTL Station, Mississippi 39529

Advanced Type Placement and Geonames Database: Comprehensive Coordination Plan

Prepared for:
NORDA Code 550
Mapping, Charting and Geodesy
Program Management Office

Sponsored by:
Defense Mapping Agency
HQ STT

R. Brown
A. Zied
A.E. Barnes*
E.C. Gough*

Mapping, Charting and Geodesy
Ocean Science and Technology Laboratory

*Present Address:
Planning Systems, Inc.
Slidell, Louisiana 70458

January 1983

EXECUTIVE SUMMARY

In FY82, the Pattern Analysis Branch, Mapping, Charting and Geodesy Division of the Naval Ocean Research and Development Activity (NORDA) began a subtask for the Defense Mapping Agency (DMA) entitled, "Advanced Type Placement and Geonames Database System Development." This effort will develop systems to address four interrelated aspects of computer automated names and symbol handling:

- Inputs: information extraction and recapture/restructuring of cartographic data from analog records and graphic documents.

- Data Storage: design/management/maintenance of a very large database of geographic names and their relationship to cartographic features independent of specific products. This database/information source will support a variety of DMA products, including maps, charts, and gazetteers.

- Data Manipulation and Editing: advanced symbol processing, document formatting, data file searching, statistic generation, etc.

- Output: names and information placement on maps, gazetteers, etc., and the associated data selection, formatting, scaling, font choice, etc., for specific products.

DMA requested that NORDA combine four original DMA requirements dealing with these areas into the current comprehensive Subtask. The original DMA Requirements Statements (see Appendix B) were listed as:

- Requirement 1: Automated Alphanumeric Data Entry System
- Requirement 2: Geographic Names Database System
- Requirement 3: Advanced Symbol Processing System
- Requirement 4: Digital Type Composition and Placement System

This Advanced Type Placement and Geonames Database Subtask is scheduled for performance during FY82-86. During the first year (FY82) funding ($40K) was provided to generate an initial Comprehensive Coordination Plan (CCP) for the technical development and integration of the above automated names capability; this NORDA Technical Note presents the results of that effort. The second year's effort (FY83; $90K) will build on this information and will generate detailed plans and functional descriptions (FD) for each specific system. This two-year (FY82-83) planning

FUNDING ESTIMATES

	FY 83		FY 84		FY 85		FY 86		FY 87		SUBYEAR TOTALS	
	EQUIP	MANPR	EQUIP	MANPR	EQUIP	MANPR	EQUIP	MANPR	EQUIP	MANPR	EQUIP	MANPR
AADES	--	3MM	81K	12MM	119K	33MM	--	37MM	--	10MM	200K*	95MM
GNDBS	5K	7MM	146K	7MM	100K	12MM	--	18MM	--	15MM	251K*	59MM*
ASP/ DTC&P	--	3MM	37K	3MM	25K	6MM	69K	12MM	45K	6MM	176K*	30MM
FY TOTALS	100K		418K		601K		538K		262K		1919K	

These costs do not include Cost of Data Capture; the hardware items are for evaluation/development system purposes.

TOTAL PROJECT COST

sequence provides (1) an opportunity for DMA and NORDA to investigate and exchange information concerning issues, functions, and technical areas involved in the overall development project; and (2) a two-level documentation procedure outlining Production Center requirements and project task elements (phase one resulted in the CCP, this technical note; phase two will present a complete Implementation Plan (IP) describing engineering task elements, development milestones, etc.). In the third year, the effort will begin actual system design based on these FDs. The outyears will develop, integrate, and deliver the systems to the DMA Centers.

This NORDA Technical Note broadly covers the technical issues associated with the systems' functional requirements; intersystem interaction; common technologies in hardware, software, databases, and machine intelligence. The information for this technical note was collected under contract by Planning Systems, Inc. (PSI). PSI also performed the initial analysis of the data and requirements that are presented here. To carry out their activities, PSI made various assumptions and configuration descriptions; these assumptions, which were basic to this analysis, are included in this document.

The following summarizes each subsystem* requirement, associated development costs, risk levels, and schedule. The proposed milestones project system integration in the FY87 time frame, and initial operating capabilities in FY88. The hardware costs indicated for each subsystem are for evaluation/development systems and do not reflect costs of operational host computers. The software costs reflect procurement of commercially available packages.

SUBSYSTEM 1. AUTOMATED ALPHANUMERIC DATA ENTRY SUBSYSTEM:

- Subsystem for optical and magnetic input of printed material including maps, map overlays, gazetteers, printed forms, magnetic tapes, etc.

- Subsystem includes interactive OCR input station, OCR document reader, magnetic tape and disc devices, keyboard, monitor, and voice recognition.

- The primary technical risk is the development of the OCR Tablet work station. This risk is considered slight to moderate.

- The subsystem will incorporate the OCR technology currently under development at NORDA.

*Throughout this document, the terms "subsystem" or "system" are interchangeably used in this initial addressal of subsystem requirements and design considerations.

• This project is estimated to require 95 man months over 4 years.

• Hardware costs may approach $200K.

SUBSYSTEM 2. GEOGRAPHIC NAMES DATABASE SUBSYSTEM:

• Subsystem for storing and manipulating up to 60 million place names with associated positions, attributes, and relationships in a partitioned (distributed) network for both the Collateral and SCI environments.

• Subsystem is envisioned as a software element of an unspecified (but sufficiently capable) computer.

• Subsystem is envisioned to be connected to the DMA DBS Phase II operational capability and the various MC&G production subsystems (MPS) at both centers via a high-speed large bandwidth local area network (LAN).

• There is no major technical risk in the implementation of the software which lies within proven bounds. Accurately loading the database with names provides a moderate to major risk.

• For data capture, this subsystem will use primarily the AADES and the Geonames Input Station (GNIS) as an auxiliary terminal. The GNIS, in its current hardware/software configuration, will require further enhancement to include better display, menu structure and overall standard operating procedure.

• Development of "intelligent" map/scene recognition algorithms to automate geonames acquisition process from maps is required. This function could be performed by the AADES.

• Hardware and software procurement/data captures cannot be estimated without a detailed design specification. Minimum funding of $251K is projected at this time.

Subsystem 3. Advanced Symbol Subsystem W/Type Placement:

A. Advanced Symbol Processing

• This portion of the subsystem allows manipulation of names and text from the Geographic Names Database or direct input, into a format compatible with Digital Type Composition and Placement.

• This portion is so closely associated with the type composition portion, the two are often considered as one subsystem in this plan.

• This portion is a software function coresident with the Database system.

• There is little technical risk with this function. It involves melding DMA experience with the Names Input Station with state-of-the-art symbol processing and text editing.

• This project will require 30 man months over 4 years.

• Hardware and software procurement costs may exceed $50K.

B. Digital Type Composition and Placement

• This portion takes names, location and attribute information and converts it into a full page image for map makeup. This system is functionally inverse to the AADES.

• The portion includes software functions to place names with respect to feature location and other text and features existing on the map (visual optimization) and to convert these into digital image files according to appropriate font size and position on the map sheet. The hardware will employ advanced technology, e.g., laser printing to ensure high resolution, production quality, full page makeup. The hardware will also provide draft quality output on paper and digital imagery on CRT monitors for editing.

• The functional design will maintain compatibility with the conceptual design of the baseline Digital Cartographic System being developed at SPOEM to ensure integrating the ASP portion, and interfacing to the DBMS. An expert system (XPS) design, to emulate the Cartographer in the names placement/pattern recognition and optimization process, is required.

• This project is estimated to require 41 man months over 4 years.

• Hardware and software procurement costs will approach $100K.

Developing the three subsystems described above is estimated to have a total cost of about 184 man months of effort and about $603K in evaluation/systems development hardware/commercial software. This development activity is planned to be carried out over a five year period as outlined in Chapter V. Details of the final integration of these subsystems into a unified operational system/methodology will be addressed in the Implementation Plan being developed under the second phase of this planning effort.

• From a practical point of view, full automation of the symbols and names handling process will most probably not be limited by the viability and availability of hardware/software/DB technologies, but by the following two restrictions.

1. Capturing DMA very large database (with sanity checks) may well extend beyond the FY86 time frame.

2. Emulating cartographic judgment via an XPS (a challenging task in the area of Artificial Intelligence) is a major key to a fully automated system. Subsystems #1 and #3 will require transfer of algorithms/software from the technology base in machine intelligence. Realizing a fully automated system may require continued development beyond the FY86 time frame.

Chapters I and II of this document assess both management and technical requirements of the various subsystems, as presented by DMA during the course of problem(s) analysis (the principal portions of the original DMA requirement statements are reproduced in Appendix B.). Chapter III describes a general functional requirements approach to construct each subsystem. Chapter IV and Appendices A and C give a broad analysis of the geonames data sources, data relations, timing, and various strategies for data capture. Initial planning considerations for each subsystem down to the schedule and level of effort are outlined in Chapter V.

NORDA is continuing the analysis of the data and the concepts which are developed in this document, and has identified a few specific areas in this material requiring modification or further investigation. Such analysis is being carried out by NORDA under the phase two study/plan and will be incorporated into the project documentation for FY83. Examples of such areas include the following:

- DMA has extensively developed raster scan technology for automated cartography. The input systems considered in this document need to exploit more fully this "soft-copy" approach. Furthermore, any significant technical risks concerning the Automated Alphanumeric Data Entry System (AADES) are not in this area or in the area of symbol recognition (OCR technology), but are in the new area of composing the recognized symbols into "words," relating them to their respective cartographic feature (point, lineal, and areal referents), and proper database descriptions of these features. These three issues constitute areas where development work must be carried out. This "Input" topic will be considered in detail during the FY83 functional design studies.

- Combining the Advanced Symbol Processing (ASP) System and the Digital Type Composition (DTC) System needs further review and discussion with DMA. These two systems appear after future considerations to have different requirements relative to the nature of required processing. (DTC requires automated names placement in a "random" map location environment; ASP edits and formats regular, line and column, "text-like" data.) There are similarities in the two systems, however, and additional investigations are required relative to combining or co-implementing these systems.

- The database will contain basic geographic names/cartographic feature information independent of specific product scales, function, etc. The requirement to generate product specific information through data selection, formatting, font choice, etc., and how this requirement can be met has been considered initially but needs further analysis and discussion with DMA.

- Complex issues must be addressed in resolving conflicts between different input sources to the database, e.g., changes in names, sizes, etc., of cartographic "objects" for political, historical, or demographic reasons. The resolution of such issues in the current production approach involves cartographic judgment. The requirement to "protect the integrity" of the database against such "confusion errors" was initially addressed in this document but also needs further analysis. Automating such procedures could involve an extension of the technology base for MC&G application in the area of machine intelligence.

- The distinction between designing/implementing a database along with its associated database management system and the loading of the information in a selected database design on the other hand has been identified in a preliminary way. Although there are related issues in these two areas, specific planning for the loading and maintenance of the geonames database must be detailed further, with close involvement by DMA through the development of achievable options for the initial loading and maintaining of such an extensive database.

- The original NORDA Subtask Data Sheet for this project indicated a concern relative to funding for the combination of four DMA Requirements into one, while cutting the budget from the sum of the four projects individually. NORDA anticipated at that time some savings due to "technology sharings"; however, a unified interconnection between the systems will also require efforts not addressed in the individual project initiatives which led to the current subtask. Accurate cost estimates for a complex multicomponent system to be developed under this subtask can not be obtained at this study phase and must be based on the functional designs which will follow from the FY83 efforts.

- Finally, this NORDA Technical Note is to form the basis for future substantive discussions and planning with DMA Headquarters and Centers.

CONTENTS

APPENDICES:

ILLUSTRATIONS

ACKNOWLEDGMENTS

This work was sponsored by DMA under Program Element 64701B, "Advanced Type Placement and Geographic Names Database System," and was performed for NORDA Code 550, the Mapping, Charting and Geodesy Program Management Office. The DMA Program Manager was Mr. Bob Penney, and LCDR Vic Hultstrand (Code 550) was the Project Manager. Dr. Robert M. Brown (Code 371) was the principal investigator, and Dr. Al Zied (Code 371) was the co-investigator. Dr. Allen Barnes and Mr. Ed Gough both of Planning Systems, Inc., were under Contract #N00014-82-C-0726 to NORDA Code 370 for a significant portion of the work in this study.

ADVANCED TYPE PLACEMENT AND GEONAMES DATABASE

COMPREHENSIVE COORDINATION PLAN

I. DMA REQUIREMENTS

A. BACKGROUND AND INTRODUCTION

The Defense Mapping Agency (DMA) publishes maps, charts, gazetteers, politico-administrative studies, and glossaries for a variety of government and military uses. These products are provided at many different scales, for many different purposes, and contain all sorts of information pertaining to the geographic and political entities they describe. These are purely information products, selected subsets of a larger information base that, ideally, names and describes every feature in the world.

It is the job of toponymists and cartographers to collect, edit, organize, and maintain this information base, and to identify, verify, and assemble the data subsets that become cartographic products. No real base they ever assemble will name and describe every feature in the world, but in a real sense the information base they do assemble will represent the world to users of their products. How well those products describe the world is critical to the safety and effectiveness of those who depend upon them. How well the toponymists and cartographers can perform their job depends largely on the tools they have for producing and manipulating that information base.

Presently, this information base resides in printed documents, on index cards, on previously prepared maps and overlays, and occasionally on magnetic storage media used to drive phototypesetters and similar equipment. New cartographic products are assembled manually by collecting these data sources and laboriously culling, checking, and organizing the information into a new product. This information is then prepared using manual, standalone tools for phototypesetting, mechanical page makeup of overlays, and draft products. This is a labor-intensive, time-consuming process.

For many years DMA has recognized the need to automate both its procedures and its information base so that the full force of the information and experience at its disposal can be focused on the production of maps, charts, gazetteers, etc. This recognition has led the agency to study requirements, purchase equipment, contract with outside agencies for the development of advanced systems, and attempt to internally develop needed systems.

This approach has met with mixed success. In many cases technology has not been available to fully implement DMA's requirements, or the requirements have been imperfectly stated

or understood. Likewise, successful systems of the recent past have been overtaken by new technology, or were not developed with the potential to integrate separate systems into a larger, more capable production system.

This is the common experience of the users of an emerging technology and a common management problem. As the technology matures, products emerge from the commercial sector that can be exploited by many users, who thus share development costs; interface standards emerge; performance and quality increase. In general, tasks that were very hard and expensive become more tractable as the technology base expands and matures. This, in turn, causes a rising expectation among operators and dissatisfaction among managers based on a new perception of productivity. Often, the user must reassess his equipment and goals in light of this change in situation.

A systems approach to requirements and planning is essential in this reappraisal. Each functional requirement must be called out and clearly specified. The input to every system must be identified; the output needed must be detailed. The hierarchy of functions must be called out. When approached this way, as functions, interfaces, and data flow, the system emulates production flow so the product and the user are the primary considerations in the system design rather than the equipment itself. The user buys a function, not equipment.

The Mapping, Charting, and Geodesy Division of the Naval Ocean Research and Development Activity (NORDA Code 370) provides DMA with advanced systems research and development for several of their automated cartographic requirements. This plan addresses some of those requirements regarding systems for:

- Automated Entry of Alphanumeric Data,
- Geographic Names Database,
- Advanced Symbol Processing, and
- Automated Typesetting and Placement.

Each of these items represents a separate, but related requirement in the automation of the mapmaking process. To emphasize their special relationships and system potential, the implementation plans for all of these subsystems have been combined. This document represents NORDA 370's planned approach to implementing these functions for DMA.

Planning Systems, Incorporated, has worked with NORDA 371 since July 1982 under contract number N00014-82-C-0726 to assist in the technical evaluation and planning of the four advanced cartographic product subsystems listed above. Conducting this work, PSI first acquired and reviewed available program documentation from NORDA 371 to determine the stated objectives, requirements, scenarios, measurements, and plans already developed prior to PSI's contract. The primary documents available addressing DMA's stated requirements and approach are reproduced in

Appendix B of this document. NORDA subtask data sheets from 1980-1982, entitled Advanced Type Placement and Geographic Names Database, were also obtained and reviewed. From these documents PSI developed preliminary requirements and identified areas for clarification by DMA site visits.

On 13 and 14 July 1982, Dr. A. E. Barnes of PSI and Dr. Robert Brown of NORDA visited DMAHTC in Washington. On 2 and 3 August, they visited DMAAC in St. Louis. Discussions with DMA personnel at these times revealed further specific requirements and technical issues, as well as problems and approaches already being pursued within DMA. The results of these trips were reported to NORDA via memoranda dated 11 August 1982 and 17 August 1982. These trips have been followed up by Dr. Barnes with several visits to DMAHTC to clarify operational and other points.

One of the objects of these trips was the collection of pertinent documents for review and inclusion into the draft plan. Several documents of particular interest were identified and made available to PSI. One of these was the Final Report of the Automated Cartography Task Force, dated April 1982 [Ref. 6]. This document outlines DMA's internal study and recommendations for implementing Automated Cartography within rigid time constraints (12-18 months), and with minimum risk of failure. Such an approach necessarily emphasizes existing equipment and techniques at the expense of some desired capabilities. Rather than repeat this approach, PSI concentrated its efforts on developing a system plan based on stated requirements without preimposed constraints on implementation time, equipment, or manpower. Three other documents provided by DMA [Refs. 5, 8, and 10] gave descriptions of two current automation efforts: TES/EMPS and SIMS. These four documents supplied background and insight into DMA's work methods, requirements, and capabilities.

Another document that deals with requirements analysis of DMA DB Phase II was separately reviewed by NORDA Code 371 [Ref. 14].

PSI's efforts to obtain primary technical information related to Automated Cartography continued by review of the technical literature, literature search through the Defense Technical Information Center, the National Technical Information Service, and preliminary market survey by telephone. These sources of information were valuable despite the limited time available to the investigation. This effort has not been exhaustive, however, and must be continued through the early effort at system specification.

From these sources, PSI developed an internal set of system objectives, functional requirements, functional descriptions, and operational scenarios upon which to base this planning document. Furthermore, the interrelationships among these systems were analyzed to identify opportunities for joint development, cost

savings, and compatibility. From these analyses PSI has developed a preliminary Implementation Plan described in this document.

This plan is not intended as a technical specification for any of these systems. Rather, it is an implementation plan based on technical insight into the problems identified to PSI by DMA/ NORDA 371 officials. It is subject to revision as the technical details of the project are encountered and problems addressed. It does, however, provide a sound approach and a reasonable schedule for system implementation.

B. GENERAL STATEMENT OF REQUIREMENTS

The functional requirements of these four systems can be stated at several levels. In this section the general requirements are meant to be the descriptive requirements of the end user. Although this is not a design document and the detailed technical requirements necessary for a design specification are not called out, the implications of the requirements are considered to some level of technical detail for planning purposes. Such detailed requirements are presented in later sections, sometimes within the context of other discussions.

Automated Alphanumeric Data Entry System (AADES)

The stated development objectives of AADES is to investigate and analyze current computer I/O devices and requirements at DMAHTC and develop an optimum cost-effective and 99% error-free system that will convert alphanumeric data into a computer-readable form. In addition, AADES must

- support present and future database implementations by providing direct data entry from a variety of printed forms.
- require minimum operator intervention (nonlabor intensive).
- be easily edited.
- have automated error checking.
- provide clean data files compatible with Database Update Files.

Geographic Names Database System (GNDBS)

The stated development objective of GNDBS is to develop a Geographic Names Database System that will be economically responsive to the needs and requirements of both the SDA and SDS Divisions. Furthermore, the GNDBS should take full advantage of the prototype Names and Input Station that has been developed by USAETL. The applications software and file structure of the GNDBS

must be designed to enable rapid update and quick response to queries. In addition, GNDBS must:

- introduce a common, efficient data format.
- be oriented to produce primary names information and placement (for maps and charts), and gazetteers.
- provide data formats that contain all pertinent information for mapmaking and gazetteer production.
- accommodate an estimated ultimate size of 60 million geonames.

Advanced Symbol Processing (ASP) System

The stated development objective of the ASP is to provide DMAHTC with a General-Purpose Symbol Processing System suitable for use with a variety of functional operations. Currently, however, the primary function is the support of Geographic Names for gazetteers and maps. Gazetteers are currently handled using Multiset III, with manual checking of printer listings. Maps are handled by using Multiset III to prepare type stickups for manual positioning. The data comes from manually prepared forms, such as Names Data Records. This system can provide the basis for a replacement of manual portions of the present Names Placement System. The system must also:

- display proper diacritics and kerning to the operator (machine representations will be transparent to the operator at all times).
- include an electronic library of fonts, type, etc., so that display is consistent with product.
- provide interface between Geonames Database and Digital Type Composition Systems.

Digital Type Composition and Placement System

Present DMA typographical systems compose and position the characters that comprise geographic names and identifiers via keyboard cursor and aperture systems. This requirement is to address development of a more advanced system that would permit these functions to be performed more interactively and efficiently, through increased use of electronic display technology. This system must:

- compose digital typography for map and chart production.
- automatically place type in locations for final product with provision for operator intervention.

- directly create full-page type makeup required for production.

C. AUTOMATED ALPHANUMERIC DATA ENTRY SYSTEM

DMA faces more than three billion bytes of geographic names information that must be assembled, selected, digitized, edited, and entered into their products. This information is now selected manually by toponymists from archives of place names, other published maps, gazetteers, previous editions of the same product, etc.

The AADES could be used as a tool by toponymists and cartographers to enter published data into a digital database from which they can more quickly and accurately construct their products. It is a work station that includes a scanned OCR data input tablet, an OCR page reader, nine-track digital tape system, magnetic disc, voice input and keyboard data entry, a system processor, and a monitor. It will operate as a multi-user, multi-input station.

Input

The AADES should be a general-purpose input system capable of handling a wide variety of analog material. For the geoname entry problem, however, there are a number of specific inputs that it must handle, viz.

- DMA maps (or map overlays)
- Non-DMA maps
- Gazetteers
- Gazetteer tapes
- Multiset III tapes

Detailed specifications for each of these is given below.

- For DMA maps, the products of interest are Joint Operations Graphics (ground, air, radar), Topographic Line Maps, and Air Target Materials. Scales of these are given in the table on page . Maps of scales over 1:250,000 generally will not yield positional data of sufficient accuracy. City graphics are not needed for geonames, but such map products may be input to AADES in the future. Names overlays are available for DMA maps. The majority of such maps may be described as follows:

  - Size: up to 3 ft x 4 ft.

  - Fonts: 75 combinations of font and size are currently used on the Multiset III system.

  - Color: names overlays are generally black or blue.

- Orientation: most text is horizontal, though some is vertical or at an angle.

- Boundary Data: large population centers have their boundaries described by a linear segmented circuit.

- Projections: see list under "non-DMA maps" (in following paragraph).

- For non-DMA maps, no detailed description can be given. However, the AADES must have a programmable type font/ size table for character recognition to handle unusual alphabets or fonts. Scales, size, and color may vary widely. The type of map projection may be any of the common forms:

  - Mercator
  - Transverse Mercator
  - Polar
  - Lambert Conformal
  - Gnomonic

- For gazetteers, the input is text printed on a 132 character impact line printer at 11" x 14" and is photoreduced to about 8-1/2" x 11". The text consists of uppercase Latin letters with hand-entered diacritics, plus some numeric information. The data entities are given in the table on page 31.

- For gazetteer tapes, the input is 7-track digital tape in BCD format produced on a Univac computer. Data is in all upper-case Latin alphabet with no diacritics.

- For Multiset III tapes, the input is a magnetic tape copy from floppy discs. It is in gazetteer format, all upper-case characters with diacritics.

Processing Function

The AADES is required to perform the following six general functions:

- Recognize character strings from map overlays and maps.

- Identify the cartographic feature associated with character string and generate appropriate description for the feature.

  - Location types:

    point--an area represented on map by a point, circle, or some other symbol rather than an actual perimeter or feature.

area--a political, administrative, or physical feature to be represented by a perimeter.The perimeter will be a set of points representing a closed traverse.

elongated feature--river, road, etc., which are a series of line segments that do not close.

- Such locations may not appear on names overlays. If so, they must be taken from the map itself.

- Rules for associating names with map features on a busy background are not well established. May require operator.

- Recognize attributes from character font and size.

  - Must recognize not only character, but font.

  - Must recognize character size from a set of available sizes.

  - From a string of characters of a given font and size, check for internal consistency and place an attribute based on LEGEND look-up.

- From this information, form an Input Geoname File. A sample format for such a file is:

  - Header: Country, Feature Designator, Positional Accuracy, Source.

    $Record_1$--Name, Position, Feature Attribute.

    $Record_2$--Name, Position, Feature Attribute.

- Error check (preliminary) with operator interaction.

- Write output to a file (for computer) and a report (for human).

The goal of the processing approach is to minimize operator interaction with the system.

Output

- If the data is geographic names, create an Input Geoname File for the GNDBS. If the data is not geographic names, create an Analyst File for the ASP.

- Produce a written report on transaction.

Specifications

Several performance specifications must be determined for this system. The major specifications are:

Accuracy: Tolerable undetected error limits.

Throughput: Files/hour (unspecified).

Access Time: Seconds (unspecified).

Productivity: Correct Entries/man hour (unspecified).

Level of Intelligence: Operator-assisted vs. fully automated.

Human Interface: Menu-driven vs. English-like front-end query language.

D. GEOGRAPHIC NAMES DATABASE SYSTEM

The detailed requirements of the Geographic Names Database are somewhat simpler than those of the AADES. The purpose of the database is to support both gazetteer and map production from a common database.

This database would support all gazetteers. It would also support map products of scales from 1:50,000 to 1:5,000,000. Currently, there is not a DMA requirement to support city maps with this database; however, it is conceivable that such a requirement could be levied (cf. Section G).

Input

The majority of input will be through the AADES. Input may have a variety of data entities. The required entities are:

- geographic names.
- country.
- position (i.e., latitude, longitude).
- feature designator (e.g., population center, waterway).

Optional entities are:

- positional resolution (i.e., how accurate is the position).
- feature attribute (e.g., population).
- non-Romanized name.
- boundary of feature (e.g., city limits).
- reference source.

The geonames will be stored in Romanized form, with the non-Romanized name being used only for bilingual products. While gazetteers have just upper-case letters, map products require upper and lower case for geographic names. Both gazetteers and maps require a diacritics capability. There is no limit on length of a geoname. The position is needed to a resolution of one minute (01') for gazetteers. For map productions, the positions must be sufficiently accurate that the Digital Type Composition and Placement system may use the data directly for names placement; i.e., a human does not have to "adjust" all the positions. It is difficult to translate this into an absolute eror, for this depends on

- the geodetic control of the map sheet (i.e., how accurate does the map reflect the features in the area).
- the size of the entity to which the geoname refers (i.e., the size of the town or the width of a river).
- the scale of a map sheet.

The smallest circles used for population centers on maps have diameters of about 40 mils (i.e., 40 thousandths of an inch). This may be converted into map distance using the scale of the map, as in the following table:

CONVERSION OF 40 MIL CIRCLE TO SCALE DISTANCE

Map Scale	Map Distance In Seconds*	In Feet
1:250,000	08"	800
1:100,000	03"	300
1:50,000	01.6"	160

*For latitude; longitude has an additional factor of cosine of latitude.

It is seen that except for very small population centers (e.g., Vietnamese hamlets), the size of the population center exceeds the distance equivalent to a 40 mil-diameter circle on maps of these scales. In most cases, this establishes a practical limit on positional accuracy. Large towns or cities may take up an area of several inches on a map (e.g., Tucson, Arizona, on a JOG): in such cases the boundary of the city is required. The boundary data, of course, is subject to the same questions of resolution.

Output

The majority of the output from this system will go to the Advanced Symbol Processing System or the Digital Tape Composition and Placement System. This output will be selected data for gazetteers or map production. Specification of the data entities is given in Section E, under "Input" to the Advanced Symbol Processor. Since both the Advanced Symbol Processing System and the Digital Type Composition and Placement Systems are part of the overall research effort, there is no current requirement on the format of this output or the physical media on which it will be placed.

The other output requirement is for the interactive displays including the Names Input Station. (The Names Input Station as it is currently configured is a standalone system without interactive capabilities. It will be interfaced to a PDP shortly. With the implementation of a Geographic Names Database, an interactive version of this work station is desirable.) This output could consist of any data entities from the base which the toponymist may wish to examine or update.

Processing Function

The system must store the data in digital form, in a representation that is product-independent (e.g., position must be in latitude and longitude, not in terms of inches on a particular map). The system must not reject data records that lack optional data entities (e.g., a town cannot be rejected because its population was not specified in the input record). The system must have a query capability adequate to supply geonames and associated data entities for the following products:

- gazetteer production.
- map revisions.
- new map production.

The query capability must also support:

- data extraction of individual geonames for special-purpose searches (e.g., where is a particular place).
- update of data entities by toponymists (e.g., change the name of a town).
- MIS (management information statistics) queries by the database manager (e.g., what is the level of use of the base, by user).

The database management system must be able to deal with incomplete or inconsistent information. (An example of incomplete information is a query in support of a dual language map, where some of the geonames that meet the map requirements [e.g., position, population] contain only the Romanized form of the name.

An example of inconsistent information is two input sources that give the same town, but with slightly different spellings.)

The system should be capable of handling all geonames used by DMA. Based on present products and anticipated production, a capacity of 60 million geonames is a reasonable upper limit. (Street names, etc., which appear on DMA city maps, are not considered to be geonames and are not included in this figure.)

E. ADVANCED SYMBOL PROCESSING (ASP) SYSTEM

The ASP system is intended to be a general-purpose symbol processing system capable of supporting a variety of products. However, the primary requirement is for text editing and type placement functions in support of gazetteers and map products. The type placement functions overlap the Digital Type Composition and Placement task area; thus, they will be considered as part of the task area.

The original DMA requirements statement (cf. Appendix B) has been overtaken by events. The Multiset III system is now in use, providing a capability that meets some of the original ASP requirements. Products such as the Notice to Mariners are handled by current systems.

Input

Geonames inputs will come from the Geographic Names Database. Input for general symbol processing tasks will come from the AADES.

For gazetteers, the ASP is required to read data records having data entities such as geonames, feature designator, position, area, UTM grid, and JOG sheet (cf. Table 2). For maps at scale 1:50,000 or larger, the ASP is required to be able to read data records containing geoname, position, feature designator, and feature attribute. The records may also contain additional entities such as boundary and non-Romanized name. The input from AADES, of course, could be almost any textual format: tabular format, paragraph format, or merely character strings. (This would be worked out in the later stages.) However, it is assumed that AADES would create some sort of header record containing pertinent information on the format of the data file.

Output

The requirements for output depend on the product. For gazetteers, the output is a data set of formatted text, which is the equivalent of the printed gazetteer (and will be used to print the gazetteer). For map products, the output is a data file for use by the Digital Type Composition and Placement function. For other symbol processing applications, the output may be for either computer or human consumption.

Processing Function

Standard symbol processing systems abound. Like most symbol processing systems, this one must have standard editing functions:

- search,
- sort/merge,
- extract,
- modify (local and global).

It also requires formatting functions to present text in tabular formats, in paragraphs, or in specialized formats. The significant difference between ASP and the majority of symbol processing systems lies in the textual requirements.

Although the majority of the text will be Romanized, a variety of alphabets are needed to handle the non-Romanized material. Input text may be all upper case, lower case, or a combination, and will often have diacritics. Output text is similar, except that in some cases, kerning is needed. It should be noted that for some applications (e.g., dual language products), the text may be in more than one alphabet, so ASP is working with a file of interleaved alphabets.

F. DIGITAL TYPE COMPOSITION AND PLACEMENT

This task requirement was generated at DMAAC. However, certain aspects of DMAHTC's ASP requirement, which deals with type fonts, are similar to this DTC&P requirement.

A digital system is required for the preparation and placement of names on maps and charts. The system would permit existing composition and placement tasks to be done more efficiently, using interactive capabilities and displays.

Input

The input to this system would be a machine-readable file of names. This file could either be generated by the Geographic Names Database or be a separate data set (e.g., or aeronautical information). If it is a separate data set, it would have to be created on some other system, such as ASP. This data set of names must contain name, position, feature designator, and feature attribute. Other fields, such as non-Romanized name (for dual language maps) would be optional inputs.

Output

The output of the DTC&P system would be a master (e.g., a plate, a negative, or a computer file from which they could be produced) of a map overlay. This master would have names placed in appropriate positions for the overlay. The names can use both

upper- and lower-case letters, diacritics, kerning, and (for some products) non-Latin alphabets. The names are placed in various type sizes and fonts, depending upon the feature designator (e.g., city, river, airfield) and feature attribute (e.g., population).

The maps that this system would support vary greatly in scale. For DMAAC, the scales run from 1:200,000 to 1:5,000,000. For support of DMAHTC products, however, the scales run from 1:50,000 to 1:250,000 (if city maps are included, the largest scale is 1:12,500).

Function

The system must provide a capability where, with minimal operations, the cartographer may call up name, place names, and create the map overlay in some medium. The level of machine "intelligence" in performing automated positioning may require use of expert system (XPS) algorithms and technology.

G. MANAGEMENT CONSIDERATIONS

DMA must make several decisions in regard to implementing the systems. For purposes of establishing an implementation plan, choices need to be made on each of these items, but they may be easily altered. The areas are:

1. The extent of automation to be employed in AADES: Extraction of names and symbols information by essentially manual methods must be replaced as DMA moves into an all-digital production mode. The automation of information-recapture/restructuring from graphic documents to generate computer compatible records is a complex problem. The choice of how to begin this change from analog to digital handling of names and symbols will determine future progress, since some methods are extensible as future technology is developed and others are not. Three approaches to the design of AADES are presented in this document; one, based on a current DMA raster scan technology, is upwardly compatible with a second, more complex/automated system; the third, using new technology in a semiautomated way, is not upwardly compatible with the second but may be more attractive in the short term. DMA needs to set its goals relative to this issue and to determine the extent of short- and long-range automation of names and symbols.

2. Volume of nongeonames data passing through the AADES: The original requirement for AADES was for a general purpose system to convert many types of data into digital form. In the past three years, many of the data types originally requested have been addressed by other systems. At present, digitization of geographic names and their location is the primary function of AADES. Thus, the approach to AADES was determined to a large degree by the geographic names entry problem. However, AADES can

be designed with enough flexibility that it could handle new requirements in support of future systems. If DMA chooses to use AADES for a number of nongeoname systems so that support of the Geographic Names Database is no longer the primary task of AADES, then this will affect the choice of equipment.

3. Utility of voice input for AADES: Currently, DMA has a voice input system for bathymetric data. The system is operational and recognizes numerals. The system appears to be a useful component of an automated data entry system; one should note, however, that the existing systems have received poor operator acceptance to date and are not currently being used.

4. Tradeoff of time and personnel in construction of Geonames Database: There are several ways to manage the building of the database. If done manually, the effort needed to enter 60 million names and their locations into a database would be in the hundreds of man-years. One may attempt to build the database in a short period of time: this will require a large personnel allocation and some kind of automated or semiautomated approach. But it will result in a "finished" database (realizing that there will always be updates due to changes) being available as soon as possible. At the other end of the scale, one may enter data into the base only when it is needed to support a current DMA gazetteer or map effort. With this approach, the data entry becomes part of the normal DMA operation, replacing the manual names manipulation. This approach will also require some advancement in the level of automation. This is easier to manage, since it does not involve a "shock wave" of database building effort, but a product-oriented development of the database. This approach, however, would require many years to build the base to 60 million names. There is no mature technology that will completely automate the recognition of symbols from maps. NORDA's effort in automated hand-printed symbol recognition, coupled with "expert system" technology, can provide the basis for automated data capture for the late 1980s time frame.

5. Who manages the Geographic Names Database? This person would be charged with management functions associated with the base, as well as overseeing the various ADP procedures common to any large scale database. The manager would set database policy, arbitrate problems arising between different groups of database users, and have ultimate control over the contents and operations of the base. DMA must decide where in its organization the task will be assigned.

6. With acceptance of the Geographic Names Database, what should be done with the Foreign Place Names File? The current file of index cards contains bibliographic and historical notes that are of interest to the toponymist, but are not considered "output" from the file (i.e., they do not appear on maps, gazetteers, etc.). If the card file is eliminated, this data must be transferred to other media or it will be lost. A number of options have to be considered, and not all or them are exclusive.

- Keep the Foreign Place Names File on cards. From an automation viewpoint, this is probably the worst option. Keeping parallel files on computer and index cards not only requires duplicate effort, but eliminates the efficiency that was a motivating force for automation of the database. Also, with data kept in two places, updating will quickly result in difference between the files. Such inconsistency is currently a problem with gazetteers (based on the Foreign Place Names File) and maps (based on analog names files, i.e., other maps).

- Place all the notes into the digital database. Conceptually, this is the most direct approach. Unfortunately, it is not easy to implement. The notes are hand-written on the cards and are not suitable for digitizing using OCRs. Thus to enter all of these notes in digital form will be time consuming and expensive. In light of the use made of these notes, one may question the cost-benefit ratio of this option.

- Place the most essential notes into the digital database. Many of the notes deal with slight variations in spelling. One may reasonably ask if the toponymist actually needs to know that 12 reference works give a particular spelling or whether it is adequate to have just the most important reference or references for that spelling in the digital database. This would reduce the amount of information to be digitized, but data entry would still be a problem.

- Place the existing historical notes on other analog media. With the Geographic Names Database to handle the often-used data, the historical notes on the current index cards may be placed on microfiche or microfilm. This analog file is then "frozen," i.e., it is not updated. New historical or bibliographic notes would be keyed directly into the digital base by the toponymist.

This problem is somewhat similar to one of the problems that the Library of Congress had when it wished to automate its card catalog. Handwritten cards from the turn of the century often contain data or comments that do not fit the format of the modern cards (which were generated on ADP equipment). Their solution was to design the computer catalog for all essential data fields and to use the computer catalog for all incoming data. The recent analog material was converted to digital form. With each passing year the card catalog becomes less relevant, as the computer becomes the primary tool for searches.

7. Approach to Digital Type Composition and Placement: The Digital Type Composition (DTC) requirement was generated by DMAAC. It has some functions in common with the ASP requirement generated by DMAHTC. Also, current efforts by the Special Projects Office for Exploitation and Modernization (SPOEM) include a

semiautomated map makeup capability. Such Digital Cartographic Systems (DCS) could be fed geonames (and associated data) from a database through a product format generator; names could then be placed by a human using computergraphics aids or semiautomated names placement algorithms. Conceptual design of the DCS is currently underway (Kottman, SPOEM) and close coordination with SPOEM will be maintained.

DMA needs to determine how the Digital Type Composition and Placements Subsystem will relate to and interface with systems being considered by the SPOEM. Clearly, electronic, all-digital (softcopy) map makeup capability is required to allow (semi) automation of this DTC cartographic function. This concept was a key element in the original Advanced Type Placement and Geonames Database Subtask. It is also a key concept in the DCSystems. Thus, common technologies (display, storage, etc.) should be used wherever possible. At one end of the commonality spectrum, the DTC could be software packages that provide the DCS with names and symbols placement capabilities and with proper interfaces (e.g., product format generation) to the GNDBS. At the other end of the spectrum, the DTC could be a standalone hardware/software capability that would operate in parallel with DCSystems that prove other map makeup functions (e.g., compilation of roads, towns, etc., overlay superposition, cartographic review, etc.). NORDA plans to work with DMA HQ, SPOEM, and the centers to implement an optimum approach to the DTC System and other related tasks.

8. Should names data from city maps be entered into the Geographic Names database? Of DMAHTC's current map production, city maps constitute a very small percentage. It is anticipated that city map production will continue using the Multiset III equipment. At some point, however, DMAHTC may choose to automate these products using a digital system (e.g., DCS, or perhaps a modernized version of LIS) fed by digital data. Then the names on a city map would need to be entered into some digital names database. A name on a city map has associated data such as feature designator (e.g., street, square, building, bridge), position (often given in an index or glossary of names, and in a UTM or similar grid system) and a boundary (which is closed for a building, and a linear string for a street). It appears simpler to use the Geographic Names Database structure for such data than to build a duplicate database structure. Data from city maps could be stored in this database structure in a variety of ways. First, the data could simply be added to the geonames, producing a very large online database. A second approach is to put them in the same database as the geonames, but hold the data on other storage media, which would be loaded only when a city map was needed. A third approach is to build a parallel database using the architecture of the Geographic Names Database, but consisting only of data from city maps.

9. Host computers: Although there are four distinct DMA requirements, the systems have numerous interfaces. There are

advantages to being able to use different systems from the same workstation (e.g., use the Geographic Names Database to extract a data set, then use the Advanced Symbol Processing System to perform text editing). Thus, some of the systems may benefit by residing on the same physical system (i.e., computer), provided that it can support more than one of them.

10. Advanced Symbol Processing System and Multiset III: The original requirement for Advanced Symbol Processing was written before Multiset III came on-line. The Multiset III system is currently used in gazetteer production and in making type stickups. Digital type placement will supersede the type stickups, but there are a number of features in Multiset III that would be needed in the Advanced Symbol Processing System. How should the Advanced Symbol Processing System relate to Multiset III? Since Multiset III is a commercial system, questions of proprietary software, etc., cloud the issue.

Three approaches are obvious: Multiset III may be kept intact and interfaced with an Advanced Symbol Processing System, which would perform those functions not in Multiset III; an Advanced Symbol Processing System may perform all functions for some products, while other products (e.g., city maps) continue to use Multiset III; and an Advanced Symbol Processing System may be built that simply replaces the Multiset III system.

11. How long should parallel systems be maintained? Initially, when the AADES, the Geographic Names Database, etc., come on line, one would expect DMA to keep parallel systems (old and new) at least through the test and evaluation phase. With system acceptance, tasks would usually be transferred over to the new system as soon as practical. Yet, one may argue that for certain products, the old system may be cheaper: e.g., a gazetteer with less than 100 diacritics may be produced by hand annotation, or a map revision of less than one dozen names may be done by type stickups. By using such parallel systems, however, database construction may be slowed, for the new data is not being captured in digital form. Before the new systems reach their acceptance, DMA should decide how long parallel systems are to be maintained and under what conditions they should be used.

12. Tradeoffs between capabilities, risks, and cost for AADES: The area that appears to have the highest technical risk is AADES. The tradeoff between various factors may be estimated by a number of methods. These range from qualitative assessments to detailed design studies on competing approaches. Since the relative costs of these methods vary significantly, this may be regarded as one of the management considerations.

13. How many of the systems will be built? The four requirements statements of DMA involve several common themes. This allows efficiency in implementation and operation by designing the systems to work together as much as possible. However, it is possible

that not all of the four requirements will be pursued. Fiscal considerations, technological breakthroughs, etc., may change DMA intentions. Or the DMAAC requirement may proceed independently. It is DMA's prerogative to implement some, but not all of the systems.

For the purpose of this Comprehensive Coordination Plan, the following choices have been assumed:

- A semiautomated approach based on existing DMA raster scan technology will be implemented initially.
- The amount of nongeonames data passing through AADES in support of any particular database is smaller than the amount of data going to the Geographic Names Database.
- The Geographic Names Database will be built as gazetteers and maps come up for revision or as new maps are produced, rather than being built all at once.
- When the Geographic Names Database is accepted by DMA, the Foreign Place Names File will be terminated. The existing historical and bibliographic notes will be placed on analog media (e.g., microfiche) for reference purposes. The digital database will have room in the audit trail database (cf. App. C, Sect. 2, para. C) for bibliographic references. Normally a data entity would have only one reference, but multiple references could be used when necessary. Essential information from the analog file would be transferred to the digital file, and all new notes would be entered directly into the digital base.
- The Digital Type Composition and Placement System requirements of DMAAC can be met by SPOEM's systems, when interfaced to the Geographic Names Database.
- The Digital Type Composition and Placement interface between the Geographic Names Database and CAMPS will be combined with the Advanced Symbol Processing System into one piece of software.
- The computer system that hosts the Geographic Names Database will also host the combined Advanced Symbol Processing/Digital Type Composition and Placement Interface system.
- Names from city maps will not be included in the Geographic Names Database. However, the functional relationships of the database (cf. App. A) can support such data.
- The Advanced Symbol Processing System will incorporate functions and possibly components of the current Multiset III system.

- The technical risks involved in AADES are best addressed by a detailed design study of competing alternatives.
- Each of the four capabilities will be developed.

II. TECHNICAL CONSIDERATIONS

A. PROBLEM AREAS

Since this document is concerned with developing a Comprehensive Coordination Plan to develop and implement systems, it is necessary to note a number of technical issues which must be addressed in the development of the systems. The purpose of the chapter is to identify major technical issues: it is not intended to solve the problems. However, the functional description given in Chapter II is based on a number of the points raised in this chapter.

Many of these problems affect only a small percentage of the data, yet any automated system must cope with them. For example, it is not necessary that an Automated Alphanumeric Data Entry System be able to read 100% of the input; but for those items which it cannot read, it must be able to accept manual inputs.

Sections B through D deal with details associated with the Automated Alphanumeric Data Entry System. Sections E through J concentrate on aspects of the Geographic Names Database. Sections K through L deal with aspects of names placement.

B. AUTOMATED INPUT FROM EXISTING MAPS

Consider for a moment what functions an ideal automated data entry system would perform. The ideal automated data entry system would duplicate in every way the cartographer's ability to read and interpret maps and other sources of cartographic information, and automatically code that information into an easily worked database scheme. This ideal system would operate without human intervention and have an error rate comparable to the best cartographer on his best day. Such a system would have to be able to:

- Read a map
  - Recognize every character on the map
  - Recognize those characters belonging to the same word
  - Recognize those words belonging to the same name
  - Recognize the attributes contained in each typeset name
  - Establish the location or region associated with each name
  - Recognize special relationships between the words on the map, e.g., notes, instructions, hierarchies, etc.
  - Recognize all abstract entities from their printed representations
- Automatically organize the information gathered into a file for entry into a digital database.
- Report the results of the map reading and database information (e.g., Management Information Statistics).

Although progress is being made in these areas by investigators in pattern recognition, artificial intelligence, optical, and electrical and systems engineering, such a capability is not yet available. The practical questions for those attempting to implement such a system now are:

- What can we accomplish?
- What is the technical risk that we cannot accomplish our stated objectives within a reasonable time/money budget?
- Can an expandable system be built in modular form, which may take advantage of new technology as it becomes available?
- Will the envisioned product be a significant contribution to the intended user?

These questions are best answered only after detailed design studies on several competing approaches and perhaps some prototype development (cf. Section G). Without such studies the quantitative relationships between capability and cost are difficult to determine.

There are, however, some areas in which the technological risks are clear and must be noticed and discussed. Two very different approaches to map reading and their risks are considered in the remainder of this section.

Option 1: This system would be a semi-automated approach based on DMA raster scan technology. The appropriate document (e.g., names overlay) would be scanned on one of DMA's existing (or newly procured) scanners (SciTex, Broomall, etc.). The digitized raster output would then be presented to an operator in a scrolling mode on an interactive graphics system such as SciTex, Intergraph, etc. The operator would outline with a light-pen-type device the information to be extracted and associate this name or symbol with its appropriate cartographic feature. This procedure captures (encodes) complex positional information. The precision of this data is dependent only on the source document and the raster scanner. The name in the graphics presentation can then be recognized/entered by the operator; if automatic recognition software is included, the digitized names can be verified by the operator and edited if necessary. This approach can be easily updated to the more automated system of Option 2.

Option 2: This approach builds on that of Option 1. It is also based on raster scan (softcopy) digital technology. An evolution of advanced algorithms can be added to Option 1 to take over more and more of the manual/operator functions. The first upgrade would be to add automatic recognition of symbols based on

the NORDA developments on OCR symbol technology. The next major step would be limited symbol-to-cartographic feature association software; this transition could be made through a semi-automated stage in which the operator still would handle complex problems. Finally, a hierarchical search and recognition could be implemented.

The following example presents one approach to such a "fully" automated map-reading problem. In this example, the overall task is reduced to a series of matched detection problems at descending scales and by having the machine "learn" the background as it solves the map. Figure 1 is an example of how an automated map reader might approach the problem.

First, the operator inputs the raster image and codes the map legend into the machine, effectively teaching the machine how to treat what it is about to read.

The machine chooses the largest scale of a single font that it expects to read on the map, calls in the table of character attributes associated with that font, sets the resolution of the input device to an appropriate scale, and scans the entire map sheet. Markings on the map which pass the initial detection are grouped and decoded against the master table so that each mark is either interpreted as a character with a location and an orientation, or labeled as an error.

From that list the machine selects the first character and from its location proceeds to search for the next letter with a similar orientation along the word path with a similar orientation. It finds the next character and begins to build a word file. This procedure continues until all acceptable letters are entered into the word file. When the word file is complete, the attribute is assigned and the location decided.

In addition, a Background Noise Database is built, indicating those parts of the map occupied by a character which should be masked in subsequent higher resolution passes.

The next pass is selected, the map scanned, the detection blocks masked, and the process repeated.

Even this approach does not fully solve the background problem. At some scale extraneous, noncharacter background such as contour lines, city designators, roads, rivers, and airfield symbols, etc., will cause a high number of character false alarms or muddled characters. This is best overcome by using map overlays as input rather than maps so the system does not have to operate against such noise. Also, selectable color filters on the input may resolve some of these problems on the map. Otherwise, there may be some relatively high error rate at some scale that represents a performance floor to the system. Even as map overlays are used, one still has a problem of selection. Many DMA map overlays

1. Place Map

2. Code Setup

3. Select Font/Size

4. Scan for Font/Size
   (If resolution is physically changed,
   then process is simplified.)

5. Recognize all characters

   Character

   Decode
   Detect

   Location

   Orientation Vector

6. Match Character Characteristics

   From location and orientation,
   predict next character.

   Search for Characters w/Location Orientation

   Search on Orientation

   Geometrical (Transformation)

   Relational (Database)

7. Decode Feature Attributes

   Form character size, font

   Get location (very uncertain)

8. Form Geonames Database Entry

9. Form Map Background Database Entry

10. If not finished, go to 3

FIGURE 1. Map Reading System Algorithm (Fully Automated)

are composites, so there may be some "names" on an overlay that are not geonames (e.g., "under construction"). As DMA has noted, automated reading of composite overlays "could cause some difficulty" [Ref. 6, p. 3].

Option 3: Recognize selected character groups.

An alternate data input approach requires significant operator involvement in entering the data.

Rather than scanning the entire data sheet, the operator manipulates a hand-held data entry OCR device over the particular word he wants to input next. This delimits the recognition/orientation problem significantly and uses the natural ability of the operator to recognize related words, the size of the word, the orientation of the word, and its attributes. The operator can also solve the symbol-to-cartographic-feature association problem.

The hand-held scanner represents the highest technical risk, and no detailed design for such a system has been proposed. Furthermore, the image distortion introduced by manually scanning a "TV camera" over an image will significantly complicate the recognition problem. This distortion, due to "hand steadiness," is not encountered in the two approaches based on existing DMA raster scan technology. Furthermore, positional accuracy is dependent on an operational system. One should also note that this approach cannot easily be extended to a more automated approach as technology becomes available.

The operation of this system, similar to the fully automated, is shown in Figure 2.

C. AUTOMATED ERROR CORRECTION

To reap the full benefits of automated data entry, there must be a mechanism, however primitive, for automated error correction. As envisioned, this system will have three aspects:

- Error correction at entry
- Continuing maintenance of the database as a background function
- Error correction at product

Essentially, these are the three opportunities to detect, control and correct error propagation within the system.

The title, "Automated Error Correction," is at best an exaggeration and a misnomer. Error correction will always be in the hands of the toponymist or cartographer. The system does, however, attempt to monitor data entry and files for discrepancies,

1. Place Map
2. Code Setup
3. Scan Word
4. Recognize Characters in Scan
   - Character
     - Detect
     - Decode
   - Location
   - Orientation Check
5. Form Database Entry
6. Display Automated Entry for Verification
7. Edit
8. For Next Word of Same Type, Go To 3
9. For Next Type, Go To 2

FIGURE 2. Map Reading System Algorithm (Operator Controlled)

redundancies, and incompleteness. When these are detected, they are flagged for the attention of system maintenance personnel, who must resolve the problems. This aspect of the system is actually an automated data quality assurance procedure.

1. Error Correction on Entry

The data entry phase extends to both the formation of the Database Input Files from the Automated Alphanumeric Data Entry System, and the admission of the DBIF into the Geographic Names Database. This represents the best opportunity to protect the integrity of the database.

The first check on the data file is for completeness, to determine if all the fields are properly filled. Then the file is checked to determine if the field attributes are tenable, i.e., that place names conform to the rules of composition, that locations are feasible and meaningful to the internal coordinate, and that the relational fields are not self-referencing, etc. Data files with discrepancies are so flagged for further action.

Data files passing the completeness test are scrutinized for redundant information. By using automated data entry techniques, it may be expected that most new entries to a mature database will be redundant, that is, will already have an internal representation. There is no benefit to re-entering a place name if, in fact, it represents a degradation of the database. If it represents an improvement in the database, such as better positional accuracy or filling in additional data entities, then the data should be used to modify the database. Such entries must be detected, the attributes compared, and a decision made as to which entry best represents the feature. In some cases the data fields can be flagged to note origin or some other relative measure of quality to allow an automatic resolution. Otherwise, the two entries must be presented to the operator for resolution.

Those data files which are both complete and not redundant are submitted to further scrutiny. In a mature database, there are likely to be few totally new entries; thus, these must be suspect and may represent errors in the data entry equipment or procedure.

To help deal with this special problem without negating the benefits of the AADES, some automated scheme must be developed which will search for similar names nearby with similar attributes. How the concept of "similar" is to be implemented is a major artificial intelligence issue, and the primary technical difficulty of this system. It is more difficult because similarity is a linguistic, phonetic, historical, and alphabetical problem and does not submit to a single mathematical measure. Whereas a trained operator may recognize similarities based on such considerations, it is not a trivial problem to teach a computer to recognize such relationships.

Nevertheless, some attempt must be made to call up all database entries within some area (based on input source scale) with similar attributes (town, village, hamlet, or river, lake, pond) for the operator to examine and make a decision.

2. Continuing Maintenance of the Database as a Background Function

During the machine idle time, a continuing maintenance program should be carried out. This program should sequentially take each data file and submit it to all categories of correctness, including completeness, accuracy, redundancy, self-reference, readability, precision, etc. The software and development requirements of this system are small since the routines must all be implemented for other tasks. Only the background queue itself is a unique application.

Data files not passing these tests will be flagged for maintenance.

3. Error Correction at Production

At production, all files considered for inclusion into the final product will be subjected to the full battery of background maintenance functions. Those entries that pass this examination will be submitted to the mapmaker, who holds final responsibility for the accuracy and pertinence of the data.

The database functional design shown in Figure 7 uses a cartographer's feedback file to handle such errors.

D. RESOLUTION OF POSITIONAL DATA

The Automated Alphanumeric Data Entry System will be used to digitize maps, and this data will be used in Type Placement on new map products. If a new map is to be created on a larger scale than the maps that were used to get the positional data, the error may be significant. The Geographic Names Database provides a common repository for this positional data, but it will only have the resolution of the sources degraded by the digitizing process.

Functionally, there are four possible sources of error:

- The location on the old map may be wrong. This would be a map error.

- The location on the old map may have been adjusted for clarity. When there are numerous close features, they are often separated so they will be readable (e.g., consider a road, a railroad and a river that are adjacent, with a small town next to all of them).

• The process of digitizing involves some error. A measure of part of this error may be obtained by observing the repeatability of the measurement.

The first concern is to estimate the size of this problem. The problem of positional resolution shows up principally for map products of scales 1:50,000 or larger. Thus, one may look at the requirements for 1:50,000 scale maps, for the larger scales (i.e., 1:25,000 and 1:12,500) are usually utilized only for maps of cities or military installations. In 1977 the requirement for 1:50,000 scale products underwent "a major reduction" [Ref. 6, p. 2]. As of 1982 the requirement is for a total of 14,653 products (Ibid., p. 4). Of these, 2797 are current, 4558 exist but need revision, and 7298 will be new maps. By digitizing names on the existing 1:50,000 scale maps, positional data of sufficient accuracy will be obtained for many geonames. These geonames constitute about 50% of the set of geonames which will appear on the 14,653 maps at 1:50,000 scale.

The number of geonames on an average DMA map (although to be precise, there is no "average" map product) is taken as 3000 in Appendix C, but for maps at a 1:50,000 scale, this number is much too high. A figure of 2500 appears conservative, but it is more reasonable. Using this figure, one obtains a total of 1.8×10^7 geonames (or about 30% of the database), which could have a problem with positional resolution. As pointed out in Chapter I, Section D, however, the size of a population center puts practical limits on accuracy for most geonames at a scale of 1:50,000. Thus, the percentage of the database, which has such a resolution problem, will be well under 30%.

Where will the positional data be obtained for this remaining percentage of geonames? At a 1:50,000 scale, one mile is about one and one-quarter inches on a map. Thus, positional data for automated type placement for these maps must be on the order of seconds, not merely degrees and minutes (one minute is one nautical mile at the equator). Gazetteers have data to degrees and minutes. Navigation features (e.g., lighthouses) have a resolution of one-tenth, i.e., six seconds. Maps at scales smaller than 1:100,000 cannot provide positional data accurate enough for 1:50,000 scale maps, and it is debatable whether data from a 1:100,000 map would be acceptable. Therefore, these secondary sources cannot provide the required resolution, and one must use primary sources (e.g., imagery).

To summarize, for map revisions at 1:50,000 scale, the old maps must provide the positional information for the majority of the geonames. For new maps at that scale, imagery can provide the needed data.

It is obvious that a bootstrap procedure must be used. Positional data must be entered from existing map products, and as new map products are needed, the resolution must be improved

using the same data sources that DMA currently uses to solve the problem.

E. TYPES OF QUERIES FOR GEOGRAPHIC NAMES DATABASE

Ideally, one would like a Database Management System (DBMS) that could answer any unambiguous, logically well-formed English question concerned with the data items in the base, and furthermore complete the task in a miniscule amount of time. Obviously, this is impossible.

There are some questions that a human could answer using the Geographic Names Database, but which a machine could not answer due to the abstract concepts involved. An example is "List all towns in England whose name rhymes with Malmesbury." Such questions should not be asked.

There are some questions about the database that would require an extremely complicated DBMS to answer. For example, consider the query "Count all the population centers in country 'X' whose altitude above sea level (measured in meters) is greater than the square root of the population, and which are also within 150 miles of a town with more than two o's in its name." This rather ridiculous English language query may be turned into a valid logical statement using data attributes such as designation, altitude, population, latitude, and longitude items that are available in this or other databases. It also requires that the DBMS be told how to compute some functions such as great circle distance, square root, and counting letters in words. But one does not need a DBMS that could handle such a query.

Adoption of a query language puts restrictions on the queries that may be asked, and the language should be powerful enough to allow all important questions to be asked and be flexible enough to accommodate the important question of future users. This implies a query language that is either very general or is modifiable.

Besides the requirement that the DBMS be able to handle the queries, the database structure must be such that often-used or important queries may be handled in a timely, efficient manner. This problem will be considered in Section J.

The types of users of the Geographic Names Database may be grouped into five classes.

- **Toponymic Queries**

  These are concerned with attributes of a place or set of places and may result in an update of the file.

- Gazetteer Production

  These are concerned with all geonames in a country whose feature attributes meet certain criteria (e.g., size). The data required is fairly standard: although the amount of information in a gazetteer varies slightly, a typical one will contain those data items in the table below.

TYPICAL ITEMS IN GAZETTEER

DATA ITEM	CHARACTERISTICS
Name	All capitals, aliases in italics
Feature Designator	3 or 4 character abbreviations
Latitude	Degrees, minutes
Longitude	Degrees, minutes
Area	4 character designator
UTM	4 character designator
JOG Sheet	7 character designator

- Map Production

  These users are concerned with the name, the location, and enough attributes of the feature to determine type size and font, and other extended features (lineal, areal, etc.).

- Outside Queries

  These are sundry requests by non-DMA people, either directly or by request to DMA. The queries may be of any sort regarding names.

- Database Builders

  These are massive updates to the database that would occur when a set of maps is digitized, or a when a new data field is added to the database structure.

Fifteen examples of typical database transactions are given in the following discussion. Twelve of these 15 are queries, two are queries with updating, and one is mass data input. These queries were chosen to illustrate the common types of queries by various DMA users, plus some uncommon (but plausible) queries. The data items listed in the following queries are, for the most

part, self-explanatory. However, formal definitions are given in Appendix A, Section 2. In the following queries, "x" represents a country name, "y" is a number, "z" is a geoname, "e" is a distance, "f_1", etc., are feature designators (e.g., population centers), and "a" is a JOG sheet number, etc.

QUERY 1

Purpose: A typical extract from the database for a names overlay of a map of one country. Only one name is required for an item. The feature designators are of particular types (e.g., population center) and the attribute (e.g., population) must be above some threshold.

Extract: Geoname, position, feature designator and boundary.

Data Qualifications: Country is "x", feature designator is "f_1" or "f_2" or "f_3", feature attribute is greater than "y", no alias names are to be used.

Output: Create a working file which the analyst (c.f. Appendix C, Section 2, Para. C) will examine and edit using the Advanced Symbol Processing System and then pass to the Digital Type Composition and Placement System.

Special Features: Provide count of number of records in output file. Flag any names whose positional accuracy is worse than "e".

QUERY 2

Purpose: To produce a gazetteer.

Extract Data Items: Geoname, aliases (if any, feature designator, position, area designator, Universal Transverse Mercator number, and Joint Operations Graphic sheet number.

Data Qualifications: Country is "x", feature designator is "f_1" or "f_2", feature attribute is greater than "y".

Output: Place the selected data in a working file. This file will then be edited by the analyst using the Advanced Symbol Processing System. The output will then be used to produce the printed gazetteer.

Special Features: Sort geonames alphabetically.

QUERY 3

Purpose: To verify or extract data for a single population center in a known country.

Extract Data Item: Geoname, aliases, feature designator, feature attribute, position, type of Romanization used in geoname, non-Romanized name, date that geoname was entered into the base, and the source from which the geoname was obtained.

Data Qualifications: Geoname is "z", country is "y".

Output: Display on user's CRT screen.

Special Features: User's CRT must handle both Latin and non-Latin alphabets with diacritics.

QUERY 4

Purpose: To verify the spelling of a place, or to locate a place whose spelling was uncertain, in a known country or its territories.

Extract Data Items: Geoname, feature designator, and position.

Data Qualifications: Geoname is similar to "z", country is "x" or one of its territories.

Output: Display on CRT.

Special Features: Check for similar spellings of name in the database.

QUERY 5

Purpose: To locate a place whose spelling is uncertain, and whose country is also uncertain.

Extract Data Items: Geoname, country, position.

Data Qualifications: Geoname is similar to "z", country is "x" or an adjacent country.

Output: Display on CRT.

Special Features: Check for similar spellings of name in database.

QUERY 6

Purpose: In support of a very large scale map product, pull just those cities of over a million people in North America.

Extract Data Items: Geoname, position.

Data Qualifications: Geoarea is "North America," no alias names, feature designator is "population center," feature attribute is greater than 1,000,000.

Output: Place the selected data in a working file. This file will be edited later using the Advanced Symbol Processing System. After

editing, the data will serve as input to the Digital Type Composition and Placement Function.

Special Features: None.

QUERY 7

Purpose: To support a particular map product.

Extract Data Items: Geoname, position, feature designator, feature attributes.

Data Qualifications: Map is JOG "a", no alias names wanted, and either:

(a) feature designator is "f_1" and feature attribute is "y_1" or

(b) feature designator is "f_2" and feature attribute is greater than "y_2".

Output: A working file which will be edited using the Advanced Symbol Processing System and then will be used in Digital Type Composition and Placement.

Special Features: The query is a more complicated logical expression than previous queries.

QUERY 8

Purpose: To extract all items in a 2° x 2° box, except for waterways.

Extract Data Items: Geoname, feature designator, feature attribute, position, and boundary.

Data Qualifications: Position between 36°N and 34°N and between 100°E and 102°E, feature designator is not "waterway."

Output: Hard copy output on printer, for future reference.

Special Features: None.

QUERY 9

Purpose: To review all database changes made by a particular person yesterday.

Extract Data Items: All changed items.

Data Qualifications: User identity is "u", date is "t".

Output: Display on CRT.

Special Features: None.

QUERY 10

Purpose: To count the number of geonames in the database which come from a particular reference source.

Extract Data Items: No items, just a count.

Data Qualifications: Geoname reference source is "r".

Output: Display on CRT.

Special Features: Provide a count of data items.

QUERY 11

Purpose: To extract all geonames of a certain form which come from a given reference source.

Extract Data Items: Geoname, country, position.

Data Qualifications: Geoname ends in "stadt," geoname reference source is "r".

Output: Printer.

Special Features: None, but this is a query designed to tax the DBMS since it requires massive searching.

QUERY 12

Purpose: A method of Romanization of names is being abandoned. Provide the toponymists with a list of all names needing to be changed, so they can determine an efficient procedure to handle the task.

Extract Data Items: Geoname.

Data Qualifications: Country is "x", type of Romanization is "p".

Output: Printer.

Special Features: Realizing that there are exceptions to almost every rule of languages, one can use an applications program to make a first pass at converting to a new type of Romanization and have the toponymist handle the nontrivial cases.

QUERY 13

Purpose: To update a geoname due to a change in spelling, or a renaming.

Modified Data Items: Geoname.

Data Qualifications: Old geoname is "z", country is "x", feature designator is "f".

Output: Indicate on CRT that update was accepted.

Special Features: None.

QUERY 14

Purpose: To improve the accuracy of a position in support of a new, small-scale map product.

Modify: Position.

Data Qualification: Geoname is "z", country is "x", feature designator is "f".

Special Features: None.

QUERY 15

Purpose: This is not a query, but rather a batch update. Read in a magnetic tape of geoname, position, etc., obtained from digitizing maps. Set feature designator and feature attribute based on overlay, type size and font. Certain features, such as country, positional accuracy and data source, which apply to all the names from a portion of the map, may be specified once and used in all the data records. Check each incoming data record against base for duplication. If it is a duplicate, check for consistency.

Input Data Items: Geoname, position, positional accuracy, feature designator, feature attribute, country, type of Romanization, data source, etc. The amount of information depends on the particular map product used.

Data Qualifications: Data must be consistent with any data already in the base.

Output: A file of data items that conflict with data already in the base, for human resolution of the discrepancies.

Special Features: None.

TECHNICAL NOTE ON QUERY 7

When pulling up data by map sheet, it is not sufficient to test merely the latitude and longitude of the feature. Although the position (usually the center of the feature) lies only on one map sheet, the feature itself may be on more than one. An example of this is Alexandria, Egypt, which lies on two JOG sheets, although the center of the city is clearly inside the boundaries of only one of the sheets. A simple way around such problems is to pull up a slightly larger area, based on latitude and longitude of the center, and then use the boundary table and feature attribute (if it is population) to decide if the place might overlap the map sheet. A river, of course, presents more of a problem.

F. IMPACT OF LINGUISTIC PROBLEMS ON SEARCH STRATEGIES

Suppose a user is engaged in searching on names, either to verify spelling or to locate other data regarding the "place." The user may not be familiar with the standard method of spelling the name, thus, the DBMS must help the user when such help is requested. In the queries given in Section E, Queries 4 and 5 use this feature, which we denote as "similar spellings."

Types of Variants

Spelling

An inquirer may not know whether a town is spelled "McBain" (as the one in Michigan) or "McBaine" (as the one in Missouri). Spelling variants are usually language dependent.

Transliteration

An inquirer may query the town of Dunkerque under the more familiar form, "Dunkirk." If someone is working from old sources, the transliteration they use may no longer be the preferred one.

Diacritics

A user not familiar with subtleties of a language may not known whether he wants Los Banos of Los Baños.

Specialized Characters

Some languages have unusual forms of characters, such as the double "s" in German, the trailing "s" in Greek, or the "oe" in French. A user unfamiliar with a language may misread an unusual character form as a different character.

Spaces

An inquirer may not know whether a town is spelled "Bayport" (as the one in Florida) or "Bay Port" (as the one in Michigan).

Prefixes or Suffixes

Does one locate the Iraq Port at the head of the Persian Gulf by querying on "Al Basrah" or "Basrah," or by some other form?

Order of Names

When the first word in a name is actually the feature designator, as in "Foret de Woevre," users are liable to try just about any combination or order of names.

Combinations

Often several of the above types of similarities are combined.

Obviously, some of the examples above are mere linguistic variants which may be handled by a software program that knows to ignore spaces and diacritics when looking for similar names, and other elementary techniques. Some of these, however, may get quite complicated and are easier handled using alias names in the Geoname database. Where to place this dividing line between similar names and aliases should be determined during detailed design specification.

G. CHOICE OF DATABASE MANAGEMENT SYSTEM FOR GEOGRAPHIC NAMES

The Geographic Names Database may be divided into two components. The control is exercised by a DBMS, a collection of software which can handle a variety of tasks. Then there is the data itself, which is organized into some database structure. Each of these may be subdivided in numerous categories, but that will not be necessary here.

In a database such as this, there are a number of types of data (called data entities) and a number of relations between them. Appendix A discusses this in detail, and it is the first step in design of the database.

One of the tasks which must be done as part of implementation is the selection of a DBMS. There are a number of DBMS's in

existence. The selection of one (either off-the-shelf, revised, or newly constructed) depends on a number of factors. The most important factor has been traditionally the data itself (which is addressed in Appendix A). This section briefly discusses some other technical factors which must be considered when the choice is made. There are also nontechnical factors (e.g., cost, copyrights) that are not addressed here.

Basic Functions

To answer the types of queries given in Section E, the DBMS must have some rather standard capabilities. It must be able to

- Select records (i.e., data groups) based on a combination of attributes and create a subfile.
- Sort on various fields.
- Merge files (sorted or unsorted merge).
- Count the number of records having certain properties.
- Create subfiles of specific data entities from more general data structures.

Almost every DBMS performs these tasks (to some degree of generality), thus, it may appear that the choice of a DBMS was not significant. However, the Geographic Names Database is rather large; thus, both storage area and response time are significant concerns. There are databases of similar size which should be investigated as part of the DBMS selection process.

The DBMS should have several standard features. It should support multiple users in an interactive environment, as well as batch processing. It should have a single controller capable of handling multiple users in an efficient manner. (For example, if the database resides on five on-line disc packs and three users need data from three different discs, the DBMS should be able to read the three discs at once.) Backup/restart facilities must be available (cf. Sect. I). Also, the DBMS should have adequate Management Information Statistics (MIS) to monitor the types of uses made of the base and how it changes with time.

The DBMS should be easy to use. Most users will probably be at remote terminals equipped with some sort of interactive display. The system should make heavy use of menus, with tricks such as reverse video and blinking video to help the user.

Adaptability

It is true that the only thing constant is change itself. Failure to appreciate this maxim has probably been the factor

that has killed the greatest number of databases in the past several decades. Flexibility and adaptability of the DBMS are very important. The database will require a substantial amount of time and effort, and invariably, DMA operations will change with time. The DBMS must be able to support new types of queries. At some time in the future, there may be need for new types of data in the base. Placing additional types of data fields in the base will (in general) require a reorganization of the data structure. Such changes should be possible without altering any of the applications programs that use the database or the overall database management design. This may be done by making the various DBMS modules functionally independent of the data. Ideally, the data storage software is completely separate from the applications software, and both are independent of the data description software (which describes relationships among the data). Also, the database structure must be sufficiently integrated to support a wide variety of queries, yet not so rigid that additional types of data elements cannot be easily incorporated.

Synopsis

It is not obvious that a DBMS exists which has all these features as well as being acceptable on size and timing issues. However, a number of commercially available DBMSs address these concerns in some degree. Section J gives size and timing estimates and shows that it is possible to build a responsive database system. Thus, the choice of DBMS becomes a matter of trade-offs between various features among a set of DBMSs which can support a Geographic Names Database.

H. SECURITY REQUIREMENTS

The current Geonames card file does not contain any classified information. And, in general, one would expect geographic names to be unclassified. However, the aggregate of 60 million geonames might become classified since one may draw certain conclusions about DoD interests in areas from the amount of coverage they are given, or the accuracy of information in that area.

Conversely, there may be information on U.S. or Allied locations that is needed by DMA, but is not for open publication. This brings up the question of authorized access as well as the possibility of misleading information in the database, i.e., cover and deception. (A typical example is the maps of Washington, D.C., of two decades ago, which had the CIA labelled as "Federal Highway Administration.")

If the Geographic Names Database is classified, then there are a number of restrictions on the design of the system and its environment. Thus, it will be necessary to determine the security level of the database at an early point in the design.

I. DATABASE INTEGRITY

There are two aspects of the database integrity that should be considered: designing the structure so as to try to prevent problems and designing the software to handle the failures which do occur.

Elimination of Inconsistencies

In a database of this size (cf. Sect. J), inconsistencies can be very difficult to eliminate once they have entered the base. Thus, it is important to design the database to eliminate as many sources of inconsistency as possible. First, a data item should be located in just one place: this eliminates inconsistencies due to partial updating of the database. Second, if new data conflicts with old data, one should not keep both in the database. If the software cannot decide which is correct, it should flag the item for resolution by a human. Only after such resolution should the new data go into the base itself. Updates should be controlled so that while any user may modify local working files, only authorized users may change the permanent database. Any summary data (such as total number of geonames in a country) which may be stored in the database must be recalculated whenever the portion of the database upon which it is based is updated. Secondary tables used for accessing the data (such as inverted pointer lists) must be updated whenever the database is changed.

Component Failure

Any complicated system is subject to component failure. With a computerized database, there are a variety of modes of failure. Backup/restart facilities should handle power surges, hardware failures, software failures, and user errors. Most computer systems have an uninterruptable power supply (UPS) or similar equipment to handle power surges. But when a serious one occurs, the DBMS must also "get back on its feet." Hardware failures that occur during modification of the database often pose a problem since one is not sure if pointers, etc., are consistent or not. (In the old days when systems were simple, it was straightforward, though tedious, to determine what was lost in a hardware failure. With the complicated DBMSs in use today, this task may be difficult.) Software failures are not limited to the development phase. Even though a piece of software has been in use for years, it may still have a "hidden bug," which appears only when some unusual combination of data comes along. Finally, the DBMS should protect itself from the "authorized klutz," i.e., the user with an access key to modify the base who inadvertantly scratches a portion of the database.

When one nonessential component of the system goes out, the entire system should not come to a halt. All functions not requiring that component should still be able to operate. For critical components, redundant hardware and automatic switching

circuits may be considered. The computer operating system might handle several of these functions [cf. Ref. 13].

J. SIZE AND TIMING ESTIMATES FOR THE GEOGRAPHIC NAMES DATABASE

As noted in Section G, size and timing are two critical parameters for the Geographic Names Database.

Size

In regard to size, it is possible to pack the essential information on 50 million names into 1.5×10^9 bytes (8 bits = 1 byte). However, such a database cannot be accessed in any reasonable period of time. The example database given in Appendix C for timing estimates has 60 million names and requires 3.09×10^9 bytes for the various data tables plus another 1.15×10^9 bytes for the Audit Trail Database (which need not be kept on-line). These size estimates come from Appendix C, Section 2. The on-line database of 3.09×10^9 bytes is possible using a few optical discs. For a multiple user environment, however, it is perhaps more efficient to use many smaller disc packs (cf. Sect. G). During construction of the base, one would not want to use optical discs due to the frequent updating of the database. For this phase, using a number of 300 Meg discs appears to be an effective hardware solution.

If names from city maps are included in this database, the figure of 60 million names will swell to about 100 million. Since most of these additional names will have boundary data, this implies that the storage requirement of the base will double (approximately). Since such data would not be used as often as other geonames, one may wish to place data for city maps on off-line storage.

Timing Estimate

To obtain some estimate of timing for queries, a simple DBMS (called the Simple Database Manager) was designed in Appendix C. The 15 queries of Section E were then analyzed in terms of this database manager.

The timing was estimated by counting the number of accesses to the database. These were called "disc accesses" for convenience, but the database need not be restricted to residing on a disc. The conversion of database access to seconds depends on a number of factors. If a disc is the storage media, access is fast for continuous data, but random access is much slower (being limited by disc rotation speed and head movement). Consider some typical access times:

- Movable Head Discs: The head movement time for a large disc pack is about 35-40 msec.

- Fixed Head Discs: Access times for these are about 10-15 msec, but the ones currently available are far too small for the database.

- Optical Discs: Since they are still new, current random access times are worse than that of standard movable head discs.

- Solid State Semiconductors: Access times are very fast (0.3-0.4 msec), but size is limited (e.g., 40-70 Meg).

- Bubble Memory: This is now available in large (e.g., 200 Meg) chunks. Current bubble memory requires about 10-15 msec per access.

If one uses the database structure of Appendix C, and places small, heavily accessed structures on fast media (such as placing the Geoname Key Table in a bubble memory) while relegating the remaining structures to a number of slower media (e.g., 300 Meg movable head disc packs), then the average access time should be around 20 msec. It is assumed that the controller will allow all storage media to be performing I/O at once. This will allow multiple users to retrieve data from different discs without delays due to interference. The query times (baesd on 20 msec access) are listed below:

Query	Time
1	2.0 min
2	20.0 min
3	0.2 sec
4	0.8 sec
5	1.5 sec
6	10.0 sec
7	23.0 min
8	20.0 min
9	1.0 sec
10	1.0 hr
11	6.0 hr
12	30.0 min
13	0.2 sec
14	0.2 sec
15	17.0 min

The queries for which a rapid response (e.g., less than 5 sec) would be desired numbers 3, 4, 5, 13, and 14. In each case, response is, at most, one and one-half seconds.

These figures do not reflect the delays in a multiuser environment when several users are competing for access to the same data file (cf. Sect. II.F). However, if the system were to have 50 remote terminals, of which 30 were active at a given time, and 10 were doing toponymic queries, ten were doing gazetteer editing

and ten were doing cartographic names editing, the timing of queries still would be reasonable. For with such a mix of users, most of them will be engaged in quick (1/5 sec) queries or editing working files at any instant, and (probabilistically) few will be doing long queries (such as Queries 2, 7, or 8) at any instant.

In summary, the simple DBMS given in Appendix C can provide a response to queries about a particular geoname or updates to a particular geoname in, at most, a few seconds. Queries to extract large amounts of data in support of map products and gazetteers have run times of several minutes (about 10 minutes for most). Unusual types of queries (such as Queries 10 or 11) require a half-hour or more in a single-user environment and (due to the nature of these two queries) may be much slower in a multiuser environment.

Conclusions

Both database size and timing are serious concerns. But they can be handled using available hardware and common software techniques.

K. DIACRITICS, FONTS, KERNING, ETC.

The feature that most distinguishes DMA's requirement for Advanced Symbol Processing system from the standard symbol processing systems is the requirement to handle text with diacritics, etc. Similarly, the Digital Type Composition and Placement system must be able to handle a variety of type fonts and have the ability to kern characters.

In terms of products, the gazetteers involve a limited number of type fonts, and all names are in capital letters. Kerning is not needed for gazetteers. Maps, however, require a much larger number of fonts and a variety of sizes. The Multiset III system currently in use has 75 type fonts. Examination of various DMA maps shows between six and 14 fonts per map. Maps have some names in all upper-case letters and some in upper and lower case. Diacritics are needed for most, if not all, type fonts and, for typesetting purposes, one must distinguish between diacritics for upper and lower case letters. Kerning is used for some names on map products. Some maps are bilingual; thus, non-Latin characters are needed. The Multiset III system has 12 print wheels, each with a capacity of 1796 characters: this set represents the various alphabets and fonts used by DMA.

There is a variety of ways to encode diacritics, but for compatibility with other (non-DMA) systems, the standard being worked out now by a committee on diacritics should be adopted if at all possible. Kerning is now handled using special characteristics (fractional spaces): this must be integrated into the encoding system along with diacritics.

One may also have more than one diacritic attached to a character. This brings up questions of centering, etc. For gazetteers, the names are sorted alphabetically. The diacritics are used as a secondary sort. For example, if the following were place names, they would be sorted in the following order:

Aboc
Aboc
Aboc a
Abod
Abod

L. USE OF DICTIONARIES

While the Geographic Names Database has little use for a dictionary (since the entries are proper names), this is not true for an automated type placement system. A percentage of words on a map are not proper names, such as "river," "mountain," and the various material printed around the edges of the map. For this material, a computer dictionary search is helpful to identify spelling errors. The Multiset III has a dictionary feature, having Webster's and the New American Dictionary. It is limited to English text, but (except for bilingual products) legends and glossaries of DMA maps are generally in English. There are many ways to store and access such dictionaries [cf. Ref. 12].

An alternative to dictionaries is a code that checks words for plausibility using various rules of thumb. These apply to proper names as well as ordinary words. Such a technique might be considered for error control in the Automated Alphanumeric Data Entry System. These rules are language dependent. For example, in English one has "Q must be followed by U" and "I bfore E except after C... ." For Spanish names, X is not used, except for Gaudix.

A third area is the problem of correcting errors, or at least suggesting likely corrections. This is difficult and language-dependent [cf. Ref. 3], and does not appear to be warranted for a Digital Type Placement System.

III. FUNCTIONAL DESCRIPTION OF THE APPROACH

Taken together, the four systems considered in this plan will:

- Capture graphic alphanumeric data from documents, map overlays, magnetic storage media, and perhaps maps and forms.
- Encode that information in a digital format held internally in some (unspecified) processor.
- Display that information to a human operator.
- Allow the information to be edited and revised.
- Organize the information into an update file for the Geographic Names Database or other organization the operator desires.
- Provide Management Information Reports on all work session activities.
- Hold all Geographic Names data entered through the AADES, through manual entry, or Geonames Input Station.
- Provide standard database managment functions, e.g., maintaining relational references, providing means for data access.
- Support applications programs for data manipulation.
- Provide data to Advanced Symbol Processing for further editing and organization.
- Select character font and size based upon attribute and operator setup.
- Locate position for each character to be displayed automatically with operator override.
- Convert font, size, and position information into digital image representation for output.
- Prepare photo type or laser printed page.

Of course, this list is neither detailed nor exhaustive, considering the detailed processing that must be performed to achieve these tasks, but the list does describe the high level functions of the overall system without being machine or implementation specific.

To describe the system functions without considering specific implementations, Figure 3 gives a data file/process diagram

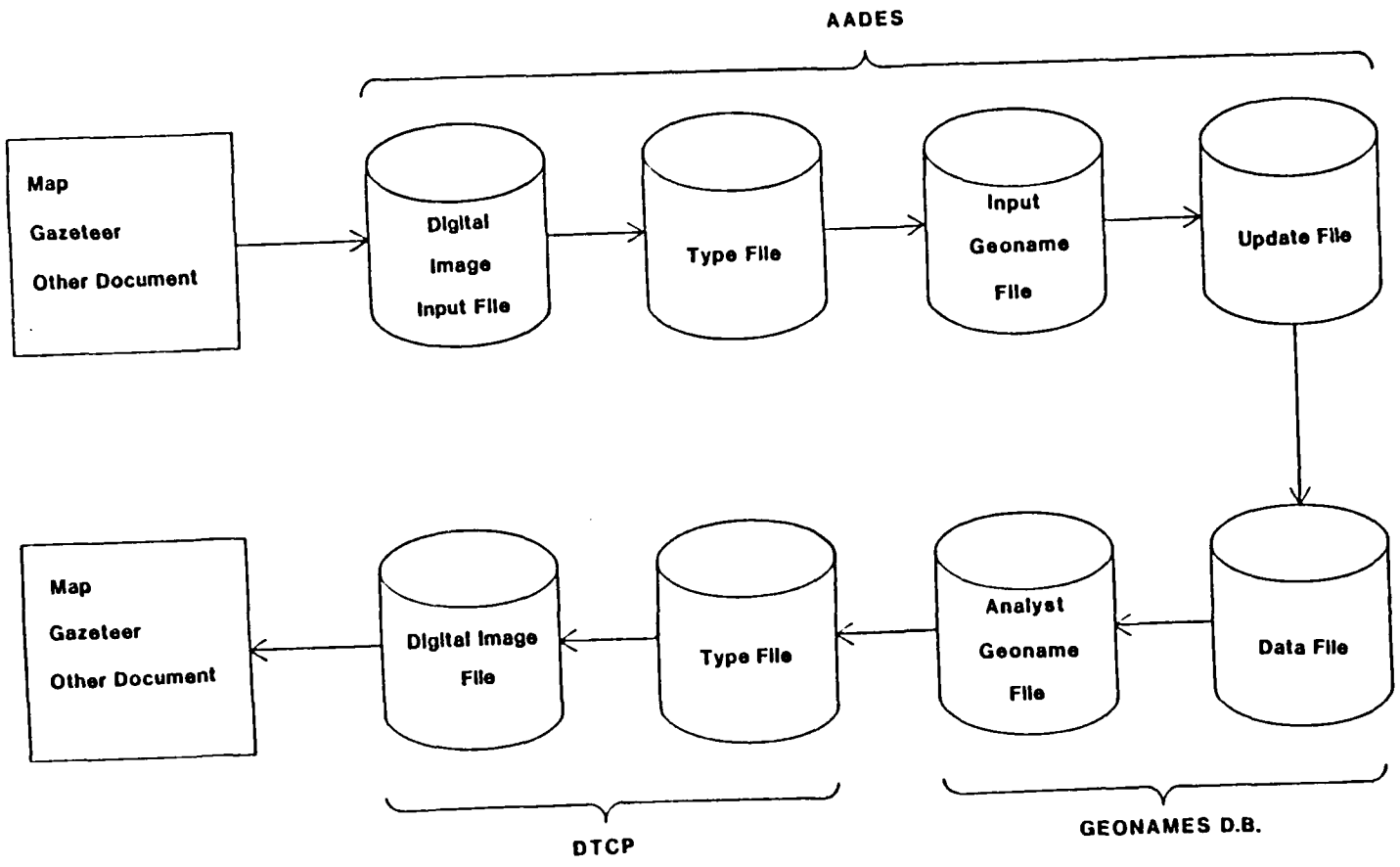


FIGURE 3. Relationships between major systems indicated by data files and processes

indicating the operation and interrelationships among the systems. This figure shows distinct data states and the transitional processes between them (self-referencing processes are not indicated), and suggests a symmetry of function that may be exploited in the system development.

Initially during data capture the graphic alphanumeric data will be converted to image data, that is, a digital representation of the graphic input. This image data will be processed through character recognition software and the image data converted into serial representations of alphanumeric characters with diacritics, font identifiers, size, location, and special symbols. This list of characters is the Type File of Figure 3. Next, these data are processed to make cartographic and semantic sense and to determine feature attributes from previously gathered information. How this work is to be partitioned between man and machine is left for later consideration, but such a process must be performed. The result is the Input Geonames File, which is a list of feature names, associated positions, and feature attributes. Presumably, there is other information associated with such a file that cannot be drawn from observing the features and characters themselves, but must be supplied at some point by the operator or some other source. This information may be stored in a file header which identifies such things as data source, data, operator, type of data, etc. When this information has been assembled, it is ready to be provided to the database for processing.

Automated data entry from printed material is a powerful text tool that should allow more flexible use than simply updating the Geographic Names Database. Programmable file output structures and reporting capabilities based on available software can also be envisioned. Likewise, the Advanced Symbol Processing can easily be configured to manipulate files such as the Input Geonames File.

Data files within the Geographic Names Database hold the information from which the data products are drawn. Applications software access and manipulate that information digitally within the analyst Geonames File, and the Advanced Symbol Processing System turns that into a list of characters with diacritics, font and size identifiers, and locations, exactly the file created on input now recreated by an inverse process. Likewise, a Digital Image File is created from the type file for use by the various output devices both for trial draft and for production via either raster or laser printer techniques.

Part of the power of this scheme is that it provides a conceptual construct for the system without being machine or implementation specific. Although each of these processes are performed and each data state realized, some of the data states may be transitory, never existing as a fully realized entity. In particular, many OCR systems capture type-file-like information

directly from the image as it is observed rather than holding the image data for later analysis, as is implied by the figure. Whether or not the realized system does this is not important, nor is it important that some elements may collect and hold image data while other system elements perform their analysis on the fly. It is only important that conceptually the processes are performed, the states achieved, and that the inverse process is used to recreate the new products.

Also, there is flexibility in this approach. Such a function can easily be envisioned implemented in several subsystems or in a single host as a single function.

It is beyond the scope of this planning document to call out the design details of such a data flow/processing scheme. In several obvious places the choice between two potentially satisfactory approaches must be studied more closely to identify the technical problems, opportunities, relative advantages, etc., of the systems. The outcome of such studies must necessarily impact the plan presented here. This impact cannot be gauged beforehand, but must be considered by responsible managers in their planning.

In each task area an approach has been chosen based on requirements and technical considerations. In this section the approach to each task is outlined along with a functional description of the system. It should be emphasized that the descriptions are purely functional and do not give, nor imply, specifications for hardware or software.

A. AUTOMATED ALPHANUMERIC DATA ENTRY SYSTEMS (AADES)

The AADES will take graphic character strings and transform them into digitally encoded character strings. These character strings will be interpreted based on operator setup instructions and the measured characteristics of the input data. The interpretations will be put into feature files. The feature files will be edited by the operator, and the edited files will generate update files for the Geonames or any other database.

Approach

The AADES system presently envisioned consists of five data input modules, special character recognition software, a system processor, magnetic storage (disc and tape), a monitor, and communication with the database processor. Details, such as whether the database processor and the AADES system processor should be separate or identical, have not been addressed. A conceptual schematic of AADES is shown in Figure 4.

The idea is to provide a data entry subsystem for each mode of data entry required. Each subsystem (which may include some virtual space in the system processor for tasks such as character recognition) will provide a character string to the processor for

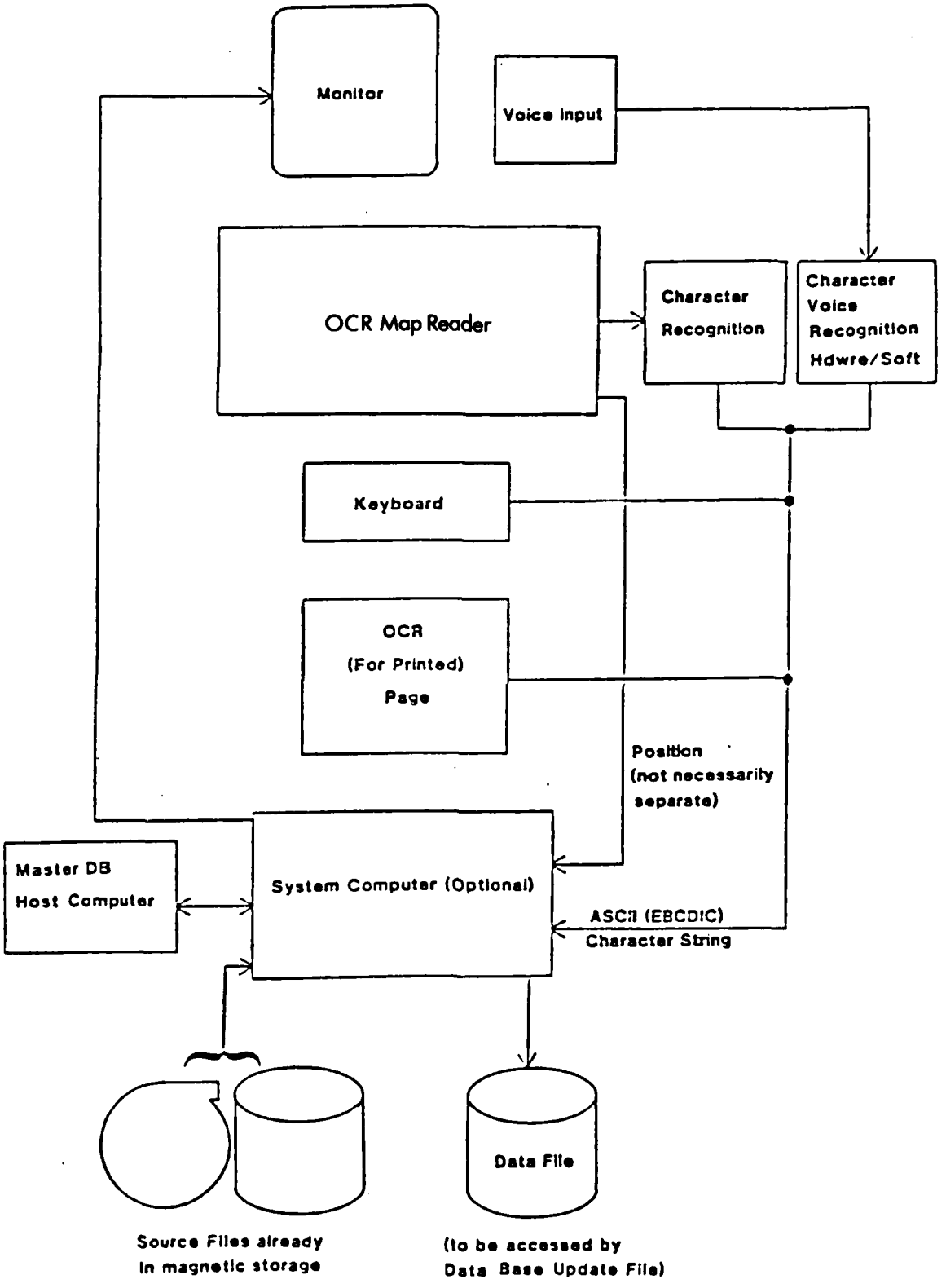


FIGURE 4. Automated Alphanumeric Data Entry Systems (AADES): conceptual schematic

building the internal type and feature files. These files are built regardless of entry method so that system software development for all processes can be standardized and thus minimized.

The output files are identical and are in the form of Geonames Database Update File.

The system is to be constructed from as many standard parts as possible, including OCR for formatted, published matter less than 11" x 14", keyboard, magnetic media, and voice entry system. The primary technical risk, and major subsystem development, is the OCR map reader, which is used to input variable formatted graphic matter, maps, overlays, etc. There are three candidate approaches to this OCR subsystem that must be considered and resolved.

Figure 5 shows the data/process flow of the three candidate systems. Option 1 represents the first approach discussed in Chapter II, Section B, wherein the entire map or other document is scanned and encoded using existing DMA raster scan technology. This data forms a Digital Map Image File that is essentially a one-dimensional contrast map of the two-dimensional original. For a given set of character parameters this list is scanned by windows of an appropriate size. When a character is recognized, its location, size, orientation, font style, etc., are indicated on the type file, and a mask file is created to avoid future muddling. Unrecognized characters are flagged for maintenance.

When the type file is created, it is subjected to interpretation in which character strings are formed for the feature file. These are created by starting at a given character location and searching along its orientation vector for similar characters, i.e., font, size. These characters are subjected to an automated "sanity check" and either admitted or rejected to the feature file.

In Option 3 of Chapter II, the operator assumes responsibility for the localization of character strings and the association of symbols with the cartographic feature involved. This approach has the operator manually scan words to be input with a hand-held scanner (as yet undeveloped) from which the character string is presented to the processor for building feature files. In this system the technical risk is a significant hardware and real-time recognition software development effort. Option 1 has the same risk relative to its ability to recognize characters of various fonts and sizes. Option 1, however, does not have to deal with distortions introduced by a hand-held scanner. It does have the added risk of automated symbol/name-to-cartographic feature identification algorithm development.

Option 2, which is also based on already-developed DMA raster scan technology, initially takes advantage of the operator judgment concerning symbol-to-cartographic feature identification

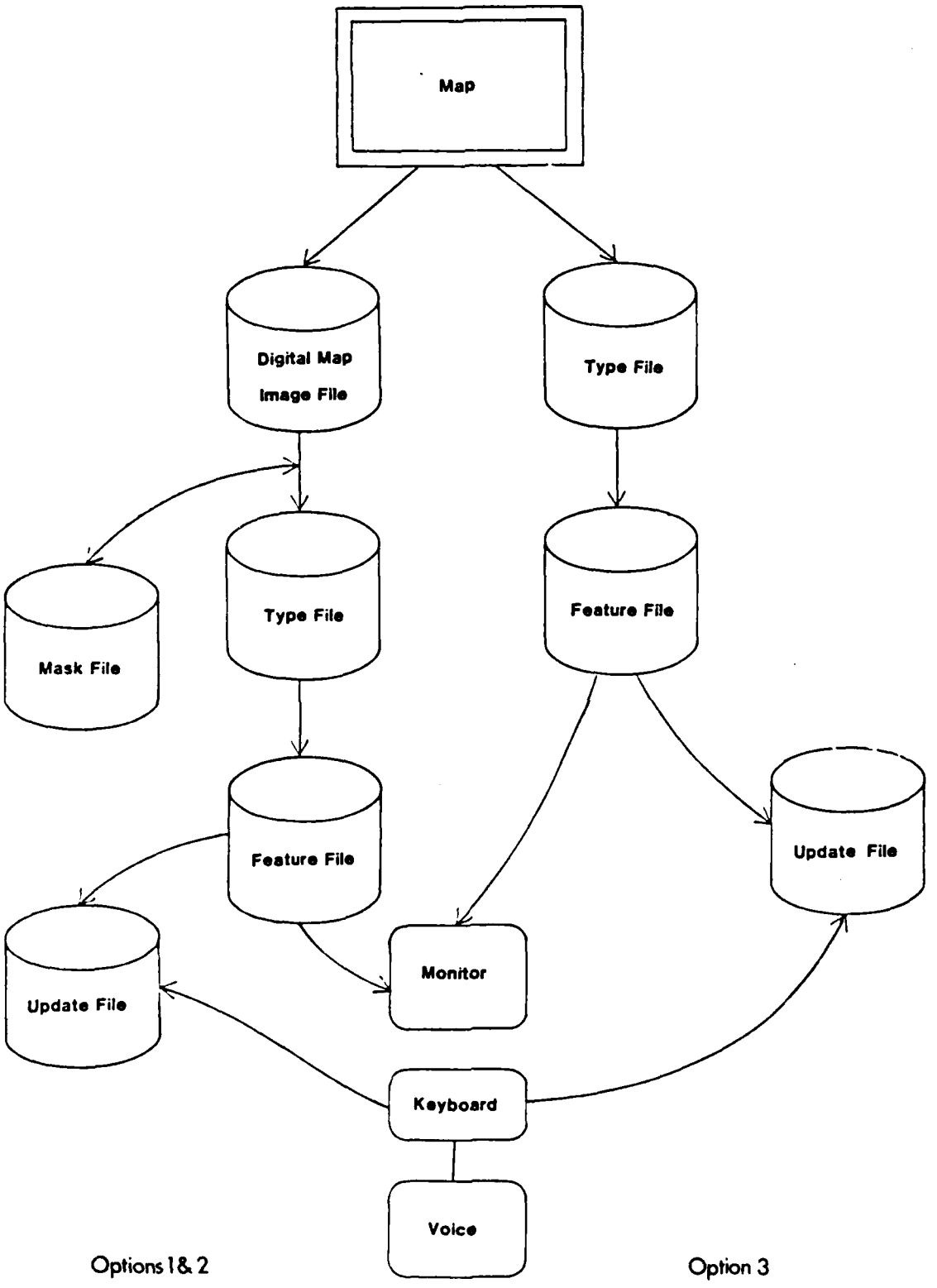


FIGURE 5. Three approaches to tablet function

but provides an explicit development route that can be exploited to minimize operator (manual) interaction in the input data stream as more advanced technology becomes available for analog/graphic information extraction, i.e., it is inherently an extensible approach. In a sense Option 2 is a hybrid or transition approach that uses an interim semi-automated technique for the solution to the symbol-to-cartographic feature recognition problem and thereby removes most of the risks involved in Option 1 relative to this complex cartographic judgment problem. Option 2 has the added important advantage of not depending on the risky development of a hand-held OCR character scanning device, since it uses mechanically stable raster scanner input systems presently in use at DMA.

The system envisioned will meet the requirements through a combination of standard and advanced subsystems. Standard subsystems will be identified, purchased, and modified as necessary. Advanced subsystems will be developed at NORDA Code 371 as required, constrained by system objectives and time requirements. NORDA Code 371 will then provide its primary technical function to develop, integrate, test, and document the AADES and deliver the product to DMA. The product will be designed to minimize operational costs measured primarily in man hours.

Input

The AADES must provide a flexible capability to read and recognize printed characters in a variety of unspecified type formats and sizes, compose character streams, recognize a limited voice vocabulary from a variety of speakers (if deemed appropriate), read computer-generated magnetic storage media in (possibly) a number of different formats, and accept keyboard entries and corrections from the operator. The capability must be flexible in the sense of easily programmed to accept characters, fonts, sizes, and formats not yet envisioned. (There is, of course, a limit to such flexibility. Part of this limit is a conscious decision based on engineering and cost decisions, and part of the limit is the unintentional consequence of selecting a particular configuration and implementation. It is hoped that the important limitations are considered and chosen in the design process based on technical and financial considerations. This approach extends the design phase, however, with attendant added cost to the system which, in limited production, will not be recovered.)

There are a number of subsystems within the AADES which fulfill specific input functions. These include the following:

- OCR Map Reader
- OCR Document Reader
- Voice Recognition System
- Digital Disc/Tape Input System
- Keyboard

Each of these subsystems has a particular input set.

The details of the OCR Map Reader implementation have not been decided, and several alternate concepts have been proposed, which are based on fundamentally different choices of future digital (softcopy) expansion capability. Basically, this subsystem provides methods of optical text entry, interpretation, and encoding from a variety of documents. Sheet size, character size, font, orientation, and background are all variable over a wide range, resulting in an envisioned system of great power and versatility, and also technical risk. No commercially available system can perform this function now.

There are, however, systems available (e.g., Kurzweil) and in development (e.g., NORDA OCR), which suggest reasonable technical approaches. Presently most commercial OCRs are designed as data entry to automated office systems. These systems, therefore, concentrate on sequential, well-ordered text entry from standard size sheets (up to 11" x 14"). One problem encountered with such systems is imbedded figures and tables that confuse the ordinary reader. Kurzweil and perhaps others have suggested that such figures may be scanned and a high-resolution image of the figure presented to an operator who, via an input device such as a light pen, selects the areas of the figure that represent text or numerals to be recognized and entered (in a manner similar to Option 2). In this way the operator limits the processors's general problem of determining all text against a confusing, nontextual background, determining orientation and word structures, and yet the operator's task of data entry is also simplified. The Kurzweil approach, however, does not appear to be sufficient to handle the random orientations and positions, extended/compressed character spacings, and kerning/diacritics that make up the bulk of information presented on maps and charts. However, this technology might form the basis for the Option 2 development effort.

When the image data is encoded, the machine may echo on the image device, superimposed on the image, the word read and its attributes for file maintenance. The operator then can immediately check the file and edit if necessary.

An extension of such a system would not only allow generalized text entry from maps, etc., but the imagery function may be constructed to allow old maps to be superimposed onto photographic imagery to check locations, changes in perimeters or features, etc., so that the total information-gathering update process for a new map construction could be automated in the AADES. This function could then also be included in the Advanced Symbol Processing/Digital Type Composition System to create from Analyst Geonames Files synthetic map imagery for editing. This approach is shown schematically in Figure 6.

A natural extension of this approach, discussed in Chapter II, Section B, is the fully automated data entry function. Functionally, the implementation is similar to that described

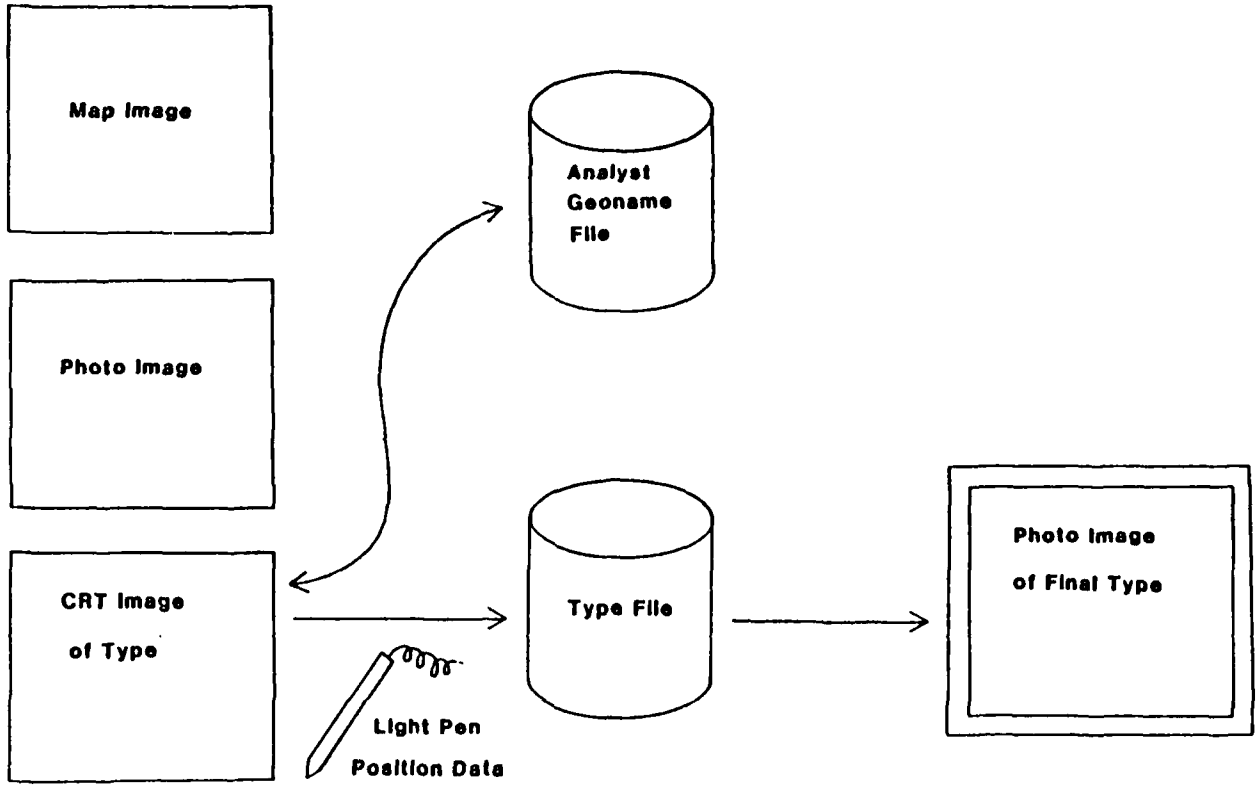


FIGURE 6. Combined AADES data capture function with AWP/DTCP output function

above, except there is minimal operator intervention, and the processor must apply advanced machine intelligence/image processing techniques to group characters in correct words and names and to associate these symbols with their cartographic feature referents. This approach can logically follow successful development of the raster scan/manual detected image identifier (Option 2) as an out-year enhancement.

The software supporting the OCR Map Reader data entry subsystem (Option 2) follow:

- Uses setup information to select data output format, enters file header information, sets status and control in the OCR function and awaits data entry.
- Acquires positional data from digitizing function and adapts to data output format.
- Accepts character string from OCR, displays string on monitor (or displays database entry on monitor), and awaits reply.
- Accepts reply and edit as necessary from keyboard entry, else continues processing.
- Provides escape functions to reconfigure setup.

Page Reader Function

The OCR Page Reader may be any commercially available OCR with programmable font capabilities. This device enables automated entry of closely printed text, e.g., gazetteers. The input page size is restricted to about 11" x 14", and there are also restrictions on format, background, and character orientation. A unit capable of detecting diacritics must be used. The OCR interprets each character read, codes the string, and supplies it to the system.

The software supporting the Page Reader follows:

- Uses setup information from the operator to select data entry and output formats, sets status and controls, and awaits data.
- Accepts formatted strings, translates into database entries, and awaits further input.
- Provides escape functions to reconfigure.

Source File Function

Source files stored on magnetic storage media represent the most straightforward data entry method. These files have been

created on other systems and represent the digital form of various gazetteers, etc. These media are mounted, the setup program function performed, data read from the media, and reconfigured into database format. Housekeeping functions are available.

Voice Data Input Function

Voice data input provides an interesting though somewhat esoteric possibility for the AADES. The technical problem of recognizing spoken characters and coding them into machine readable binary formats is conceptually similar to optical character recognition, and systems have been implemented and are available to provide limited voice functions.

The voice data input system will monitor the acoustic energy at a microphone, probably worked by the operator. The operator's spoken word will alert the voice recognition system which will attempt to recognize

- a number,
- a character, e.g., "Ay," "Bee," "See". . ., and
- a command, e.g., "Edit . . .".

This data will be used to form a character string identical to the OCR output and will be handled the same way. However, such an input method cannot easily acquire the required positional information contained on graphic (map) source documents; furthermore, it is not needed for tabular documents that can be handled by OCR Page Readers. Finally, DMA trial use of such voice entry systems has indicated limited operator acceptance of this procedure/technique. Thus, whether this input method is a basis for a cost-effective subsystem needs to be more closely examined before implementation is attempted.

In the integrated system it may be possible to use this input as a backup alternative to other systems. For example, data entered by the OCR Map Reader may be edited and corrected by voice instead of keyboard entry.

Keyboard Function

The keyboard provides the standard manual data entry method and assures the operator will be able to enter correct data or make special corrections. This function may be especially useful if the edit environment fulfills a symbol processor function.

Output

AADES system output is a data file. It will be used as input either to the Geographic Names Database or the Advanced Symbol Processing System. Formats will be compatible and need not be specified in detail at this time.

Process Setup

When the operator is ready to initiate a work session, s/he must first configure the system to his/her task. S/he will log in and initiate (or check) system functions such as time and present status. S/he will specify:

- Input Subsystem(s)
  - Configuration of Input Subsystem
  - Special character or voice programming, etc.
- Output File
- Special Processing

When the setup is satisfactory, s/he will initiate work procedures.

Reading

Of the three options discussed at the beginning of Chapter II, Section B, Option 2 appears preferable. Under this option, processing of input material proceeds as follows: Each input subsystem provides a formatted character string to the processor. This character string is echoed to the monitor immediately for oeprator verification (this may be suppressed during setup). When verified or edited via keyboard or voice entry, the string is combined with ancillary data to form the geonames data entry. The geonames data entry may then be displayed on the monitor for verification. The output file is then updated and the operator prompted for next entry.

Multiuser Operation

Multiuser system software is widely available to allow all of the input subsystems to be used by different users simultaneously.

B. GEOGRAPHIC NAMES DATABASE SYSTEM

There are two very different problems that this database system must handle. Primary, of course, is the servicing of requests. This involves queries of the type given in Chapter II, Section E. Such use involves frequent accesses, and the percentage of accesses involved in database update is small. Due to the size of the database, there is a secondary problem in building it. Database construction is a large effort and should meld smoothly with DMA procedures. Thus, a DBMS is needed that will also support large volumes of input data, much of which duplicates and possibly conflicts with existing data. Several aspects of the Geographic Names Database system are noted:

Approach

The approach taken is to first develop a detailed Functional Design Specification (FDS). This will be the basic document setting forth the specifications by which system construction, testing, and acceptance will be measured. It is related to the FDSs of the other systems since the systems interface with each other via data files. The approach proceeds with choice of a DBMS which can be extended to meet the complexities of the Geographic Names Database in a timely manner (cf. Chap. II, Sect. J), choice and acquisition of hardware to support the database and the accessing strategies (i.e., the database management system), and then database construction. Construction is by a boot-strap procedure: first, the base is expanded with data generated by the Automated Alphanumeric Data Entry System. It is assumed (cf. Chap. I, Sect. G) that data would be entered into this base as it was used by DMA staff in support of current products, and the database, in turn, would be used to support those products. With such an approach to database construction, problems involving more than cursory human action (such as improving the resolution of a position for use on a large-scale map, or finding the boundary of a city which has expanded since the former edition of the map) could be handled similarly to current DMA procedures, except that the data would be captured in digital form and entered into the database. If the database is built according to this strategy, then the completeness of the database will be very area-dependent in the early stages. As various map products are constructed or revised, the database will improve in those areas. Eventually, all areas important to DMA would be reasonably complete with high-quality data.

Database Contents and Relationships

These are given in detail in Appendix A. While the database is large, the relationships among the data entities are not complex.

Input

The system must accept input from the Automated Alphanumeric Data Entry System, users at various interactive terminals, and batch inputs of existing machine-readable data, as well as the Geonames Input Station. While the Geonames Input Station will be an important tool for the toponymist, the large data transfers will come directly from the Automated Alphanumeric Data Entry System. The current state of the data and the strategy for building the base are addressed in greater detail in Chapter IV.

While the AADES will impose a standardized format on the data fields in the Input Geonames files that it will create for this database, there will still be a large number of possible combinations of data entities (cf. Chap. I, Sect. D). The AADES will obtain the data from a variety of sources, and different sources may provide different combinations of data entities.

Functions

At this stage we are not concerned with data storage and retrieval techniques (e.g., conversion, packing, virtual memory paging, disc head movement, etc.), but with the functions that the DBMS must perform. Chapter II, Section E, discusses the general DBMS functions required, as well as the specific types of queries it must support. As noted above, the DBMS must be able to efficiently deal with the large input volume needed to build the base. It must also support a number of applications programs. These application programs would assist the users in input (e.g., transliteration of geonames obtained from a foreign map with non-Romanized names), analysis (e.g., searching the database for suspicious data entries), and output (e.g., computing Universal Transverse Mercator grid numbers from latitude-longitude pairs). If this system is not on the same machine as the Advanced Symbol Processing system, then it should also have a basic editing capability for user's working files. This editing capability is not meant to duplicate the Advanced Symbol Processing system, but rather to provide an elementary capability for touching up those outputs which otherwise would not be routed to the Advanced Symbol Processing system.

Output

The output of this system is a data set. This data set is obtained by selection of geoname "records" from the database based on user-supplied criteria, and extraction from these "records" of the desired data entities. In some cases the output may not be from the database, but instead may be based on data from the base. (This can occur when applications programs are used. The output from this system will often go to a temporary file for use by the Advanced Symbol Processing system or the Digital Type Composition and Placement system. The mechanics of this data transfer will depend on whether systems share the same hardware, and need not concern us now.) But there are additional modes of output, such as interactive display (e.g., CRT), intelligent interactive unit (e.g., Geonames Input Station), hard copy (e.g. line printer), or off-line storage media (e.g., magnetic tape).

System

The system would function as a centralized DBMS, with a single controller monitoring system used to optimize the system performance. (The physical system, of course, need not be located in one place.) It would support a number of interactive users, as well as batch users. Long requests from interactive users would be switched to batch processing. Although most of the interactive terminals would be located at DMAHTC, an interactive capability at DMAAC would be advisable. This may be difficult, however, if the Geographic Names Database is classified (cf. Chap. II). In this case, an alternative is to provide DMAAC with the Geonames

Input Station as a standalone processor, and send unclassified data between the database and the input station over a data link. The user terminals should be distributed (whenever possible) in the users' offices, preferably in a desktop configuration. For physical security, database backups should be kept in a different building from the operational database. (One may treat this as a remote site, with covered transmission lines, so that backup may proceed without physical transportation of storage media.)

Interfaces

The Geographic Names Database will get most of its data from the AADES (e.g., digitized names overlays and gazetteer tapes). The majority of the output will go to the ASP and DTC&P System (e.g., for gazetteers and map products). The database itself may be physically divided between different media (and perhaps even different locations). Conceptually, the major components of the database would be an Update Journal Database, an Audit Trail Database, a Geonames Database, and a Boundary Database. Secondary components are the Accessing Strategy Database, Analyst Geoname Files, and Cartographer's Feedback File. (The simple Database Manager described in Appendix C has 30 data structures, which implement these seven conceptual data files.) Thus, there are interfaces between different types of files within the Geographic Names Database. A schematic of these interfaces (internal and external) is shown in Figure 7. The subset of these, which will be used in building the database, is shown in Figure 8.

Formats

The Geographic Names Database system must provide users with outputs in easily readable form. Diacritics, etc., are needed on both interactive display and hard copy outputs. Representation of diacritics should be in accordance with standards (not yet released) for ease of data transfer between non-DMA computers and this database. Outputs destined for the Advanced Symbol Processing system (in support of gazetteers or map products) should be designed so that a minimum amount of reformatting is needed by the symbol processor.

C. ADVANCED SYMBOL PROCESSING (ASP) AND DIGITAL TYPE COMPOSITION AND PLACEMENT (DTC&P) SYSTEMS

As noted in Chapter I, Sections E and F, the ASP requirement was primarily a symbol processing requirement with certain aspects of type placement, while the DTC&P requirement was strictly a type placement requirement. Although these two systems are stated as separate requirements, they have been combined in the approach that was chosen (see assumptions in Chap. I, Sect. G). The decision to combine them was due to two considerations:

- At the top level, the requirements statements are similar for the two areas.

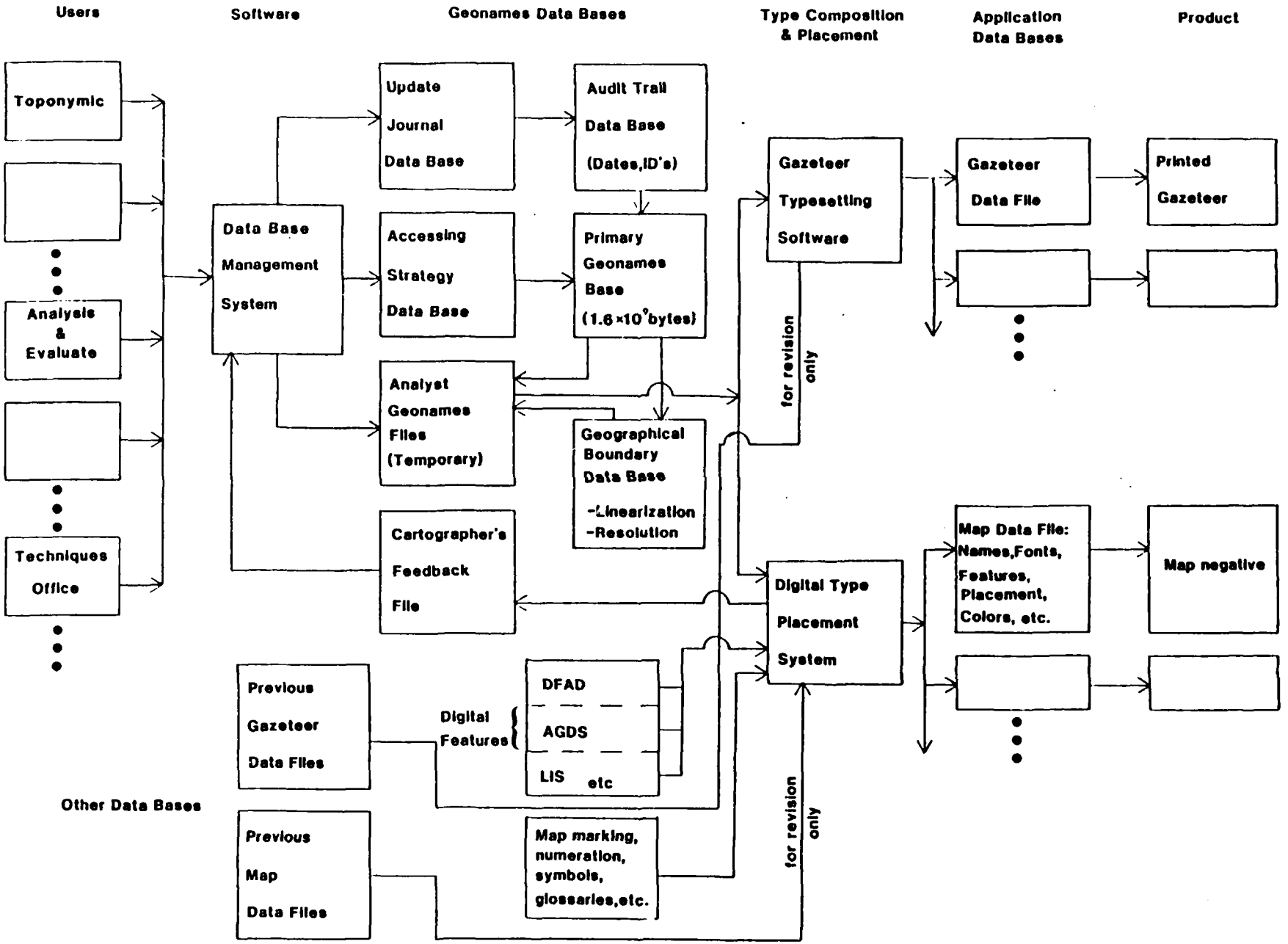


FIGURE 7. Geographic Names Data Base System (GNDBS): conceptual schematic

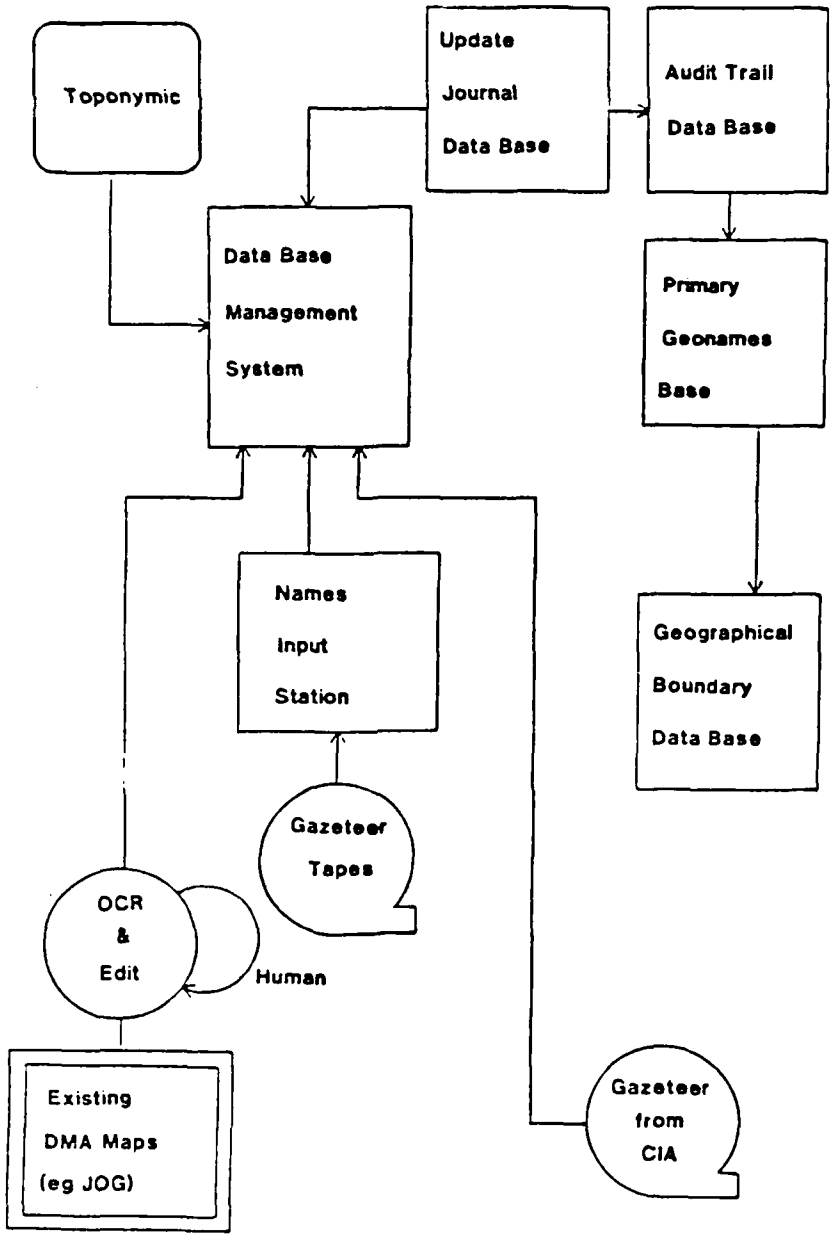


FIGURE 8. GNDBS: conceptual schematic of data base construction

- At the implementation level, both systems will make heavy use of symbol positioning algorithms and software, albeit of quite different kinds.

One of the decisions which DMA must make is how it wishes to approach DTC&P (cf. Chap. I, Sect. G), and the functional design of this system depends heavily on this decision.

The functional description presented below is based on an all-electronic information (names and symbols) handling capability. This approach, originally described in the 1980 Subtask Data Sheet, provides the cartographers and toponymists with tools that they can use effectively to: review, manipulate, select, merge/sort, and compare the map information on an all-digital (soft copy) visual display system attuned to the tasks they need to perform in exercising their specialized judgments; and to review the outputs of their tasks in context with the actual shape, geometry, fonts, diacritics, etc., which will eventually be used in the final product generated.

The key to the success of this approach is to maintain the basic MC&G data in all-digital (computer compatible) form while providing a mechanism for interactive viewing and modification through graphic/image systems appropriate to the cartographic/toponymic tasks to be performed. Thus, the map specialist can exercise his judgment and expertise to control/manage this data but is not required to "remove" it from the computer compatible domain (e.g., through overlay pullups and manuscripts, handwritten tables and reports, photographic scaling or mosaics, etc.).

The DMA SPOEM is concerned with the development of all-digital technology for use in the mid-to-late 1980s. The development efforts of the Advanced Type Placement and Geographic Names Database subtasks support this basic DMA goal. Therefore, the all-electronic mapmaker associated with the Digital Type Composition and Placement of names and symbols needs to be carefully coordinated with the SPOEM efforts on all Digital Cartographic Systems (DCS) since these systems are to provide a softcopy capability for map makeup and review in digital form for all information on the map product.

Under this subtask a number of initial discussions with the SPOEM (C. Kottman et al.) and with HQ (R. Penney) have already begun to determine areas of common support and objectives. Based on these discussions, it appears feasible for the planning purposes of this CCP document to consider the Digital Type Composition function of this subtask in the context of the Digital Cartographic Systems being considered by the SPOEM. In particular, the development results of this NORDA subtask can provide an implementation of the digital (semi) automated names and symbol composition/placement problem for both DMA HQ and SPOEM DCS efforts.

From this point of view (cf. Chap. I, Sect. G, para. 7), the two items needed to complete the names and symbols data flow are (1) a subsystem to interface the Geographic Names Database and the DCS, and (2) specialized software for automated (or semiautomated) names placement. The subsystem acts as a product specific format generator; it selects data from the Geonames Database, edits it as necessary, provides proper scaling, font selection, etc., for the specific product to be generated, makes a preliminary automated composition/placement of this names and symbol information for transfer to the DCS for final cartographic review.

Approach

The approach to building this combined system begins with a Functional Design Specification. The most crucial aspect of the FDS will be issues relating to the interfaces. Next, software is selected (many editors, sort/merge routines, text manipulators, etc., are available) which may be adapted to meet DMA needs (such as diacritics). The experience gained in the prototype Geonames Input Station may be applicable here. The system is then built and documented. Next is the integration of the system with several other systems. It is to be expected that most of the problems will surface in this step, for it is here that one can deal with real data in support of real products, and follow the data flow from the Geographic Names Database through the ASO and DTC&P system, and into the output systems. After system integration, system testing follows, culminating in delivery and acceptance.

Alternate Approaches

Although one approach was chosen, the implementation plan in Chapter V, Section C, was developed to accommodate alternate approaches. If the DCS system is not used, then software construction will be a much larger task (the schedule will permit this). If the ASP system is not put on the same machine as the Geographic Names Database, then a hardware selection and acquisition task is necessary. If the ASP system and the DTC&P systems are separated, the Functional Design Specification may be organized along parallel lines, taking advantage of the similarities in data and interfaces.

Input

The majority of input for this system will come from the Geographic Names Database, via some sort of temporary data set. However, the system must be able to accept inputs from other sources. This preserves the system's flexibility for symbol processing applications other than geonames (cf. Chap. I, Sect. E) and allows data sets created by Multiset III (for example) to be entered directly. In addition, keyboard entry may be utilized for small data sets.

Functions

The system will contain a number of standard data manipulation functions: merge files, sort on a field, change type fonts (Latin and non-Latin alphabets), search for character strings, and edit (add, change, delete) data. It would have sophisticated formatting capability to produce gazetteers or other texts. The system would have applications programs for specialized functions (e.g., in support of a given map product, for each geoname it would choose type size, font, and color based on feature designator and feature attribute). It is envisioned that this system should have many features now available in the Multiset III system (though the output media would be quite different).

Output

The primary outputs are computer files for gazetteer or map production. Formats of these files are determined by the systems which will use them. Other output media (e.g., line printer, magnetic tape) would be needed to support nongeoname symbol processing functions. These other media are also needed during construction for debugging the software.

Data Structure

Unlike the Geographic Names Database, where a complicated structure is needed to support the data and different types of data may be stored in physically separated areas, in the DTC&P system there is an advantage to keeping all data associated with a particular geoname together. This implies a variable-length record approach to formats, for the amount and type of data may vary, since a geoname may correspond to a point, line, or area feature, and the geoname with its position (e.g., center of mass) may be accompanied by boundary information. This information may be described by a simple circuit (e.g., limits of a city), or a tree structure (e.g., a river with tributaries), or another feature (e.g., a country's border may be a river). Also, for cartographic purposes one is interested in the accuracy of the positional data. For map products, all data associated with a geoname must be passed together to the cartographer.

For the ASP system, the problem of data structures is relatively easy. This becomes an aspect of software selection, for the candidate software systems generally have some sort of data structure associated with them.

System

The system may be implemented in a variety of ways. The initial recommendation is to place it on the machine that hosts the Geographic Names Database, so that users may perform either database queries or text editing functions from the same interactive terminals. These terminals, however, could be multiported so as to allow access to different systems if one host is not feasible.

IV. BUILDING THE DATABASES AND CONVERSION OF EXISTING DATA

The number of geonames required is stated as either 50 million or 60 million, depending on source. (For database design purposes, the figure 60 million was used.) Regardless of which number is the better estimate, building such a database is not a trivial task. This section considers the problem and provides an approach.

A. EXISTING DATA SOURCES

There are a number of sources of data for geographic names, from the United Nations membership list to rent-a-car maps and telephone books. While primary sources such as these may be of great interest to DMA, we will assume that it is sufficient to use secondary sources in the initial construction of the base. Five secondary sources that may be considered as data banks are discussed below.

Foreign Place Names File

This database is kept on index cards (5" x 7") at HTC, and is maintained by SDS. Its current size is 4.5 million geonames (one per card), which refer to about 2.5 million distinct places (the remaining 2 million names are aliases). The amount of information on these cards varies greatly. The index cards have a number of data fields on the top half of the card, while the bottom half is used for variant spellings, historical notes, references, etc. (continued on the back of the card or additional cards if necessary). Many of the cards have minimal information (i.e., name, location, etc., but no historical notes, etc.), while some are covered with handwriting. The geonames are located by one position, which is usually the center of the feature. The highest resolution used on these cards is one-tenth of a minute (00.1'), but many are given only to the nearest minute. There is no boundary information in this file. This card file is not in what one would say is a computer readable form.

Gazetteer Tapes

Several sources mention a computer readable file of geonames [e.g., Ref. 9, Ref. 1], and it appears that everyone is referring to essentially the same data. The size estimates vary somewhat, but 3.5 million names seem to be a typical figure. To be specific we will consider the set of approximately 155 reels of 7-track magnetic tape produced years ago on a Univac computer. The oldest ones are about 25 years old (and are still readable, we are told); the newest are seven years old (these tapes all predate Multiset III). These tapes usually have 7-10 data items, which are less data fields than the card file has. The tapes are in BCD code so that all names are Romanized in upper-case characters with no diacritics. Position resolution is one minute, but apparently some geonames have a much poorer resolution. Most, if not all, of these geonames are in the card file.

Multiset III Files

For the past seven years the Multiset system has been creating files of geonames. These are on floppy discs and are transferred to magnetic tape. The geonames are in ASCII code and have diacritics. Data on these tapes is a subset of the Foreign Place Names File: in particular, it is a subset of the data fields on the top half of the index card.

Existing Maps

Large-scale (e.g., 1:50,000) maps contain many geonames that will not be found in the Foreign Place Names File. These maps essentially form HTC's "analog database" for all geonames other than those in the card file. The number of such names is given as 50 million, but this probably includes new large-scale maps that have not yet been produced. Names overlays exist for all current DMA maps.

Names Data Records

These are internal work-sheets generated by SDAN for use in map revision. While the vast majority of these names will appear on maps, some are deleted due to lack of space on the maps. The Names Data Records are normally handwritten, then typed. While the names are therefore available for machine capture, they still must be associated with a position. Currently, this is done by indexing them to a marked-up overlay to the particular map.

Other Sources

USGS has an automated database for the United States, which contains about two million names [Ref. 11]. The Naval Oceanographic Office has about 500,000 names in machine-readable form [Encl. 6 of Ref. 9]. DIA also has a names file. Undoubtedly, other U.S. government groups have some sort of file of names and locations. These data sources were not investigated for this report, but they should be investigated prior to the actual construction of the Geographic Names Database. One may summarize these sources in the following table.

Source	Volume	Diacritics	Resolution
Foreign Place Names File	4.5×10^6	Yes	01' to 00.1'
Gazetteer Tapes	3.5×10^6	No	At best 01'
Multiset III Files	3.0×10^5	Yes	01'
Maps	50.0×10^6	Yes	Depends on scale
Names Data Records	Unknown	Yes	None

Obviously, the majority of the 60 million names must come from existing maps, for the other sources (combined) can furnish only a few million geonames.

Although names from city maps are not expected to be in the Geographic Names Database, a very brief comment on data sources is included. Multiset III floppy discs are available for those city maps which have been produced in the past few years. For city maps which have been revised in the past few years, the revisions are available on Multiset III floppy discs. Many DMA city maps have indices relating streets and points of interest to grid numbers. These indices are in the form of tabular text, which may easily be read with an OCR.

B. EXISTING OPERATIONS

Often the first solution to a future problem is seen by considering the solution used for a current problem. This section considers some of the operations which DMA currently has or will soon utilize, which may be useful in construction of the Geographic Names Database. This section is arranged by problem area.

Diacritics

Problem: From Chapter IV, it is seen that there are over three million geonames on magnetic tape (gazetteer tapes minus Multiset III), which may easily be entered into the database but lack diacritics.

Current Operations: For map revisions, the diacritics are placed on the Names Data Record and Multiset III is used to produce the names product (in this case, type stickups) with those diacritics. For gazetteers, the procedure is more complicated. The 3.5 million names on the gazetteer tapes are currently used during gazetteer revisions. These revisions are now done on Multiset III, but with the Names Input Station coming on-line, there is more latitude. The approach described to us has three pieces:

- For gazetteers with few names with diacritics (e.g., 2%), the typesetters (e.g., Multiset III) will put them on.
- For gazetteers with a reasonable percentage of names with diacritics (e.g., 10%), the toponymic branch will add diacritics using the Names Input Station.
- For gazetteers with a lot of names with diacritics (e.g., 40% or more), they will scan the old gazetteer with an OCR.

The gazetteer of Syria was recently produced by scanning the old gazetteer (about 80,000 names) with an OCR to form a machine-readable base of names with diacritics.

Approach: With these tools available to handle diacritics, the gazetteer names should not be a problem.

New Names

Problem: Of the 60 million names expected to be in the database, some of them are not on current DMA maps or data files. These residual names may not be available on a source that is easily converted to machine input.

Current Operations: When a new map product at a large scale (e.g., 1:50,000) is created for an area, there is normally a requirement for more names than are available on the existing smaller scale maps of that area. These new maps are normally based on imagery, and the names are obtained from primary sources. These names are placed on Names Data Records.

Approach: The current method of Names Data Records and marked-up overlays may be functionally adapted to an automated system for data capture. The important element is to capture the data in digital form at the earliest practical step. Digitizing the overlay is one option.

Digitization

Problem: The Automated Alphanumeric Data Entry System will involve a lot of man-machine interface.

Current Operations: Much effort is currently expended in digitization of feature data at DMA.

Approach: It is probable that knowledge gained from operating current equipment (e.g., human factors considerations) will be of use in designing the Automated Alphanumeric Data Entry System.

C. DATA CAPTURE TECHNIQUES

Filling the envisioned database with its estimated 60 million different place names, assuring its collective accuracy, and building the appropriate relational tables is no trivial task. Although filling the database is not considered to be one of NORDA's tasks in the development program, the techniques available to fill the database are an immediate and direct concern. First, techniques must be envisioned which make data entry tractable and economical. The techniques for assuring that newly entered data improve rather than degrade the database must be designed and implemented. The latter techniques are discussed in Sections D and E. This section concerns techniques for capturing the data.

There are four primary data capture modes for the GNDBS through the AADES:

- OCR Map Reader--maps, map overlays, unformatted characters.
- OCR Page Reader--printed material.
- Digital Files--magnetic tapes and disc, and video disc.
- Keyboard--for information not directly readable by above means.

Data will be in a form appropriate for one of these systems and will usually come from one of the following sources:

- Multiset III--magnetic tapes for typesetting.
- Gazetteer Tapes--magnetic tape expressions of gazetteers.
- DMA Map Overlays--printed characters on clean 2-D background.
- Non-DMA Maps--uncontrolled printed characters on cluttered 2-D background.
- Gazetteers--printed, formatted tables of geo-names information.
- Miscellaneous Printed Material--printed forms, tables, text.
- Handwritten Text--in some advanced version of AADES there may be some provision for the hand-written text entry; initially, keyboard entry will be used.

Some of the available data is already configured in a machine-readable format, e.g., Multiset III data. This information has already been typeset for existing or in-progress products, and the typesetting instructions have been stored on digital tape. To recapture this data, routines must be written to read the tapes, interpret the status of the typeset characters, e.g., font, size, etc., express certain characteristics such as features, city size, country name, etc., and create update feature files for the DBS. Unfortunately, the Multiset III files will not typically be configured with the information needed to create the update file. Place location, exact size, exact feature type, and all relational information will not be available in the machine. Specialized operator editing will probably be required but should be kept to a minimum, if possible.

Similarly, tapes generated to produce gazetteers are available for data entry. These tapes contain from seven to ten data

items, most of which amount to data entries. The gazetteer tapes do not contain diacritics, however, and must be run through the Names Input Station to add these. Because of the added labor this function requires, gazetteers with many diacritics should probably be input via OCR. Reading these tapes will require additional separate processing software. There would be a significant effort to visually read and enter (via the keyboard) the historical notes on the lower half of the Foreign Place Names index cards. Fortunately, DMA indicates that the information on the lower half of the cards would not often be needed in an automated names and symbols handling system.

There are a variety of data fields available in the Geographic Names Database. How many can be filled from a source?

- Gazetteer tapes have seven to ten items, most of which convert directly to data entities.

- DMA maps give name and position directly, but also give feature designator (e.g., town, river, etc.) and feature attribute (e.g., population category) by type size, font, and color. They may also supply boundary data (e.g., city limits, province boundaries). Relations such as "inclusion" (cf. App. A) are visually apparent, but are not easy to reduce to digital form.

- Positional resolution (i.e., accuracy and reliability) can be deduced for DMA products based on a knowledge of how the map was made. For non-DMA maps, however, one may not know the standards to which it was made; thus, positional resolution must be estimated based on map scale, accuracy of known positions, etc.

The philosophy underlying the data input is to accept whatever data files are available, "fill in" as many data entities as possible in the base from them, and leave the other entities blank.

In practice, this could be done by having the operator enter information into a header record, which would apply to all records following (until the next header). For example, if the operator is working on a names overlay of a map of Germany, the header may indicate that all names following it are population centers in the state of Bayern in the F.R.G., and that the population category may be deduced from type size and font using some internally stored table.

D. SANITY CHECKS ON INCOMING DATA

The technical aspects of error correction were addressed in Chapter II and will not be repeated here. Rather, this section identifies some specific tests that may be performed. Basically, there are three places to check data:

- Operator check at time of capture.
- Software check at time data moves from Update Journal Database into Geographic Names Database proper.
- Software checks on the Geographic Names Database as a whole.

As a minimum the software should be capable of performing the following data checks:

- Is the value within its legal range (i.e., latitude between 90° S and 90° N, population less than 20 million, etc.)?
- Is the position within the country?
- If there is an inclusion relation, is the position within the area?
- Is the boundary size reasonable (e.g., less than 50-mile perimeter for a city, less than 1000 miles for a state?
- Is the spelling consistent with any rules known for that language?
- If two places in a country have the same name, determine if the latitudes or longitudes differ by some elementary amount (e.g., 10'). If so, it may be the same place, with a typo in some of the coordinates.
- If two places have almost the same position (e.g., 00.1' separation), are they really distinct (i.e., one has a bad position) or are they aliases?
- If two places in a country have almost the same spelling (where "almost the same" depends on linguistics; e.g., "ph" and "f" are linguistically close), and are reasonably close (e.g., less than 60 miles), they may refer to the same place.

E. COMPARISON OF DATA FROM DIFFERENT SOURCES AND DATA SELECTION

Most of the 60 million names will come from digitizing maps. Thus, one would expect that there will be duplication of names from overlapping map sheets. For example, when two positions are obtained from digitizing a single place on map sheets*, the

*Because the earth is not flat, rectangular map sheets at a given scale (e.g., JOGs) have some overlap. With different map scales, overlap becomes a major consideration.

coordinates will not agree exactly. The computer must first decide whether the coordinates are "close enough," or whether the first one is in error*. If they are close enough, which should be entered into the base**?

In a database this size, the computer must resolve as many inconsistencies as possible. Consider the following scenario: one-quarter of the names in the database are entered twice, and 1/20 of these have latitude/longitude which are not "close enough." That is still nearly one million conflicts on position alone, and it becomes a significant task to resolve them by hand. Interactive graphics aids are useful for this task.

Another problem is feature designation: What one map calls a hill, another map may call a mountain. Thus, data going into the base via OCR must go through a "translation table."

Deliberate errors are a constant problem. If a road, railroad, and river run parallel (a common occurrence in rugged terrain), the cartographer will separate them on the map so they can be distinguished. If a town is on that river, has the road passing through it, and is serviced by the railroad, how is the accuracy of its position on the map affected? Such situations should be noted by the operator when digitizing the input map.

*That is, if one draws error ellipses around each point, do they overlap?

**Normally, the one with the best positional accuracy, which is often the one from the map with the largest scale.

V. SUGGESTED IMPLEMENTATION PLAN AND SCHEDULE

In Chapter I, Section A, the four requirement areas were described as forming a symmetric sequence of related operations. This preliminary implementation plan is developed to allow each function to take advantage of the other functions. It is a basic premise of this implementation plan that all four DMA requirement areas will be developed.

One of the goals in the functional descriptions of Chapter III was to keep redundancy at a minimum--using redundancy only to eliminate potential bottlenecks. With such an approach, one has the interfaces:

- The AADES feeds data to the Geographic Names Database and the Advanced Symbol Processing System.
- The Geographic Names Database feeds data to both the Advanced Symbol Processing and the Digital Type Composition and Placement.
- The Advanced Symbol Processing component may feed data to the Digital Type Composition and Placement.

This design philosophy has been carried over to the schedule. The four functions (three systems under the assumptions made in Chap. I, Sect. G) all begin about the same time and reach system integration at the same point. Hence, one may use one system to aid in the testing of another. For example, the Geographic Names Database may execute a query and create an Analyst Geoname File to pass data to the Advanced Symbol Processing System, which then edits it. Thus, the Geographic Names Database is supplying data to the ASP to test its sort, merge, and editing functions, while ASP's text editing capabilities can be of great assistance in checking the output of voluminous Geographic Names Database queries.

Each of the three systems (AADES, GNDB, and ASP/DTC&P) has been laid out as a sequence of tasks (some having subtasks). The tasks are developed [illegible]hat there is some parallel in activities and terminology be[illegible] the three systems.

A. TASKS AND SCHEDULE FOR AADES DEVELOPMENT

The AADES is an integrated combination of several hardware subsystems and software functions. Some of these subsystems are available off-the-shelf, requiring minimal effort to procure, install, and test. Other off-the-shelf items represent sophisticated application technologies and must be carefully monitored through specification, procurement, installation, and testing. The OCR subsystem, on the other hand, is a full-scale development effort in its own right, requiring full management and engineering attention throughout its development. The system integration

phase will also represent a substantial engineering and management effort. Because the tasks involved in bringing each subsystem together into an integrated system are similarly labeled (but require vastly different levels of effort), the subsystems have been combined in the following schedule. The 14 tasks for AADES are discussed below.

1. Performance Requirements Document: 2 Months

Although this document brings together the fundamental requirements for the AADES, this effort will establish in writing a firm, detailed, specific set of performance requirements from which a design specification can be drawn. Such a set of requirements will call out in detail:

- The specific set of input characters, tape and disc formats, possible phased implementation, etc.
- Limits of operator involvement and human factors.
- Limits of input and processing time.
- Error rates.
- Level of automation or semiautomation.
- Specific output forms (from Sect. B, Task 1).

Producing this document may require up to two months.

2. OCR Map Reader Design Study: 6 Months

There are several outstanding issues regarding the design and implementation approach to the general automatic input of information on maps. To resolve these, a design study between two or more candidate approaches should begin immediately after the requirements have been finalized. This study will propose alternate approaches and design details for automated or semiautomated map reading and interpretation. Measures developed in the performance requirements task (1) will be weighted and combined to form a single cost function that includes factors for risk, flexibility, cost, development time, etc. This cost function will be used to evaluate candidate systems. Additionally, the study will examine the commercial market closely to exploit potential advances there.

When approved, the results of the OCR Map Reader study will be incorporated into the Functional Design Specification.

3. Functional Design Specification: 4 Months

The FDS represents the second level of design details. This document will call out:

- Functional Requirements,
- Input,
- Process,
- Output,
- Subsystem Specification,
- Hardware Specification,
- Interface Specification,
- Integration Plan for AADES, and
- Test Plan.

When approved, this document will be the authoritative documentation on system performance and function until acceptance by DMA of the finished system.

4. Hardware Selection: 1 Month

All hardware will be chosen based upon requirements called out in the design specification.

5. Hardware Procurement: 10 Months

All selected hardware will be procured.

6. Hardware Development/Extensions: 18-24 Months

At this stage of planning, this task is the most ill-defined and difficult to accurately estimate. Without the Design Study of Task 2, the details of this task cannot be known and cannot be considered firm.

In this task the hardware which must be developed and assembled will be, with specific design functions, tested and refined as the process continues.

7. Software Selection: 2 Months

Software, which can be procured commercially without development, will be identified and selected.

8. Software Procurement: 10 Months

Selected software will be procured.

9. Software Development: 15-18 Months

All software called for in the design specification, but not procured, will be developed.

10. Subsystem Integration and Testing: 7 Months

Each subsystem will be configured with hardware and software for testing and refinement.

11. System Integration: 7 Months

Each working subsystem will be integrated into the final AADES.

12. System Testing: 2 Months

When the system integration is complete so that there is an apparently working system, AADES will be tested extensively before delivery. This testing is for NORDA, but DMA personnel should monitor these tests to smooth delivery and acceptance.

13. Delivery: 3-4 Months

The AADES will be delivered to DMA for final testing and installation.

14. Documentation: 12 Months

Full documentation for the AADES will be provided, including

- User documentation.
- System documentation.
  - Hardware engineering documentation
  - Software documentation
  - Troubleshooting documentation
- Final report on development.

Other project documentation will include

- Performance requirements (Task 1)
- OCR (Task 2)
- FDS (Task 3)

SCHEDULE TASK/SUBTASK		START (MONTH)	END (MONTH)
1.	Performance Requirements	0	2
2.	OCR Map Reader Design Study	6	12
3.	Functional Design Specification	3	7
4.	Hardware Selection	9	10
5.	Hardware Procurement	9	19
6.	Hardware Development	13	41
7.	Software Selection	9	11
8.	Software Procurement	9	19
9.	Software Development	19	35
10.	Subsystem Integration and Testing	28	35
11.	System Integration	35	42
12.	System Testing	42	44
13.	Delivery and Acceptance	44	48
14.	Documentation	36	48

B. TASKS AND SCHEDULE FOR GEOGRAPHIC NAMES DATABASE

The Geographic Names Database is a combination of a large database and a Database Management System (DBMS). The system requires high speed access to large storage devices, but existing commercial hardware is capable of providing this. The overall task is low risk. Filling the database, (i.e., 60 million geonames) will require much effort and is most easily done over a number of years as the data are needed for DMA projects. The actual loading of the database with geonames, i.e., the mechanics of entering the data is not considered one of NORDA's tasks (see Chap. IV). There are eight tasks involved in this task area.

1. Write Functional Design Specifications (FDS): 4 Months

This document should address:

- Database specifications
- Input modules
- Query modules
- Transliteration of names from maps in non-Latin alphabets
- Update modules
- Output modules
- Intermediate (working) files
- Query and editing capabilities
- Applications programs
- Security classification of database
- Backup/restart capabilities
- Interface with AADES and type placement systems

2. Selection of a DBMS: 6 Months

This task includes:

- Examination of databases of similar size that deal with similar data
- Review of commercially available DBMSs
- Review of DBMSs already owned by U.S. Government with particular attention to those at DMA
- Examination of hardware requirements of DBMSs and hardware availability to DMA
- Performance simulation studies, where appropriate
- Categorization of alternate choices, noting

  - Technical specifications
  - Time required to implement
  - Risks, costs
  - Maintenance/support by developer

- Recommendation of a DBMS

3. Hardware: 18 Months

This task first involves making hardware decisions based on the DBMS, other DMA needs, costs, etc. This portion is concurrent with Task 2. The second portion of the task is acquisition of the DB evaluation/development system hardware. An auxiliary effort to enhance the Geonames Input Station for diacritics is also required. This involves both hardware and software enhancements, with emphasis on reliability, maintainability, and friendly interaction. This can take a long time, since ADP acquisition in DoD is generally a complicated process.

4. Implementation of a DBMS: 11 Months

This task involves construction of the management system for the Geographic Names Database. The level of effort will depend on the choice of DBMS made in Task 2. A commercially available DBMS may go through system (SysGen) in a week or two. A DBMS constructed from a number of existing software building blocks could take one year. The time estimate of 11 months for this task is considered a conservative estimate that will be adequate for most choices. This task has been divided into four subtasks.

Assuming one utilizes an existing DBMS, the first subtask is to **specify the various parameters, tables, etc., needed for system generation, and then generate the system.** For commercial DBMS systems, this is a straightforward adaptation of the generalized software to the specific database relationships. If a database already owned by the U.S. Government was chosen in Task 2, then this specification subtask consists of adapting those modules to new tables, etc.

The second subtask under the DBMS Implementation Task is **software modification and development.** If a commercial DBMS is used, there will probably be a number of modules that will be needed to perform specific functions for the DBMS (e.g., handling "similar" geonames, cf. Chap. II, Sect. F). Also, applications programs using the DBMS will be needed (e.g., transliterations of non-Romanized names from foreign maps). If a DBMS already owned by the U.S. Government is used, then significant software modification may be required. If the DBMS is constructed from various existing modules, then extensive software development is necessary.

Requirements analysis of DMA DB phase II (Corporate Distributed Database System, including the Map Production Centers) introduces a new requirement: to be both hardware and software compatible with the DB phase II architecture.

The third subtask is **software testing.** This is testing by the programmers involved in the software construction. The magnitude of this task depends on the amount of software development in the previous subtask.

The fourth subtask, **system integration**, involves linking the Geographic Names Database to the other systems via working files (called Input Geonames Files and Analyst Geonames Files).

5. Develop Data Capture Strategy/DB Construction:

This task requires developing a strategy for capturing a "limited size" databae for system testing, and a detailed plan (logistics, liaison, etc.) for constructing the complete database. There are various approaches to constructing the database. Because of the magnitude of the task, the database cannot be built instantaneously. The first part of this task is to develop a sequence for data entry into the base. It is recommended that the Geographic Names Database be built in line with DMA requirements for maps and gazetteers. The second portion of this task is to maintain a liaison with DMA personnel as they begin to build this base, carefully monitoring the database and addressing any problems which the DMA staff encounters with the systems.

This approach will:

- Allow the Geographic Names Database to be used on products as soon as possible.
- Cause minimal disruption of DMA work schedules, since the data being entered into the base is the data which DMA would be working on anyway in support of their maps and gazetteers.
- Allow rapid identification of problems. If there is some peculiarity in maps that cause the AADES to reconstruct erroneous data, or if the Mass Input modules of the DBMS sometimes attach the wrong attribute to a name, such problems will be discovered quickly since the data will be extracted from the base in support of products shortly after it is entered.

Although the detailed sequence of data entry will be developed during this task, one can outline a general approach based on the data sources discussed in Chapter IV.

The general sequence is expected to be:

- Load Multiset III tapes.
- Load gazetteer tapes for countries whose names list does not contain a lot of diacritics (e.g., less than 40% of the names need diacritics). Add diacritics to these names using the Names Input station.
- OCR gazetteers of countries whose names list contain many diacritics (e.g., over 40% of the names have diacritics).

- Use AADES to digitize existing DMA maps for areas of current importance to DMA in support of their production schedules.

Non-DMA sources (cf. Chap. IV, Sect. A) of data which are chosen for inclusion in the database will, of course, also be reflected in the detailed sequence.

6. Operational Test and Evaluation: 5 Months

In computer system development, testing occurs at several levels. Basic debugging occurs in software development, program testing, and system testing. The testing is done by computer-oriented staff, as they seek to verify the performance of the codes under a variety of conditions. The objectives of these tests are to ensure accuracy of software codes, accuracy of algorithms, ability to meet functions objectives of the software modules, consistency of interfaces, and ability to meet technical objectives. These three levels of tests are prior to this task.

After all the problems identified in the testing mentioned above have been corrected, a computer system normally undergoes Operational Test and Evaluation. For the Geographic Names Database, however, there is a five-month gap between the end of the system integration and the beginning of the operation test. During this period, the Geographic Names Database will grow from a small prototype base suitable for software testing to a database with a reasonable number of entries (e.g., 750,000). At this point one has enough real data to perform a thorough Operational Test. (While artificial test cases may be used to great advantage during Task 4, few sets of artificial data can match the extreme diversity (or perversity) of large amounts of real data.)

The Operational Test and Evaluation begins with construction of a test plan. This plan is developed with both the system configuration and the functional requirements in mind. The objectives of this task are to:

- Ensure that the system can meet the functional specifications laid out in the FDS.
- Ensure that the Geographic Names Database can be operated by people who have not been involved with the design, construction, or previous testing of the software.
- Ensure that the codes can provide correct outputs to a variety of user-oriented test problems.
- Ensure that the codes can handle erroneous inputs or problem conditions without catastrophic failure.

7. Computer Documentation: 16 Months

The Geographic Names Database must be documented in accordance with military standards [Ref. 2]. This proceeds in parallel with Tasks 4, 5, and 6. This documentation includes:

- Software Maintenance Manuals,
- DBMS Manual,
- User's Manual,
- Operator's Manual.

8. Delivery and Acceptance: 4 Months

Although physical delivery of the system occurs prior to Task 5, formal delivery does not take place until Operational Test and Evalutaion is complete. Acceptance occurs when DMA decides that the base performs to the requirements stated in the FDS. At this time, the Foreign Place Names File (i.e., the index cards) would be frozen, and all updates would be made just to the digital base. Since it may take several years to get the base up to 60 million names, the delivery and acceptance can occur before the database is fully loaded.

DOCUMENTS

FDS (Task 1)
Report on DBMS Options (Task 2)
Detailed Specification of Database Tables and Structures (Task 4)
Strategy/Schedule for Data Capture (Task 5)
Test Plan (Task 6)
Hardware Acquisition Plan (Task 3)
Documentation of System (Task 7)
Test and Evaluation Report (Task 6)

SCHEDULE

TASK/SUBTASK	START (MONTH)	END (MONTH)
1. FDS	0	4
2. DBMS Choice	4	10
3. Hardware	7	25
4. Implement	23	34
• Specification	23	25
• Software	23	30
• Software Testing	27	30
• System Integration	29	34
5. Data Capture Strategy/Construction	12	--
6. Test and Evaluation	36	41
7. Documentation	25	41
8. Delivery and Acceptance	40	44

C. ADVANCED SYMBOL PROCESSING SYSTEM AND DIGITAL TYPE COMPOSITION AND PLACEMENT SYSTEM

The initial plan for this task area is designed to handle either the approach chosen in Chapter III, Section C, or the Alternates in Chapter VI. The level of effort, of course, will vary greatly depending on alternative, but due to the systems with which it must interface, the schedule would not change dramatically.

1. Write Functional Design Specification (FDS)

This document should address:

- Input sources and formats to ASP System,
- Functions needed in ASP System,
- Functions needed in DTC&P System,
- Algorithms used for DTC&P,
- Output format for gazetteers,
- Output format for digital Names Overlay,
- File management,
- Interface with other systems,
- Level of automation in the DTC&P.

2. Hardware

This task involves choice of hardware that can support the functions laid out in the FDS, while being compatible with all computerized systems that will interface with it. The latter is probably the more difficult requirement. Hardware may be either a separate system or a component of another computer system. Coordination with DMA SPOEM will be especially important during the subtask.

3. Software Selection and Adaptation

- Adaptation of Editing Modules
- Adaptation of Sort/Merge Modules
- Adaptation of Symbol Processor Modules
- Detailed Software Specification for Other Modules

4. Construction of System

Using the specification of Task 1 and the results of Tasks 2 and 3, construct the ASP and DTC&P Systems. This task includes the extension and development of software to handle the names/symbols placement task at a semiautomated level. In particular, a study of available symbol placement technology will be made; this activity will include a review of the use of operator interactive graphic techniques for placement, semiautomated names techniques for placement and orientation by computer algorithms, man-in-the-loop cartographic review of computer names placement, and manual editing where necessary (symbol moving by interactive graphics).

Software to accomplish the names placement task will be synthesized from extensions of existing and newly developed software.

5. System Integration

The computer supporting these systems must be linked to the Geographic Names Database, etc., for inputs and to Gazetteer and Map Production equipment for output.

6. Test and Evaluation

Once system integration (and its associated testing) is complete, formal test and evaluation will commence. This will be done on data files typically used in actual DMA products.

7. Documentation

The software, hardware, algorithms, and interfaces will be documented in accordance with military standards.

8. Delivery and Acceptance

This occurs when performance is acceptable to DMA in light of the requirements stated in the FDS.

DOCUMENTS

FDS (Task 1)
Hardware Specification (Task 2)
Software Selection/Specification Report (Task 3)
Test Plan (Task 6)
Test and Evaluation Report (Task 6)
Documentation of System (Task 7)

SCHEDULE

The schedule may be specified up to system integration. At this point, it depends on a number of other systems. Thus, it is premature to give detailed estimates for the remaining steps.

TASK		START (MONTH)	END (MONTH)
1.	FDS	0	3
2.	Hardware	7	25
3.	Software	7	10
4.	Construction	12	25
5.	System Integration	25	31
6.	Test and Evaluation	33	35
7.	Documentation	28	41
8.	Delivery and Acceptance	38	41

D. LEVEL OF EFFORT

The level of effort for the task areas was estimated by considering each subtask and comparing it to similar ADP projects. These were then summed to obtain the following estimates:

TASK	LEVEL OF EFFORT (MAN MONTHS)
Automated Alphanumeric Data Entry System	95
Geographic Names Database	59
ASP and DTC&P	30
TOTAL:	184

E. HARDWARE CONSIDERATIONS

Operational hardware costs are additional to the manpower estimated above and any detailed estimate must await a hardware specification. The hardware costs estimated in Chapter I are for evaluation/prototype development system utilization. The majority of these items should be upgradable for production center use; this issue needs further investigation. The hardware complement listed below is presented to give a general idea of the class of computer and terminals/workstations involved in meeting the requirments of the systems being developed under this subtask.

At present the major pieces of hardware appear to be the following items listed by system.

Automated Alphanumeric Data Entry System

(1) OCR Data Entry System with programmable font recognition capability, diacritic recognition, and programmable symbol recognition.

(1) Keyboard and High Resolution Monitor (for binary image processing).

(1) General Purpose, Large Format OCR Map Reader Workstation.

(2) Dual density (1600/6250 bpi) 9-track digital magnetic tape drives.

(1) Not less than 256KB internal memory.

(2) Hard Magnetic Discs (at least 300 MB each).

(1) General or Special Purpose Computer (unspecified in this study).

Geographic Names Database

(1) General Purpose Computer with very large virtual memory.

(12) Disc drives with 300 Megabyte discs and 35 msec random access times.

(1) Bubble Memory, about 200 Megabytes and 10-15 msec access time, if appropriate.

(20) Workstations: Interactive displays capable of handling diacritics. These may be a mix between simple CRT displays and Enhanced Geonames Input Stations.

(3) Magnetic tape drives, high (6250 bpi and above) density.

Advanced Symbol Processing System and Digital Type Composition and Placement

(1) General Purpose Computer System with high-speed ports for graphics terminals.

(2) Magnetic tape drives dual density (1600/6250 bpi) 9-track digital.

(3) Disc drives with 300 Megabyte removable discs.

(15) Interactive displays capable of handling diacritics. These stations would be the same as those for the Geographic Names Database System and would be used to support Advanced Symbol Processing activities.

(10-15) Image/graphics displays for all-electronic manipulation of names and symbols composition for use for the Digital Type Placement activities.

F. COMPATIBILITY REQUIREMENTS

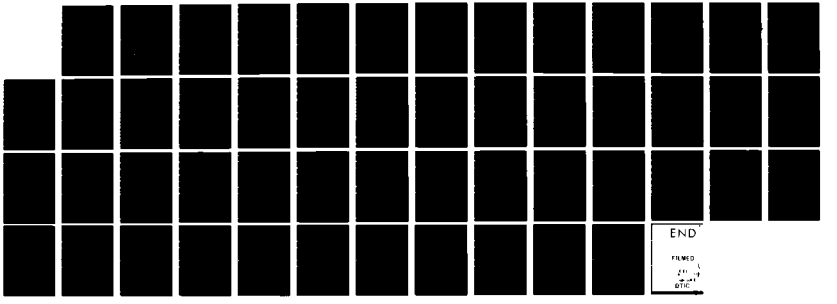
DMA HQ guidance for the preparation of the Subtask Data Sheet requested that four individual DMA development requirements be combined and addressed under the present subtask. Three specific concepts indicated such a combined systems development approach to meet these requirements.

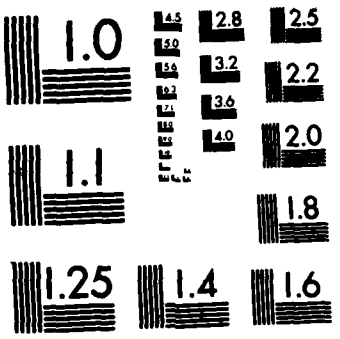
1. Uniform formats for names and symbols information to guarantee to the extent possible easy data sharing/exchange between a number of different application users.

2. A development cycle that can provide a technology and implementation cost sharing.

AD-A126 241 ADVANCED TYPE PLACEMENT AND GEONAMES DATABASE: COMPREHENSIVE COORDINATION PLAN(U) PLANNING SYSTEMS INC SLIDELL LA A E BARNES ET AL. JAN 83 NORDA-TN-189 2/2

UNCLASSIFIED N00014-82-C-0726 F/G 5/2 NL





3. Compatible system implementation in hardware, operating system software, transportable application packages, ease of maintenance, redundancy, etc., will provide considerable logistics and advantages for DMA Centers.

This Comprehensive Coordination Plan is the first step leading to the uniform design and implementation of the systems to meet the DMA requirements. Phase Two (FY83) of the overall planning function for this subtask, preparation of a detailed Implementation Plan, will build on the function descriptions presented above and will explicitly outline and incorporate compatibility requirements necessary to meet the three goals for combining the development efforts for these systems. Close coordination with DMA HQ, Centers, and SPOEM will be required for design and implementing a development project that will effectively provide the advantages listed in these goals.

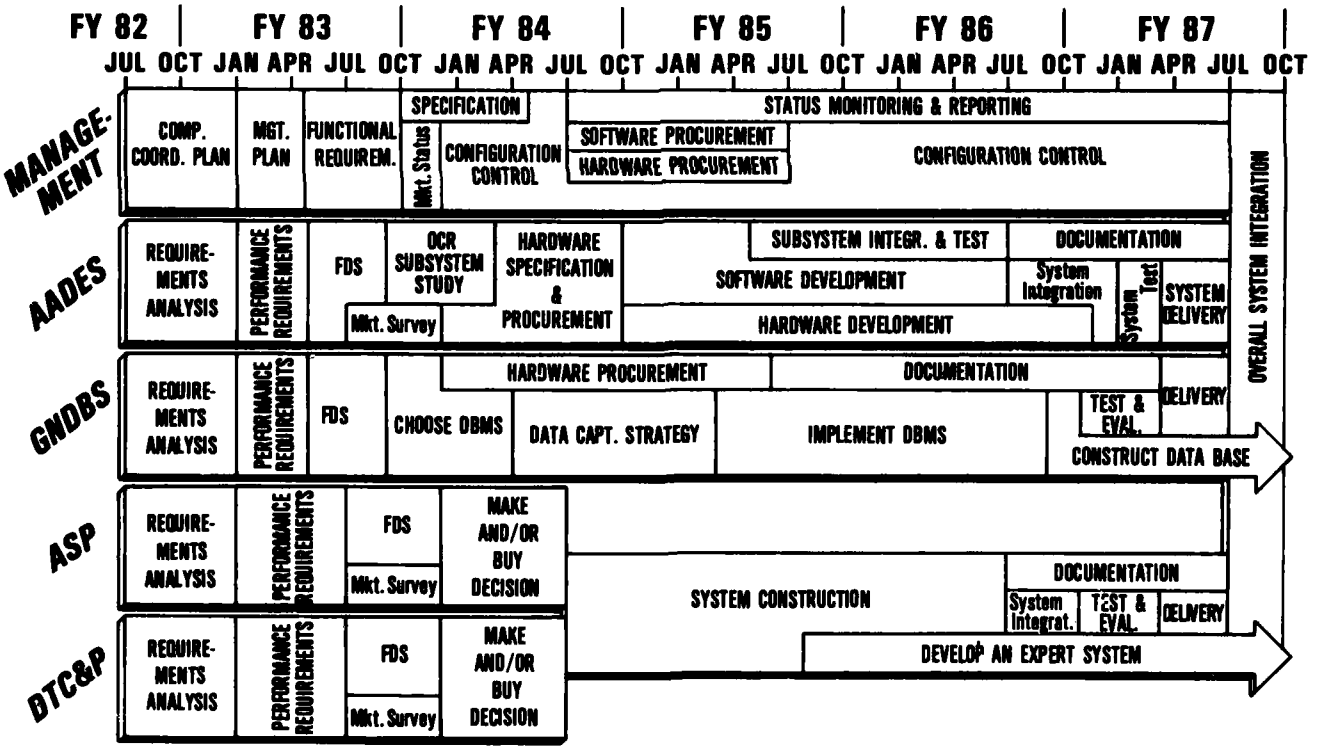


FIGURE 9. Proposed schedule

VI. ALTERNATIVES

The preliminary implementation plan in Chapter V is based on the function descriptions of Chapter III and the functional requirements of Chapter I. Chapter III, in turn, is dependent upon the technical considerations given in Chapter II. The approaches given in those chapters are not the only means to the end, and this chapter explores some of the alternatives. Although the approaches outlined in this chapter were rejected in favor of those found elsewhere in the report, changing conditions may reverse such judgments.

A. DISTRIBUTED DATABASE FOR GEONAMES

In Chapter II three aspects of the database were singled out as requiring attention: the large size, the need for timely access, and the possibility of security classification. The size and timing were addressed in Chapter II, Section J, using a central database, and the question of security was raised for DMA attention. This section presents an alternate approach to these problems: use of a distributed database.

A totally distributed database--e.g, one where each user or small group of users has a microcomputer with a complete, read-only copy of the database--is not practical with current technology. However, one might envision a database that functions as a "lending library": users interested in working with a particular geographic area could "check out" a disc or two for the country or countries of interest and load just that portion of the database onto their microcomputer. Using the figures from Appendix C (3.09×10^9 bytes for the database, 150 geopolitical entities) gives an average database storage requirement of 2×10^7 bytes per country. Twenty megabytes is well within peripheral capacities for microcomputers. The access time for these discs would be somewhat longer than those given in Chapter II, Section J, and the CPU time for the microcomputer would certainly be longer than that of a fast mainframe.

This approach would be more applicable to the cartographers than to the toponymists, for updating the main database would be difficult with such a distributed system. This system might be used for DMAAC if the Geographic Names Database is classified. In such a situation, DMAHTC (which would have the database) could ship selected classified files to DMAAC, which is often less expensive than secure data links.

B. COMPLETE AUTOMATION OF NAMES PLACEMENT

One may envision a Digital Type Composition and Placement system where the placement of names is completely automated. Such a system would operate along the following lines. All geonames would have an associated type font and size (based on feature designator and attribute). From this, the actual size and shape

of the typeset word can be computed, as is currently done in type composition and placement approaches. Geonames of small population centers would then be placed with a standard orientation (e.g., location at lower left corner of the word), geonames of large population centers would be placed using the boundary of the population center. Geonames of features (e.g., rivers) would be placed in some orientation based on the feature, and names of countries would be placed as a sequence of individual letters across the area. All these initial placements would be made by the same set of rules used in the semi-automated approach. In the completely automated approach the computer would then look for overlaps between names. For small type sizes, this may be done by constructing a rectangle around the name and testing for overlapping rectangles. For very large type sizes, such as the name of a country, one would use a rectangle around each letter or, in some cases, possibly the shape of the letter itself. All overlaps would be identified. Using techniques such as integer programming algorithm, attempt to relocate overlapping names by simple movement of some geonames. For example, geonames of small population centers can be moved so that the location is at one of the corners of the name other than the lower left. Geonames of features such as rivers can be shifted along the feature. This can be called the first algorithmic level of overlap resolution, for the operations involved are straightforward for a computer. This would remove some, but not necessarily all of the overlapping.

The second algorithmic level would consist of operations such as reorienting names so they were no longer horizontal and deleting names. Unfortunately, these operations involve more than straightforward algorithms. Orienting names other than horizontally may produce confusion for the users and involves an artistic sense of balance. Deletion of names involves decision. Although the computer may know that five geonames cannot fit into a small portion of the map, if it must choose which one to delete, it must do so based on information available to it. The computer may delete the town of the smallest population, not realizing that the town is of great importance to the map user. Such information, however, could be made available from the geographic names of the database in such an advanced system.

It would appear that while this "second algorithmic level" is easy for a human, it is not a simple task for a computer. To develop a fully automated system, the emerging technology of knowledge-based systems can provide the tools for the necessary knowledge acquisition by encoding the cartographer's decision rules. Testbeds in artificial intelligence laboratories have successfully developed such "transferable" technology (e.g., image understanding via symbolic representation [Ref. 15]. This totally automated approach would reduce the portion of the names placement task, which must be left to the cartographer to a minimum and uses the computer primarily as a decisionmaker rather than simply a graphics aid device.

C. ORGANIZING THE GEOGRAPHIC NAMES DATABASE BY OTHER CRITERIA

The Geographic Names Database is defined in terms of entities and relationships in Appendix A, and a sample database structure is given in Appendix C. What alternative forms are of interest?

In regard to data entities and relationships, there is not much latitude. Some data entities may be dropped (e.g., type of Romanization) or some relations may be ignored (e.g., the inclusion relation), but these are rather minor parts of the database concept. Conversely, new data entities may be added (e.g., military grid reference, intelligence subject code), but these should not drastically alter the database concept. Basic concepts such as geoname, position, feature designator, boundary, etc., will underlie any Geographic Names Database that has the capabilities to meet DMA requirements.

When one passes from concepts of the database to actual database architecture, however, the situation changes. There are many ways to build a database structure that will reflect the concepts of Appendix A. The Simple Database Manager (SDBM) defined in Appendix C is just one approach, and it was used merely to obtain size and timing estimates. By using a different database architecture, could one obtain significantly faster access or significantly reduce storage?

First, one must define a "significant" reduction. By applying various software and storage "tricks" one may reduce the storage requirement of the SDBM by about 30% and the timing by about 10%. But such "tricks" make the database extremely inflexible and very difficult to change. Thus, such methods would not be used. A "significant" savings would be a reduction by a factor of 5 to 10. Such a storage reduction factor would allow different options in data storage. The current approach (cf. Chap. II, Sect. J) involves using a number of large storage media. One must reduce storage to the 300-500 megabyte range to put it on one large movable head disc or two large bubble memories; i.e., a reduction factor of 5-10 is needed to change the hardware approach.

The SDBM of Appendix C is organized by country, and within a country it is organized alphabetically. In this sense, it preserves the same order as the current Foreign Place Names File (i.e., the card file). This reflects the gazetteer needs (i.e., gazetteers by country, with geonames sorted alphabetically). Is such an organization necessary or even warranted?

- Should geonames be grouped by country? Gazetteers are produced by country and maps are produced by area, which contains one or more countries. Thus, such an approach seems to be a reasonable idea. But anything that preserves the type of grouping used for maps and gazetteers will suffice such as grouping by latitude-longitude.

- Should Geonames be stored alphabetically? Alphabetical order is a requirement for gazetteers, and it is helpful in other instances. Alphabetical order is great for humans, who use it all the time (e.g., phone books, dictionaries), but is is not particularly good for computers. There is a very uneven distribution of letters in languages (e.g., in English, a preponderance of e's, a's, etc.), and combinations of letters (e.g., in English, th, qu, ough, etc.). Thus, computers often store alphanumeric data data by schemes other than alphabetical order (e.g., hashing algorithms). And there are methods of compressing alphanumeric data in computers by recoding it [e.g., Ref. 7], although they do not seem useful to this database. Assuming that the average gazetteer has 3×10^4 geonames (cf. Appendix C, Sect. 2), this data can be sorted in a few minutes.

From the answers given above, it is seen that other database architectures are feasible. Three are briefly considered:

- Storing all geonames alphabetically: Geonames from all countries could be combined, sorted alphabetically, and stored. There are some minor advantages to this system, but not significant ones. There are several disadvantages, however, primarily in slower access times, but also in maintenance difficulties. Such an organization of the database is not recommended.

- Storing geonames by position: Since a map covers a definite area of the world, organizing data by position seems to be a good idea. Since a country usually occupies a contiguous area, such an organization is useful for gazetteers as well. The major factor in this approach is that position is two-dimensional (i.e., latitude and longitude) while computer storage is essentially one-dimensional (i.e., addresses). There are various ways to transform the two-dimensional scheme into a one-dimensional approach, but probably the most useful one is to divide the world into "strips" in either latitude or longitude and order the geonames within each strip. A variant of this is to divide the world into latitude-longitude "squares," and store all geonames of a square together. For timing, the results are mixed. Map production queries will be faster under this organization than under the SDBM, but gazetteers will now be slower for the query and, in addition, will require a sort. Storage requirements are reduced, but only by about 2%, which is not significant.

- Storing geonames by attribute: There is a standard set of map scales used by DMA (see following table). (This is not to say that DMA only produces maps in these scales, for some DMA maps are at other scales.) Depending on feature

designator (e.g., river, population center) and feature attribute (e.g., population, importance to DMA product user), a geoname will appear on a certain set of DMA maps. Normally, one can state that if a geoname appears on maps of one scale, it will appear on maps of smaller scales that cover the same area (i.e., the vicinity of the object). Because maps are produced for different users, this is not always true. (For example, a geoname found on a JOG [ground] at 1:250,000 may not appear on a corresponding Air Target Material product at 1:200,000.) However, one may use a mathematical relation called a Partial Ordering to circumvent this problem. Thus, for simplicity, we will call map A "finer" than map B if the geonames of map B are found on map A in the (nonempty) region common to both maps.

STANDARD DMA MAP SCALES

SCALE	PRODUCT
1:5,000,000	Global Navigation Chart
1:3,000,000	Jet Navigation Chart
1:2,000,000	Jet Navigation Chart
1:1,000,000	Operational Navigation Chart
1:500,000	Tactical Pilotage Chart
1:250,000	Joint Operations Graphic (Ground, Air, and Radar)
1:200,000	Air Target Material
1:100,000	Topographic Line Map
1:50,000	Topographic Line Map
1:25,000	City Map
1:12,500	City Map

Consider the following organization of geonames: For each standard map product, build a file of all geonames that would be found on that map product (whether or not a map in that series has been produced which covers that area). If map A is finer than map B, delete all geonames from A's file that are found in B's file. This reduces the redundancy. Within a map product class, sort geonames by area*. Then, to query the database in support of a particular map product A, pull up data in that geographical region from all maps B such that map A is finer than map B. Since gazetteers contain a set of geonames that are roughly equivalent to those found on 1:250,000 maps, gazetteer queries can be handled in a similar fashion.

*There is a slight problem with sorting by map sheet, since due to earth curvature and flat maps, adjacent maps of the same product line overlap.

A nonmathematical description of this approach is as follows: Consider a visual display (e.g., a very high resolution CRT) hooked to the database. One may display portions of the world at various fixed scales. At any scale, the display shows an electronic image of the geonames available at that level. As the scale becomes larger the field of view (measured in latitude and longitude) becomes smaller, but new geonames appear between the old ones, as smaller towns, etc., come into view. The display is, in a sense, an electronic map whose contents depend on the scale at which it is viewed.

Such an approach to database architecture requires somewhat more care than the SDBM and is more difficult to modify (e.g., changing the population of a town may result in the geoname being moved from one part of the database to another). This is a highly integrated database approach that sacrifices some flexibility. Although a detailed study has not been done, this approach may have faster access times for queries than the SDBM.

As pointed out in the beginning of Appendix C, we are not trying to establish a database architecture at this time, but are merely investigating the feasibility (in terms of size and timing) of approaches.

REFERENCES

[1] R.F. Augustine, D.R. Caldwell, and D.E. Strife, A Prototype Geographic Names Input Station for the Defense Mapping Agency, presented at Auto Carto IV, September 1982.

[2] Automated Data Systems Documentation Standards, MILSTD 7931-5S, OSD, September 1977.

[3] G. Carlson, Techniques for Replacing Characters that are Garbled on Input, Proc. 1966 Spring Joint Computer Conference, p. 189-192, 1966.

[4] E.F. Codd, Relational Database: A Practical Foundation for Productivity, Communications of the ACM, Vol. 25, No. 2, p. 109-117, 1982.

[5] Conceptual Sims User's Manual, Technical Services Corp., June 1981.

[6] Development of an Automated Cartographic Capability, DMAHTC, April 1982.

[7] D.J. Dodds, Reducing Dictionary Size by Using a Hashing Technique, Communications of the ACM, Vol. 25, No. 6, 1982.

[8] Functional Specifications for Terrain Edit System/Evaluation Matrix Processing System (TES/EMPS), DMAHTC, March 1981.

[9] Geographic Names Data Base Study, U.S. Army Topographic Command, Corps of Engineers, Washington, D.C., July 1969.

[10] W.C. Liles, E.D. Nugent, T.V. Russotto, and J.T. Serelis, Source Information Management System (SIMS), System/Subsystem Specification TSC-PD-B687-1, September 1981.

[11] Geographic Names Information System, U.S. Geologic Survey, September 1982.

[12] Computer Programs for Detecting and Correcting Spelling Errors, Communications of the ACM, Vol. 23, p. 676-687, 1980.

[13] M. Stonebraker, Operating System Support for Database Management, Communications of the ACM, Vol. 24, No. 7, 1981.

[14] Requirement Analysis for Phase II DBS, Technology Service Corp., Landover, Md., July 1982.

[15] P. Raulefs, Expert Systems: State of the Art and Future Prospects, German Workshop on Artifical Intelligence, p. 98-111, 1981.

APPENDIX A

CONCEPTUAL RELATIONS OF THE GEONAMES DATABASE

1.0 INTRODUCTION

The Geonames Database is a major component of this report: it affects all of the original DMA task statements. Thus, this database receives special attention in this report. Chapter II, Section E, discussed the types of questions that such a database must answer. This appendix addresses the Geonames Database from a conceptual point of view. It is concerned with what the nature of the data is and what sort of interrelationships exist between data components. This appendix does not attempt to design a database management structure (DBMS). The choice of a DBMS, either by

- selection of one of the many commercially available DBMSs,
- adaptation of a DBMS currently used by the U.S. Government, and
- building a DBMS tailored to DMA needs, using existing components of other DBMSs whenever possible.

is a separate study in itself and is beyond the scope of the FY-82 effort. Although some of the terms used (e.g., relation) seem to have implications about the underlying DBMS (e.g., that it is a relational database), no such implications are warranted.

The database should reflect facts, such as "London is the name of a place in the country of England."* This consists of entities, such as the names "London" and "England" and relations between entities, such as "x is a place-name in country y." The next two sections consider these components.

2.0 ENTITIES

2.1 LIST OF ENTITIES

There are a number of entities relating to geonames. The computer entities are given names corresponding to their real-world counterparts; however, these terms may not correspond to the "official" definitions. This is a list of all entities that the database can deal with: it is not a list of all data items one needs in an input record to build the database!

*The human mind deals with ambiguity much better than the computer does. The simpler statement "London is in England" becomes problematical when dealing with London, Ontario.

2.1.1 Geoname: A geoname is a name of a population center, feature, etc. Geonames, by themselves, are often not unique. To uniquely specify a "place," additional information is needed. The three most common additional attributes are country, type of feature, and position (i.e., latitude and longitude). Often, specification of country is adequate to uniquely associate a geoname with a database entry.

2.1.2 Country Name: This is what is commonly used to refer to a country. Normally, a country will have just one name, but it may be referred to by aliases or abbreviations. For example, Union of Soviet Socialist Republics, U.S.S.R, and C.C.C.P. each refer to the same country. Unlike a geoname that may refer to a variety of places, we impose the restriction that a country name apply to only one country.

2.1.3 Official Country Name: These names are contained in the set of country names, but now there are no aliases. Each country has only one official name (DMA may base the choice on State Department policy). The important property (as far as the database is concerned) is that the number of official country names equals the number of countries.

2.1.4 Territorial Name: Territories, as far as the database is concerned, are regions administered by a country, which the user may not want to consider when making a query on the country. For example, a user working on a map of Europe may call up Spain in the database, but not be interested in Melilla, since it is in Africa. Or a user interested in locating certain towns in the United States may not be interested in all the little islands that the U.S. holds as trust territories. As with country names, aliases and abbreviations are allowed.

2.1.5 Official Territorial Name: As with official country name, there is one official territorial name for each territory.

2.1.6 Geopolitical Entity Name: These are either official country names or official territorial names. The term is used simply as a database convenience to refer to something that may be either a country or a territory.

2.1.7 Index of Geoname: The index of a geoname is an attribute not visible to the user, but one which will make the discussion of the database much simpler. In practice (cf. App. C), it may not even be stored in the database.

The pair (geoname, geopolitical entity) often forms a unique identifier, but not always. Addition of type of feature and position of feature would give a unique designator, but neither of these two entities are ideal for this purpose. Due to the problems associated with building this database from different sources, conflicts may arise between sources on type of feature (e.g., when does a hill become a mountain?) or position of

feature (e.g., latitude and longitude of a large feature, such as Los Angeles, California, is ambiguous). Since conflict resolution of disparate data sources will be an important part of this database, neither the type of feature nor its position should be part of the primary database key. To obtain a unique identifier without resorting to type of feature or position, the index of a geoname is introduced. If for a particular pair (geoname, geopolitical entity) the database contains n such entries, then an index (from 1 to n) is associated with each of them, so that the resulting triple (geoname, index, geopolitical entity) be unique within the database.

2.1.8 Geoname Data Items: This term covers a number of entities which are associated with a triple (geoname, index, geopolitical entity). A given triple need not have values for all of these entities, and those entities which it does have may have come from different data sources.

- **Position.** Position is an ordered pair (latitude, longitude). The database keeps it in a form (such as degrees, minutes, seconds) adequate to support its users. However, the data source may not be of adequate resolution to support all users.

- **Source and Accuracy of Position.** This entity describes the source of the position (e.g., from a JOG) and the resolution of that source. It would be in the form of a set of codes.

- **Feature Designator.** This describes the type of feature (e.g., population center, mountain).

- **Attributes of Feature.** Depending on the type of feature this would describe some attribute such as population or military importance.

- **Boundary.** If the real-world object corresponding to the geoname is large (e.g., New York City), then for some map products one may need to represent it, not as a standard symbol, but in outline. The boundary entity would be equivalent to a string of data describing the boundary.

- **Type of Romanization.** If the object is in a country using a non-Latin alphabet and the geoname is Romanized, then this describes the system of Romanization used to transliterate the name.

- **Non-Romanized Name.** Since DMA produces some bilingual products, certain names are required in their non-Romanized form.

- **Alphabet Translation Code.** If a non-Romanized name is entered, some designation is needed to tell how the non-Latin alphabet was stored in the computer. (This could be a designator to a translation table, similar to the system used now on Multiset III.)

2.1.9 Adjacency Possibilities: These entities are part of a fixed table of three elements. Two geopolitical entities may be either

- identical (e.g., France and France),
- adjacent (e.g., Spain and Portugal),
- not adjacent (e.g., France and Portugal).

These entities are used to support queries of the form "country x and adjacent countries."

2.1.10 Map Sheet Identification: This is a code that identifies both the type of map and the individual map within the product series.

2.1.11 Map Sheet Limits: This specifies the limits (in terms of latitude and longitude) of a map sheet.

2.1.12 Geoarea Names: There are times when a user may wish to consider several nearby geopolitical entities. For convenience, geoarea names may be defined to refer to often-used collections of political entities. Typical geoarea names might be "Africa" or "Indian Ocean Islands."

2.1.13 Other Entities: There are other entities which the DBMS will use, but they are used for audit trail or management information statistics (MIS) purposes. These include things like user identification or date-time-group of update. These types of entities are fairly standard and need not concern us now.

2.2 DOMAINS

Each of the entities described in Section 2.1 comes from some domain of values. Most of these domains will grow as the database grows. These domains may be considered as sets (in the mathematical sense). For convenience in the next section, symbols will be given to these sets. The domains are:

- G, the set of geonames
- C, the set of country names
- C_O, the set of official country names
- T, the set of territorial names
- T_O, the set of official territorial names

- P_O, the set of geopolitical entities
- Z, the set of indices of geonames
- For geoname data items, domains are considered individually
  - B, the set of positions (i.e., latitude, longitude pairs)
  - S, the set of source/accuracy codes
  - D, the set of feature designators
  - Att, the set of feature attributes
  - Y, the set of boundary strings
  - R, the set of Romanization types
  - N, the set of non-Romanized names
  - T, the set of Translation codes
- Adj, the set of three Adjacency codes
- M, the set of map sheet identification
- L, the set of map limits
- O, the set of geoarea names

In the notation of set theory, C_O is a subset of C, T_O is a subset of T, and P_O is the union of C_O and T_O, or

$$C_O \subset C$$
$$T_O \subset T$$
$$P_O = C_O \subset T_O$$

For convenience, one may also wish to define P to be the union of C and T.

3.0 RELATIONSHIPS AMONG ENTITIES

3.1 RELATIONS

There are a number of relations among the various entities. These may be listed as follows.

3.1.1 To each country name, a unique official country name is associated.

3.1.2 To each territorial name, a unique official territorial name is associated.

3.1.3 To each official territorial name, a unique official country name is associated. This is the name of the country which the U.S. recognizes as having cognizance of that territory at the present time.

3.1.4 To each pair of political entities, a unique adjacency possibility code is associated, indicating whether or not they are adjacent.

3.1.5 To each geoarea name, more than one geopolitical entity is associated.

3.1.6 To each map sheet identification, a unique map sheet limit is associated.

3.1.7 We will insist that every geoname be associated with a geopolitical entity (even if a geopolitical entity such as "noncountry" has to be invented for geographical features such as the Mid-Atlantic Ridge, which cannot be associated with any standard country). There is a relation between geoname, index of geoname and geopolitical entity, indicating that the geoname correspond to one or more "places" in the geopolitical entity. This relation may be represented by triples of the form (geoname, index, geopolitical entity). From the way the index was defined (cf. Sect. 2.0), every such triple in the database is unique. This triple will form the primary key to the relation between the geonames data items: i.e., by specifying a valid triple (geoname, index, geopolitical entity), the DBMS may identify a unique "place" and locate the data items that "belong" to it. This triple will be referred to as the "geoname key."

3.1.8 To each geoname key, one or more geoname data items are associated. There are a few restrictions on this relation.

- For each type of geoname data item, only one entity may be associated with a given geoname key. For example, two different population figures cannot be associated to a given city. This is important for conflict resolution when two data sources differ. If someone asks what the population of the city is, the DBMS should not answer "either 80,000 or 4,250,000 people."

- To query by map sheet or by specifying latitude/longitude limits, we will insist that every geoname key have a position associated with it. If in some unusual situation the latitude and longitude were unknown, an estimate can be made (in this case the source/accuracy field would note that the position is not accurate).

- The data item "attribute of feature" may be associated with a geoname key only if the data item "feature designator" is present.

- The data item "alphabet translation code" may be associated with a geoname key only if the data item "non-Romanized name" is present.

3.1.9 Within a given country, there may be geonames corresponding to various levels (e.g., the Bronx is a borough within New

York City, which is within New York State). Thus, there is a relation on pairs of geoname keys that may be called "inclusion."

3.1.10 A given "place" may be known by more than one name, and the database will have more "names" than places. For example, although "Saigon" is now "Ho Chi Minh City," a database user working from historical material may ask for data on "Saigon." Thus, there is a relation among pairs of geoname keys which may be called "aliases."

3.1.11 If a given object has aliases in regard to its name(s), a mapmaker may want only one of the names. One may call this the "official name" or the "preferred name" of the object. Thus, there is a relation among pairs of geonames that may be called "preferred name."

3.2 NOTATION FOR RELATIONS

These various relations may be abbreviated by symbols, using mathematical notation. This will simplify diagrams, as well as the database structure described in Appendix C.

3.2.1 The function that maps country name into official country name may be called the Name function, and is represented by the symbol n. The mapping is

$$n: C \rightarrow C_O$$

In practice, this is most easily visualized as a two-column list.

3.2.2 The function that maps territorial name into official territorial name may also be called the Name function, and is represented by the symbol n. The mapping is

$$n: T \rightarrow T_O$$

As noted in Section 2.2 of this appendix, in the database itself the sets C and T may be combined, just as C_O and T_O were combined to form P_O. Thus, the use of the same symbol as in Section 3.2.1 is natural.

3.2.3 The function that maps official territorial name to official country name may be called the Cognizance function and is represented by the symbol c. The mapping is

$$c: T_O \rightarrow C_O$$

3.2.4 The function that shows whether a pair of geopolitical entities are adjacent may be called the Adjacent function and is represented by the symbol a. The mapping is

$$a: P_O \times P_O \rightarrow Adj$$

In practice this may be most easily visualized as a square table.

3.2.5 The relation that associates a set of geopolitical entities to each geoarea name is one-to-many and may be many-to-many (if a geopolitical entity is in more than one geoarea). This relation may be reprsented by the symbol h, and the many-to-many mapping as

$$h: O \langle\!\langle\!\rangle\!\rangle P_O$$

This may be visualized as a hierarchical list, where under each geoarea name is listed the appropriate geopolitical entities.

3.2.6 The function that takes a map sheet identifier and provides the map sheet limits may be represented by the symbol m. It would appear to be a one-to-one mapping, i.e.,

$$m: M \rightarrow L$$

It may be visualized as a table of map sheet numbers and corresponding strings of (latitude, longitude) pairs.

3.2.7 The relation between geoname, index, and geopolitical area will be central to the DBMS. No matter what database software and structure are used, they will conceptually depend on this relation, although it is likely to appear in a very disguised form in the actual code. Let the symbol k represent this relation on three domains, i.e.,

$$k: G \times Z \times P_O$$

For convenience, denote the subset of $G \times Z \times P_O$ for which the relation holds (i.e., the set of triples in the database) by the symbol K. The relation may be visualized as a table of three columns, where each row represents a valid combination.

3.2.8 The relation k is part of a larger relation between geoname, index and geopolitical entity, position, source/accuracy of position, feature designator, attribute of feature, boundary, type of Romanization, non-Romanized form, and alphabet translation code, which may be called the Geoname data item relation and is represented by the rather cumbersome form

$$g: G \times Z \times P_O \times S \times B \times D \times Att \times Y \times R \times N \times T$$

This relation may be thought of as an electronic equivalent of the current card file: different cards have different amounts of information, but all have the basic information.

3.2.9 The relation of "inclusion" is a binary relation on the set K of geoname keys. It is a transitive relation, and may be represented by the symbol inc.

3.2.10 The relation of "aliases" is also a binary relation on the set K. It is a symmetric, reflexive, transitive relation (i.e., an equivalence relation).

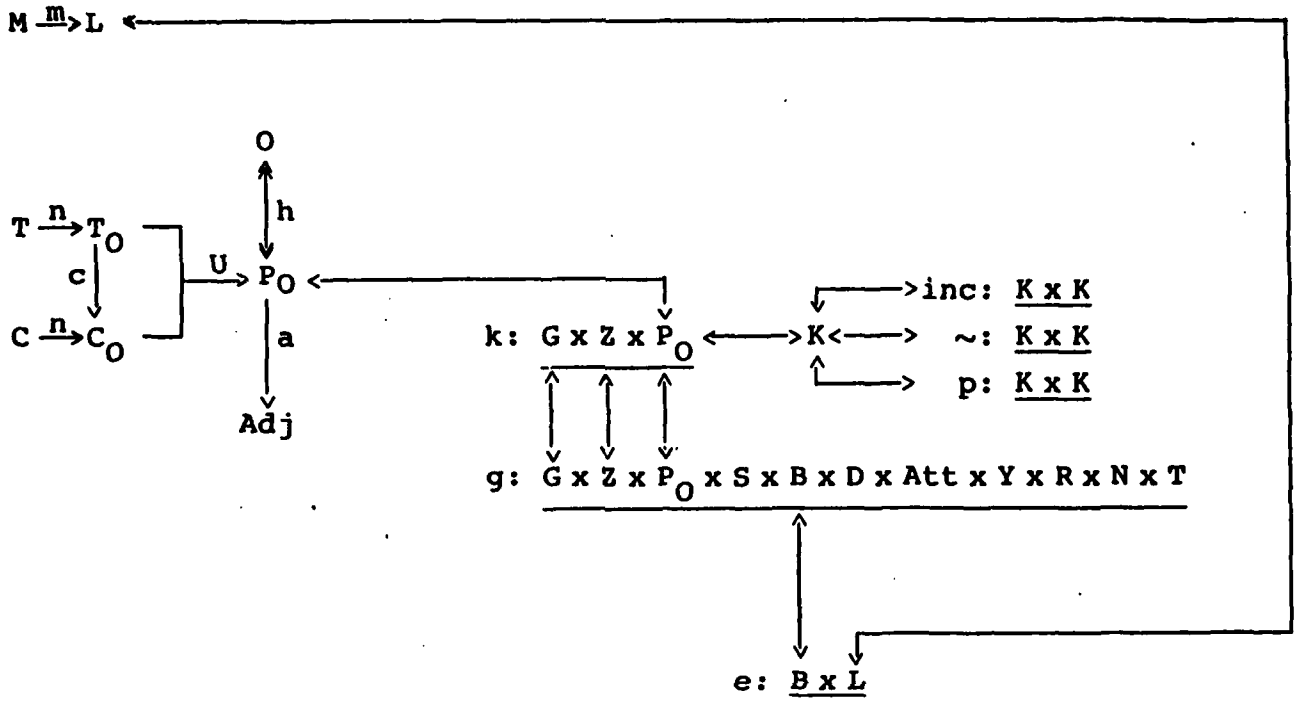
3.2.11 The "*preferred name*" relation is many-to-one, but it is not a function, for it is not defined for names not having aliases. (In the case of a name without aliases, one could define the preferred name to be the only name, but this only gives a notational advantage.) This is a binary relation on K, and may be denoted as p.

3.2.12 The relation between (latitude, longitude) pairs and map limits is expressed by the set theory concept "is an element of." This relation may be represented by the symbol e. The relation is then written

$$e: \underline{B \times L}$$

3.3 DIAGRAM OF CONCEPTUAL DATABASE

Putting the entities and relations of Sections 3.1.2 and 3.2.2 together gives the following functional diagram. Again, it is stressed that in actual implementation the data structure will, in all likelihood, look vastly different from this relational chart.



Legend:

Sets (and domains) are capital letters

<———> indicates the same set

——n——> indicates a function

«——h——» indicates a mapping

e: B x L indicates a relation

FIGURE 10.

APPENDIX B

ORIGINAL DMA REQUIREMENT STATEMENTS

The four original DMA requirements statements date from 1979. The principal portions of these statements are reproduced in this appendix. The first three requirements are from HTC, the last is from AC.

1.0 REQUIREMENT 1--AUTOMATED ALPHANUMERIC DATA ENTRY SYSTEM

1.1 BACKGROUND

1.1.1 Operational Setting: The present capablity for inputting text and numerical data into a computer for processing consists of keypunching cards, keying a scope terminal or by using a highly constrained Optical Character Recognition device. A significant amount of resources are expended for data preparation for such items as Time and Attendance Records, Topo Data Library System, Bathymetric Data Library System, Geographic Names, Positional Data, Imagery Data Fiels, and Hydrography Files.

1.1.2 Deficiency: This labor-intensive method is error-prone and results in many hours being spent in correcting data that has been inputted. All of the data that was originally scheduled to be in the Topo Data Library System (TDLS) has not been inputted due to the lack of resources (there are approximately 600,000 unique items in TDLS). There are numerous other databases in the planning stage, and an effective economic way must be found to input data or else the systems will never become fully developed.

1.1.3 Related Work: R&D Project "Voice Recognition." RADC has been working on a system that will automatically digitally enter sounding data onto a magnetic tape. This system is activated and operated by using an operator who would speak into a microphone. DMAHTC is procuring a "Data Entry Edit System" that will support several existing and proposed database activities. This is to be a minicomputer system with interactive entry/edit terminals; however, it still has to be manually fed.

1.2 R&D OBJECTIVES

Investigate and analyze current computer I/O devices and requirements at DMAHTC and develop an optimum cost-effective, 99% error-free system that will convert alphanumeric data into a computer-readable form.

2.0 REQUIREMENT 2--GEOGRAPHIC NAMES DATABASE SYSTEM

2.1 BACKGROUND

2.1.1 Operational Setting: There are two branches in the Geographic Names Data Division, Scientific Data Department.

- The Toponymic Branch accomplishes geographic and toponymic research that leads to the development of policies, procedures, and directives for the treatment of geographic names. They maintain a worldwide Geographic Names Database on 4-1/2 million index cards and the DoD Foreign Place Names File. They are responsible for the publication of gazetteers, politico/administrative studies, and glossaries. Approximately 270 country gazetteers have been published to date. The name information in the gazetteers is stored on magnetic tapes.

- The Applications Branch assembles geographic names data and related descriptive information for DMA topographic maps, nautical charts, place names indexes, and special-purpose products. They maintain a population file, a boundary inventory, and provide identification, description, and designation category of all natural and cultural features that require labeling on DMA graphic products. They also develop specifications for names presentation on DMA cartographic products. Names placement data is furnished to the Symbols and Names Placement System (SNAPS) on handwritten data sheets. The SNAPS personnel convert the data to either magnetic tape or paper tape for the actual names placement operation.

2.1.2 Deficiency: The development of a Geographic Names Database into a common and efficient reusable format (digital) is required to minimize rework of the same data by the two branches. Significant savings in time and production funds may be realized through the design and implementation of this database. The 270 gazetteers are supposed to be updated at the rate of 6-10 per year. This is due to the limited amount of resources available. Some gazetteers are 10-25 years old, and their names cannot be used for "Names Placement" since more recent and modern locally used names can be found by researching text, magazines, phone books, foreign-published maps, or other sources. The result is that some names appearing on DMA graphic products are not the same as those in the gazetteers. The current magnetic tape method of producing a gazetteer has no provisions for furnishing diacritics, special alphanumeric characters, and lower-case letters. The tape that is used to drive SNAPS is normally destroyed after it is used, since it is formatted to place names on a particular map sheet at a prescribed scale. This tape is not reusable if a map of a different scale is made of the same area. The magnetic

tape that is used for the publication of a gazetteer cannot be used by "Names Placement" because it does not contain geographic coordinates to arc seconds, population data, type font size, style color, and "placement instructions." Also, the country gazetteers are printed in upper case letters, while names on maps are in upper and lower case.

2.2 R&D OBJECTIVE

Develop a Geographic Names Database System that will be economically responsive to the needs and requirements of both Branches in the Geographic Names Database Division. Possible candidates for elements of system hardware that could be used are the Electron Beam Recorder, the Cathode Ray Tube Print Head, and ETL's interactive 3-D view graphic system. The applications software and file structure must be designed to enable rapid update and quick response to queries.

3.0 REQUIREMENT 3--ADVANCED SYMBOL PROCESSING SYSTEM

3.1 BACKGROUND

3.1.1 Operational Setting: The type placement and symbol processing system situation at DMAHTC is presently in a chaotic state. The present Photon type placement systems used at DMAHTC are out of date and replacement parts are nonexistent. The Geographic Names Files currently consist of trig lists, card files, catalogs, and several other archaic methods of retaining information. Presently, diacritics and special notations are corrected or initially performed by manual methods, with no records being kept on what is done. Digital data are being used more and more, but the present fonts, type, and words are not in the proper format or digitized to be used in these newer data types.

3.1.2 Deficiency: Commercially available or R&D-developed type placement systems do not satisfy DMAHTC requirements for diacritics and kerning. For effective productivity, operators must view the diacritics as they will appear in final type. There is need within DMAHTC to establish a library or disc file of the most commonly used fonts, words, and type. To fully utilize the CRT Print Head System, the Electron Beam Recorder or future systems, this library or disc file is a must. In the near future, several systems will need to draw from these digitized files, and these digitized files will also provide DMA with a standardized file. An interface between the old system SNAPS (Symbols and Names Placement system) and the newer systems being delivered is lacking.

3.1.3 Related Work:

R&D--Type Composition Console
R&D--Geo (Geographic) Names Files

R&D--Font Digitization
R&D--SNAPS to CRT Print Head Conversion
TIP--OT&E of the CRT Print Head System

The Type Composition Console is an R&D item or system being developed for DMAAC. Presently undergoing Operational Testing and Evaluation (OT&E) at DMAAC, it lacks a diacritics or kerning capability. Geonames Files are an R&D item to develop a capability for storing approximately 10,000 commonly used names on disc or magnetic tape, with easy access and retrieval functions. Font digitization, a USAETL in-house effort, is to digitize the most commonly used fonts and type within DMA, again with an easy retrieval capability. The SNAPS to CRT conversion is being accomplished under R&D, but it should be expanded to cover other type or word placment systems, as well as the Electron Beam Recorder. The TIP is to operationally test and evaluate the CRT Print Head Systems newly installed on the Gerber Precision Plotter and Concord Plotter at DMAHTC.

3.2 R&D OBJECTIVE

To provide DMAHTC with a General Purpose Symbol Processing System suitable for use with a variety of functional operations (i.e., Names and Placement System, Notice to Mariners System, Geographic Names system, etc.). This system can provide the basis for a replacement to the Photon and drum coordinatograph portions of the present Names Placement System, and with a standardized font, word, and type library or file of the most commonly used fonts, words, and types. To provide the proper interfaces to better utilize the current and projected resources within DMA. To provide easy access to the digitized fonts, names, words, and type by the many new items of equipment.

4.0 REQUIREMENT 4--DIGITAL TYPE COMPOSITION AND PLACEMENT SYSTEM

4.1 BACKGROUND

4.1.1 Basic Objective: To develop an all-digital system for the composition and placement of typographical names shown on chart products.

4.1.2 Requirement: This system is needed to prepare names information depicted on Air Target Material, and Navigation and Planning Chart products used in the operation of U.S.A.F. weapon systems, as well as in support of space exploration programs.

4.2 SPECIFIC R&D REQUIREMENT

Present DMA typographical systems compose and position the characters that comprise geographic names and identifiers via keyboard cursor and aperture systems. This requirement is to address development of a more advanced system that would permit these functions to be performed more interactively and efficiently, through increased use of electronic display technology.

APPENDIX C

ADDITIONAL DISCUSSION OF DATABASE STRUCTURE AND TIMING

1.0 APPROACH

The utility of the Geonames Database depends to a degree on how fast it can respond to queries. Obviously, if it takes a half an hour to look up a name, it would be faster to go to the existing card file (assuming that it was one of the 4-1/2 million names in the file). If the database requires two days to retrieve all geonames for a map product, then the user must request the data long before it is actually needed. If the Geonames Database is envisioned as a DBMS with both interactive and batch capabilities, then the interactive queries must have a sufficiently short response time that the user at the I/O station (CRT or whatever) does not "get bored" waiting for a response.

In the body of this report, it is stressed that the DBMS should be flexible, be independent of both applications program and actual data storage and be easily adaptable to new needs. While such an approach is necessary from the standpoint of good database design, it offers no help in answering questions about timing. Specifying that the DBMS should determine an efficient way of retrieving data does not tell us how it will retrieve the data. Also, as DBMSs become more generalized and complicated, it becomes difficult to estimate how long the "black box" will take for various types of queries. However, for a very simple DBMS working with a simple database structure, timing estimates are usually easy to obtain.

The analysis in this appendix rests on one basic assumption:

> **Assumption**: A DBMS using state-of-the-art methodologies and techniques, which possesses (in some degree) attributes such as flexibility, data dependence, etc., can be built which is almost as efficient as an inflexible, simplistic, and specialized DBMS using archaic (i.e., 1960) design practices.

In other words, the additional overhead one pays for generality and adaptability is more or less offset by the advance in software techniques over the past two decades. A DBMS that can choose its accessing strategy could (conceptually) choose the same strategy that the specialized DBMS used. This assumption is a generalization of a claim by E. F. Codd [Ref. 4] that a relational database should be able to do anything that a nonrelational database can do, and do it almost as fast. While his claim is in regard to a particular type of database and does not imply that our assumption is true, nevertheless, it is conceptually similar.

The approach used in this appendix is to design a very simple DBMS that has only two redeeming attributes:

- It can support the types of queries listed in Chapter II, Section E.

- One can obtain an upper bound on the number of disc accesses needed to perform a typical query of each type listed in Chapter II, Section E.

This example will be used solely to provide an estimated timing: it is in no way implied that such a structure should be adapted for the DBMS that will run the Geonames Database. This structure fails to provide the flexibility and adaptability needed to prevent premature obsolescence: it is simply a demonstration that a technique exists whereby reasonable accessing times may be obtained. This structure will be called the Simplistic Database Manager (SDBM).

2.0 DEFINING THE DATABASE STRUCTURE

2.1 BASIC OBSERVATIONS

The SDBM makes use of the following facts:

- A query on a Map Sheet may be converted to a query on one or more geopolitical entities, with a qualification on position.

- A query on Geographic area (latitude-longitude limits) may be converted to a query on one or more geopolitical entities, with a qualification on position.

- The qualification clause on feature designator and feature attribute may be a complicated logical expression involving the logical operators "and," "or," and "not." Since the number of feature designators is finite, and the feature attributes (e.g., population) may be grouped into a small number of classes, negation of any feature or attribute may be represented by the "or-ing" of the other attributes or features. This allows elimination of the negation operator "not" at the lowest levels. All negation operations may be reduced to this level using DeMorgan's laws. Thus negation may be eliminated from an expression concerning just feature designator and feature attribute. By applying the distributive laws, the resulting logical expression may be reduced to the "or-ing" of a number of "and" clauses.

2.2 DEFINITIONS OF VARIABLES ASSOCIATED WITH THE SDBM

The SDBM will use a number of the sets defined in Appendix A, Section 2.2, and relations defined in Appendix A, Section 3.2. For estimating sizes of files and number of accesses, several variables are defined that reflect the sizes of these sets. These generally have the notation N.X, where X is some set. Also, the

entities used in Appendix A were in "human readable" form, which was often a name. Names are a particularly poor choice of indexing for computers, however. They require more storage than other forms of indexing, they are slow to search, and their variable length causes data storage headaches. Thus, some of these name fields will be replaced with other indexing systems involving numbers. Such indices generally have the notation #.X where X is some set.

2.2.1 Let N.P be the total number of country or territorial names (including aliases). Let N.P_O be the number of distinct geopolitical entity names.

2.2.2 Let N.K be the total number of geoname entries in the database. In the terminology of Appendix A, this is the total number of triples of (geoname, index, geopolitical entities). Let N.K.pref be the total number of distinct places (i.e., the total number of preferred names), and let N.K.alias be the number of secondary nams. Note N.K.pref N.K.alias = N.K.

2.2.3 If i denotes an index to a country, let N.K(i) be the total number of geoname entries for that country. Obviously, there are N.P_O such subtotals, and the sum of them is N.K.

2.2.4 Let N.D be the total number of feature designators.

2.2.5 Let the feature attributes such as population be grouped together into classes. Let N.Att be the total number of attribute classes.

2.2.6 Let N.M be the number of maps that are "known" to the database, i.e., the number of maps for which the map identifier and map limits are held in the sets M and L, respectively.

2.2.7 To each element of P (country name or territorial name), an index #.P will be assigned. These indices will be unique and will range from 1 to N.P inclusive. To each distinct geopolitical entity in P_O, an index #.P_O will be assigned. These indices will be unique and will range from 1 to N.P_O. If a country name is in both P and P_O, its indices #.P and #.P_O need not be identical and, in practice, they normally will differ.

2.2.8 To each map identification entity, two indices #.M.id and #.M sheet will be assigned. The first index identifies the type of product (e.g., JOG) and the second index identifies the individual sheet. Both indices begin at 1 and are assigned sequentially as needed. The total number of distinct pairs (#.M.id, #.M.sheet) is N.M, of course.

2.2.9 To each preferred geoname key, associate an index #.K.pref. These indices will be unique and will range from 1 to N.K.pref.

2.2.10 To each alias that is not a preferred geoname key, associate an index #.K.alias, which will be a unique index with range from 1 to N.K.alias.

2.2.11 To each feature designator, an index #.D will be assigned. These indices will be unique and will range from 1 to N.Att.

2.3 DATA STRUCTURES

A number of tables, lists, etc., are needed for the SDBM. These are described individually in this section. Each structure is, in a sense, a relation between entities in the form of a table. The entities may either be explicit, as data fields in the structure, or implicit, represented by ordering (such as the row or column number of a matrix). In the following descriptions, implicit entities are written with brackets around them, e.g., [#.P_O], and these must always be fields whose range is some set of sequential positive integers, beginning with 1.

Figure 11 shows the linkages between these data structures.

2.3.1 "Set P" Table

Type of Structure. List of variable-length alphanumeric entries.

Size of Structure. The list contains N.P. entries.

Entities of Structure. There is only one type of entry in the list: the country or territorial names in the set P.

Conceptual Functions and Sets Involved with this Structure. This list represents the set P defined in Appendix A, Section 2.2.

Remarks. Since this list has variable length entries, it is accessed using an address list found in the Name Function Table.

2.3.2 Name Function Table

Type of Structure. Table of fixed length entries.

Size of Structure. The table consists of N.P. rows, each containing two explicit entries, and one implicit one.

Entities of Structure. Each row contains:

- [#.P], the index of the name is represented by the number of the row in the table.
- Address of the alphanumeric name in Set P table.

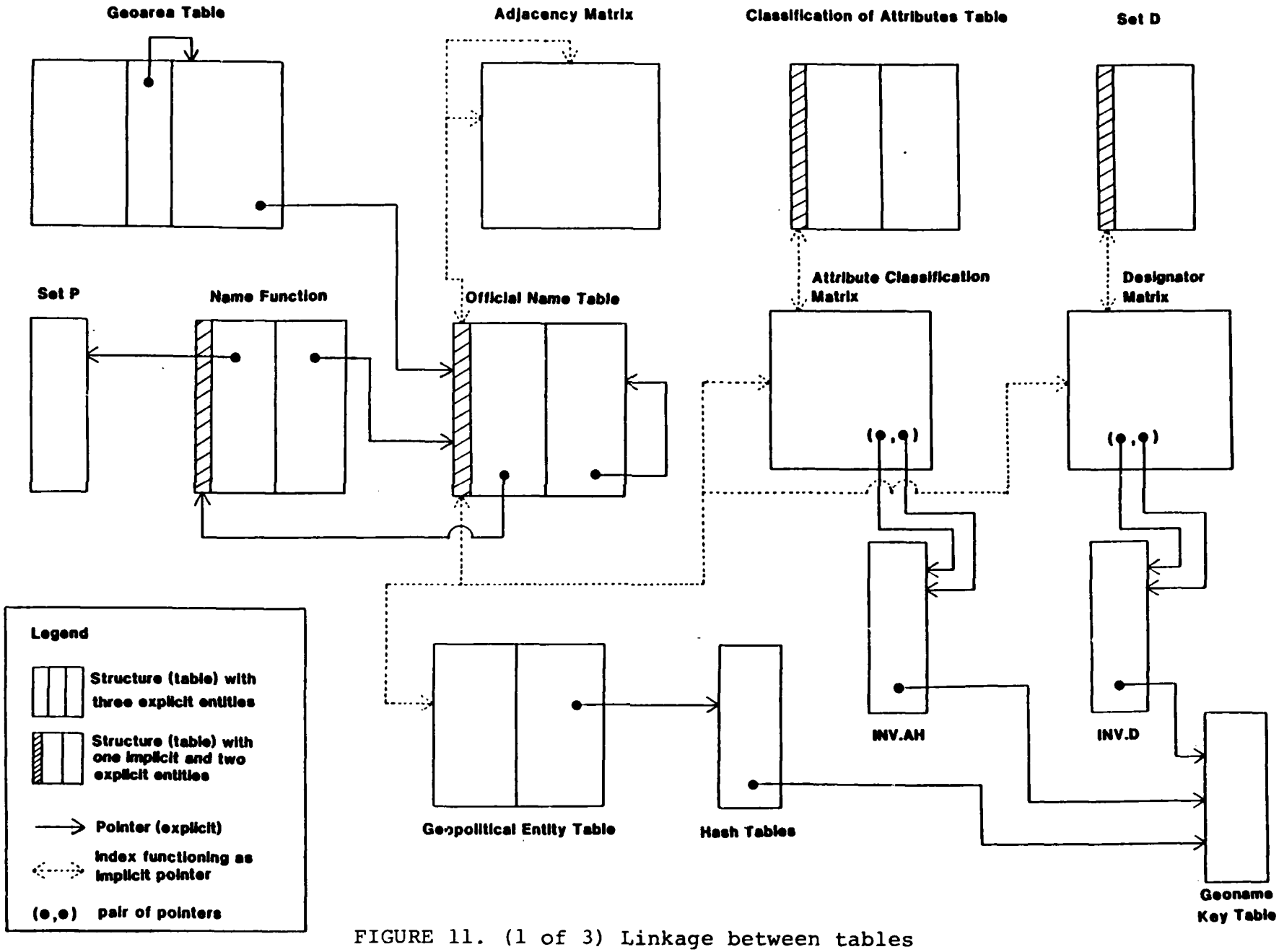


FIGURE 11. (1 of 3) Linkage between tables

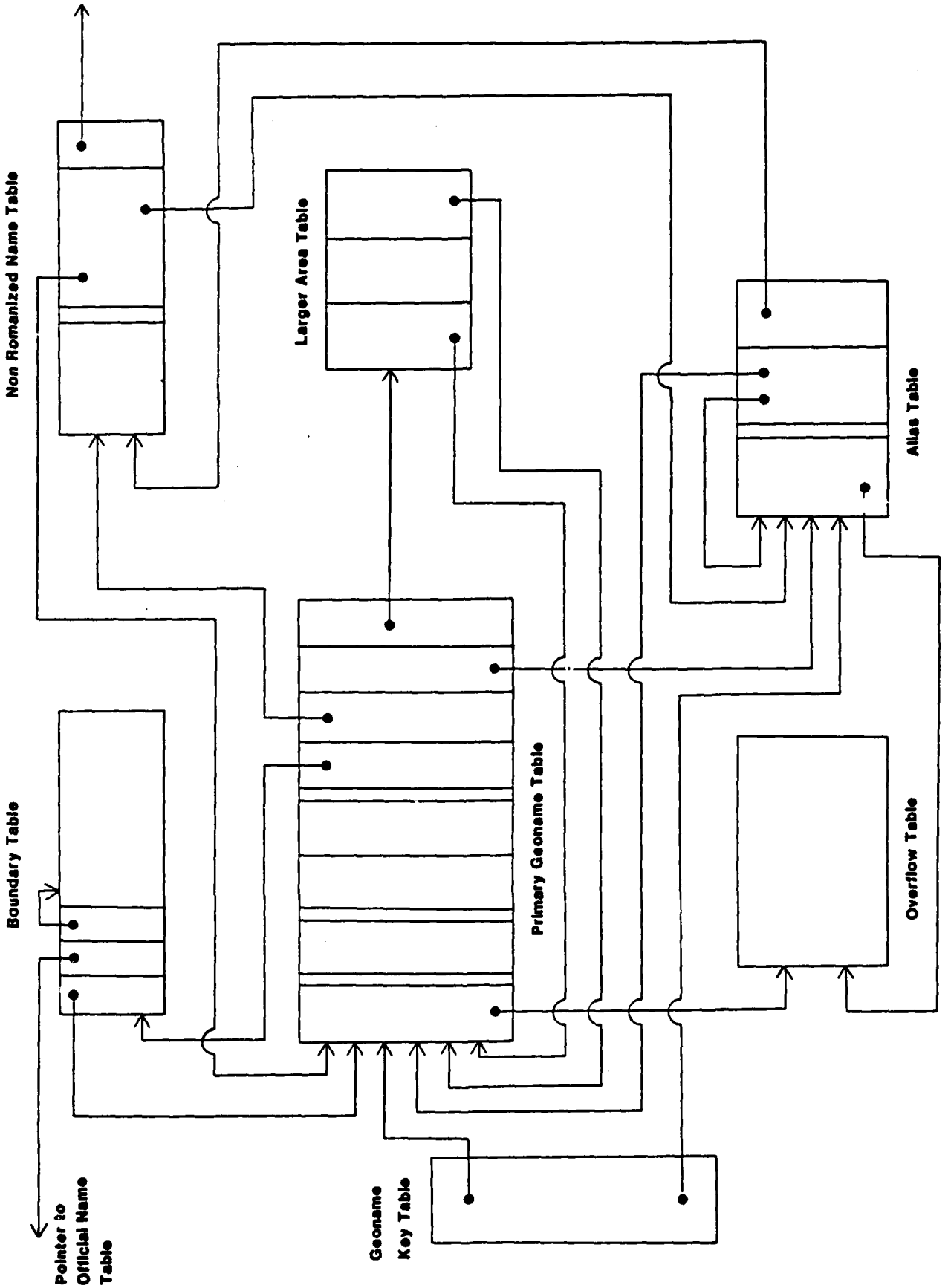


FIGURE 11. (2 of 3) Linkages between tables

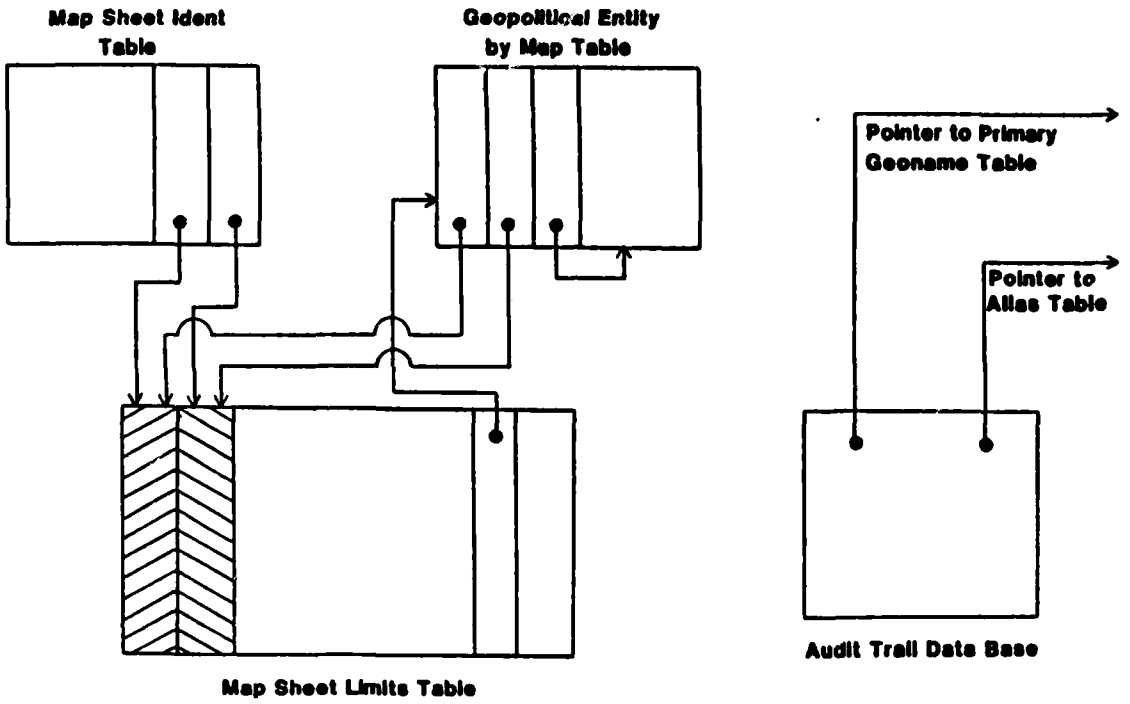


FIGURE 11. (3 of 3) Linking between tables

- P.IND(#.P), the index of the official name corresponding to the name whose index is #.P.

Conceptual Functions and Sets Involved with this Structure. The column P.IND is a tabular representation of the Name Function n, defined in Appendix A.

2.3.3 Official Name Table

Type of Structure. Table of fixed length entries.

Size of Structure. The table consists of $N.P_O$ rows, each containing two explicit entries and one implicit one.

Entities of Structure. Each row contains:

- $[\#.P_O]$, the index of the official name is represented by the number of the row in the table.
- $P_O IND\ (\#.P_O)$, the index in Set P of this official name.
- $T_O(\#.P_O)$, the territorial cognizance function. If $\#P_O$ represents a country, then $T_O(\#.P_O)$ is zero. If $\#.P_O$ represents a territory, then $T_O(\#.P_O)$ is the index of the official name of the country administering the territory.

Conceptual Functions and Sets Involved with this Structure. The column T_O is a tabular representation of the Cognizance Function c, defined in Appendix A.

Remarks. For consistency of the Name Function table and this table, one must have these two relations for all $\#.P_O$ from 1 to $N.P_O$.

$$P.IND(P_O.IND(\#.P_O)) = P\ \#.P_O$$

If $T_O.IND(\#.P_O) = 0$, then

$$T_O.IND)T_O.IND(\#.P_O)) = 0$$

The first relation just means that the pointers between the two tables are consistent. The second means that the function c maps T_O in to C_O.

2.3.4 Adjacency Table

Type of Structure. Matrix.

Size of Structure. The matrix is $N.P_O$ by $N.P_O$.

Entities of Structure. There are three entities: The matrix element itself and two implicit entities, the number of the row and the column.

- [#.P_O], the index of the first geopolitical entity argument in the Adjacency function, is represented by the row number.
- [#.P_O], the index of the second geopolitical entity argument, is represented by the column number.

Conceptual Functions and Sets Involved with this Structure. This represents the Adjacency function Adj of Appendix A.

Remarks. For consistency, this matris must be symmetric, all diagonal elements must be equal, and no off-diagonal element may have the same value as the diagonal elements.

2.3.5 Geoarea Table

Type of Structure. Hierarchy, represented as a table of variable-length lists.

Size of Structure. The number of units in this structure is equal to the number of geoarea names.

Entities of Structure. The structure is divided into units. Each unit consists of:

- Geoarea name: an alphanumeric name.
- List length: the number of geopolitical entities in the following list.
- Geopolitical entity list: the indices (#.P_O) of the geopolitical entities associated with the geoarea name.

Conceptual Functions and Sets Involved with this Structure. This structure contains the set O and represents the relation h of Appendix A.

2.3.6 Geopolitical Entity Table

Type of Structure. Table of fixed-length elements.

Size of Structure. There are N.P_O rows in the table.

Entities of Structure. Each row consists of:

- [#.P_O] the index of the geopolitical entity is represented by the row number.

- Hash constants: a set of integers used in the hash alrithm for the particular geopolitical entry.

- Pointer to portion of hash table containing entries for geopolitical entity #.P_0.

Conceptual Functions and Sets Involved with this Structure. This structure does not encompass any conceptual function, but it does provide information for the immersion of P_0 into K, the subset of $GxAxP_0$ (cf. App. A).

2.3.7 Hash Tables

Type of Structure. This is a large list of pointers, not necessarily dense, which is organized by geopolitical area. Thus, one may alternately think of it as $N.P_0$ individual lists.

Size of Structure. For efficiency, the length of this list should be approximately 1.8*N.K.

Entities of Structure. The only type of entity is the pointers to the Geoname Key Table (see para. 2.3.14). These pointers are grouped by geopolitical entity. Within the area assigned for a geopolitical entity, the pointers are hashed according to the parameters in the Geopolitical Entity Table.

Conceptual Functions and Sets Involved with this Structure. Hashing is an internal database storage technique. As such, it has nothing to do with the Conceptual Database of Appendix A.

2.3.8 "Set D" Table

Type of Structure. List of fixed-length entries.

Size of Structure. The list has N.D entries.

Entities of Structure. There is one explicit entry and one implicit one.

- [#.D], the index to feature designator, is represented by the position in the list.

- Feature Designator: this alphanumeric entry is the code used for the type of feature.

Conceptual Functions and Sets Involved with this Structure. This list is the Set D mentioned in Appendix A.

2.3.9 Classification of Attributes Table

Type of Structure. List of attribute ranges for continuous attributes (e.g., population) or discrete attributes.

Size of Structure. The list consists of N.Att entries, an entry is either a range or a list.

Entities of Structure. Each entry consists of:

- [#.Att], the index to attribute classification, is represented by the position in the list.
- Beginning range of attribute class.
- End range of attribute class.

If there is a list instead of a range, then the last two entities are replaced by the list length and list address.

Conceptual Functions and Sets Involved with this Structure. This table reduces continuous parameters (such as population) to a set of discrete ranges.

2.3.10 Designator Table

Type of Structure. Matrix whose elements are ordered pairs of pointers.

Size of Structure. The matrix is N.P_O by N.D.

Entities of Structure. There are four entities: the matrix element itself (an ordered pair) may be considered as two entities, and there are two implicit entities:

- [#.P_O], the index to geopolitical entity, is represented by the row number.
- [#.D], the index to feature designator, is represented by the column number.

Conceptual Functions and Sets Involved with this Structure. The matrix is a way of looking at a small portion of the relation g of Appendix A.

2.3.11 Attribute Class Table

Type of Structure. Matrix whose elements are ordered pairs of pointers.

Size of Structure. The matrix is N.P_O by N.Att.

Entities of Structure. There are four entities: the matrix element itself (an ordered pair) may be considered as two entities, and there are two implicit entities.

- [#.P_O], the index to geopolitical entity, is represented by the row number.

- [#.Att], the index to class of attribute, is represented by the column number.

- The matrix element is a pair of pointers to the inverted pointer file INV.Att. The pair points to the beginning and end of the data list for the particular geopolitical entity and feature attribute class.

Conceptual Functions and Sets Involved with this Structure. This matrix is a way of looking at a small portion of the relation g of Appendix A.

2.3.12 INV.D. File

Type of Structure. Inverted pointer file by geopolitical entity and feature designator.

Size of Structure. Approximately 1.2*N.K. pointers.

Entities of Structure. The only type of entity is the pointer to the Geoname Key table. For each geopolitical entity and each feature designator, there is a set (possibly empty) of pointers to those entities in the Geoname Key table which possess those two values. Each of these sets of pointers is sorted in ascending order. Since these sets have different lengths, they are located using the pointers from the Feature Designation table.

Conceptual Functions and Sets Involved with this Structure. An inverted pointer file is simply a technique of data accessing: it has no place in the conceptual design of Appendix A.

2.3.13 INV.Att File

Type of Structure. Inverted pointer file by geopolitical entity and feature attribute class.

Size of Structure. Approximately 1.2*N.K.

Entities of Structure. The only type of entity is the pointer to the Geoname Key table. For each geopolitical entity and each attribute class of the feature, there is a set of pointers to those entries in the Geoname Key table that possesses those two values. Each of these sets of pointers is sorted in ascending sequence.

Conceptual Functions and Sets Involved with this Structure. None (cf. para. 2.3.12 above).

2.3.14 Geoname Key Table

Type of Structure. List of pointers, grouped by geopolitical entity.

Size of Structure. The list contains N.K. pointers.

Entities of Structure. The only type of entity is a pointer which may be either to the Primary Geoname file (see para. 2.3.15) or the Alias file (see para. 2.3.16). These entities are grouped by geopolitical entity, and within a geopolitical entity, the pointers are sorted so that they access the geoname keys in alphabetical order. If several geoname keys have the same geoname and geopolitical entity, then their pointers are sorted so that the access of geoname keys is in descending index-of-geoname order (it is important that descending order be used here).

Conceptual Functions and Sets Involved with this Structure. This set of pointers represents the set K, which is the range of the relation k.

Remarks. For each geopolitical entity $\#.P_0$, there are $N.K.(\#.P_0)$ geoname keys. For ease of updates/maintenance, rather than store the geoname keys themselves in this "alphabetical" file, they are stored in a separate file and accessed by pointers.

2.3.15 Primary Geoname File

Type of Structure. Large table of diverse data items. The table is of fixed-length data items; if the geoname is too large for the field, an overflow table is used.

Size of Structure. The table has N.K.pref rows, each with 12 data fields, one of which is implicit.

Entities of Structure. Each row represents a geoname key. Certain information fields must be filled in (e.g., geoname), others may be missing (e.g., Attribute of feature). The entities are:

- Geoname--this is the alphanumeric name, with diacritics. If the geoname key has aliases, this is the preferred name (other geoname keys are in the Alias file). If the name is too long for the field, this is a pointer to the Overflow table (see para. 2.3.18).

- Index Code--this is the Index of the Geoname. If there are more geonames farther down in the Geoname key table with the same Geoname and Geopolitical entity, then this code tells you how many more such names there are.

For most practical applications, it is not necessary to know the actual index, but just whether or not the index value is 1. But for flexibility, the index may be kept, rather than just a binary code.

- Position--this is the (latitude, longitude) pair.
- Source/Accuracy of Position Code--for most records, this would be supplied automatically when the data first entered the base.
- #.D--the index of the feature designator.
- Attribute of feature--the index of the attribute class may not be used here, since it is insufficient.
- Type of Romanization--this would be encoded.
- Pointer to Boundary Table--if boundary information were available, this would point to its location (cf. para. 2.3.19).
- Pointer to first alias (if any)--if the geoname key has aliases, then this points to the first one in the Alias file (cf. para. 2.3.16).
- [#.K.pref]--this is the row number of the table.

Conceptual Functions and Sets Involved with this Structure. This table incorporates much of the relatic.' g, as well as setting the pointers used in implementation of the relations "inc", "~", and p.

Remarks. The geoname keys themselves are divided between this file and the Alias file. This accomplishes two functions.

- It reduces the chance of inconsistent data between aliases (in the terminology of relational databases, this performs a normalization of the database structure). Without this, it is possible that a toponymist might somehow declare two geonames to be aliases, yet some of their data items (e.g., boundary data) may not agree.
- It saves storage space.

2.3.16 Alias File

Type of Structure. Table of fixed-length data items.

Size of Structure. The table has N.K.alias rows, each row has five entities, one of which is implicit.

Entities of Structure. Each row consists of

- Geoname--this is the alphanumeric name, with diacritics, of an alias geoname key. If the name is too long for the field, this field then contains a pointer to the Overflow table (cf. para. 2.3.10).

- Index Code--this is the Index of the Geoname (cf. para. 2.3.15).

- Geoname Pointer--this pointer is to the next alias, if there are more aliases associated with the geoname. If there is only one alias, or if this is the last alias, the pointer is to the preferred geoname in the Primary Geoname file.

- Pointer to non-Romanized name--if the non-Romanized form of this alias name is stored, this points to it (cf. para. 2.3.17).

- [#.K.alias]--this is the row number of the table.

Conceptual Functions and Sets Involved with this Structure. This table represents the relations $\sim$ and p. From the preferred geoname in the Parimary Geoname file, one obtains a pointer to the first alias. This row of the Alias file contains a pointer to the second alias, and so forth. The row corresponding to the last alias contains a pointer to the preferred geoname. This chain represents the equivalence relation $\sim$. It also represents the relation p, for only one of these geonames is in the Primary Geonames file.

2.3.17 Non-Romanized Name Table

Type of Structure. Table of variable length entries.

Size of Structure. Unknown number of rows (obviously less than N.K.), four entities per row.

Entities of Structure. Each non-Romanized name in the database gives rise to one row in this table. Each row contains the entries.

- Non-Romanized Name--this is a variable-length field with the alphanumeric (in whatever character set) name.

- Alphabet Translation Code--this code tells how the non-Latin alphabet was encoded for storage in the database.

- Pointer to Geoname--this pointer is to the Latinized form of the Geoname, which is either in the Primary Geoname file or the Alias file.

- #.P_O--Index of Geopolitical Entity which contains the "place" corresponding to the geoname.

Conceptual Functions and Sets Involved with this Structure. This table represents a small portion of the relation g of Appendix A.

Remarks. The pointer to geoname and the Index of Geopolitical entity are not needed to support the queries listed in subsection E. However, they are needed for some types of queries, which might be desired from the database at a future time.

2.3.18 Overflow File

Type of Structure. List of variable-length entries.

Size of Structure. Approximately 0.02*N.K.

Entities of Structure. The only type of entry is the geoname. This is the list of names that are too long to fit in the fixed-length field in either the Primary Geoname file or the Alias file.

Conceptual Functions and Sets Involved with this Structure. An overflow table is a consequence of fixed-length data fields. As such, it is a consequence of the database implementation, and has nothing to do with the conceptual design.

Remarks. The size of this overflow table depends on both the total number of total number of geonames (i.e., N.K.) and the amount of space allowed for geonames in the fixed length data fields. Too large of a fixed length field means much wasted space in the Primary Geoname file and the Alias file. Too small of a fixed length field means excessive accessing times, since too many names must be placed in the overflow table. We assume that the field length is chosen so that the number of names which do not fit into the fixed field will be beween 1 and 2%.

2.3.19 Boundary Table

Type of Structure. Hierarchical Structure. Each pointer to this table (from the Primary Geoname file) points to a unique unit. Each unit has a substructure.

Size of Structure. One unit per geoname having boundary data. Unit size varies.

Entities of Structure. Each unit consists of the following entities.

- Pointer to geoname--this pointer is to the Primary Geoname file, and is the inverse of the pointer to boundary table found there (cf. para. 2.3.15).

- #.P_O--index to Geopolitical Entity. This is the geopolitical entity which the boundary lies within.

- Number of linear segments needed to describe boundary.

- Linearized boundary--this is a list of positions, i.e. (latitude, longitude), pairs that describe the boundary. The boundary is a closed curve consisting of linear segments with turnpoints at these positions.

- Resolution--this indicates the accuracy of the linearization. Different map scales require different resolutions.

Conceptual Functions and Sets Involved with this Structure. This structure corresponds to the set Y of Appendix A and a small portion of the relation g.

Remarks. The size of this file and its importance will depend on whether very large scale maps (e.g., city maps) are done using this database. The amount of data needed to describe a city boundary on a 1:50,000 map is much less than would be needed for a 1:12,500 map.

2.3.20 Larger Area Table

Type of Structure. Table, with each rows corresponds to a relation of inclusion between areas.

Size of Structure. Number of rows is twice the number of valid pairs for the relation "inc". Each row has three entities.

Entities of Structure. The Primary Geonames file has pointers into this table. Each pointer is to the beginning of some row. The data associated with that Primary Geoname entry is at least the one row of this table. If the Inverse pointer is the same on the next m rows, then these m rows all belong to the same entity of the Primary Geonames file. This table has three entities:

- Inverse Pointer--this points back to the entry in the Primary Geoname file, which contained the pointer to the Larger Area table. If several successive rows have the same inverse pointer, it means that they all are to be treated together, for they all apply to the same geoname.

- Larger/Smaller Designator--this is a binary code, which we may write as < or >, depending if this entry is giving a larger or smaller area (see following subparagraph).

- Larger/Smaller Areas--this is the pointer to the geoname entry in the Primary Geoname file, which either includes or is included in the subject geoname area.

Conceptual Functions and Sets Involved with this Structure. For convenience, let a and b be two geoname keys such that the pair (a,b,) is in the range of the relation "inc" (i.e., the place associated with geoname key a is part of the place associated with geoname key b. Then there are two rows in the Larger Area table which reflect this relation. One row is the triple (a, <, b) and the other row is (b, >, a). The larger area table is grouped into subsets such that the first entitity (i.e., the Inverse pointer) is the same within each subset.

Remarks. A geoname key may be part of several larger entities, and these need not be ordered. Thus, the need for multiple rows in this table corresponding to one geoname key.

2.3.21 Map Sheet Ident Table

Type of Structure. Table of fixed length items.

Size of Structure. The table has N.M. rows and three entities per row.

Entities of Structure. For each row, all these data fields must exist.

- Map Sheet Identification--this is the alphanumeric code by which the map is known.

- #.M.id--the index to map sheet.

- #M.sheet--the index to map sheet.

Conceptual Functions and Sets Involved with this Structure. The map sheet identification column is the set M of Appendix A.

Remarks. The table is sorted by Map Sheet Identification Code.

2.3.22 Map Sheet Limits

Type of Structure. Table of fixed length items. Alternately, one may consider this as a three-dimensional array,

where #.M.id and #.M.sheet replace the row number by a "double index."

Size of Structure. The table has N.M. rows, each with three explicit entities and two implicit entities.

Entities of Structure:

- [#.M.id]--index to type of map.
- [#.M.sheet]--index to sheet.
- Map Sheet Limits--a series of (latitude, longitude) pairs.
- Pointer to list of geopolitical entities in the map region (cf. para. 2.3.23).
- Type of map projection--a code to indicate the projection used.

Conceptual Functions and Sets Involved with this Structure. This table contains the set L, and by its arrangement, serves as the function m.

Remarks. Although the number of sheets differs between various map products, it is trivial to convert a double index (#.M.id, #.M.sheet) into a single index whose range is 1 to N.M.

2.2.23 Geopolitical Entity by Map Table

Type of Structure. Variable length lists.

Size of Structure. The table has N.M. rows and a variable number of entries per row.

Entities of Structure

- #.M.id--indicates the beginning of a row in the variable length list.
- #.M.sheet--also indicates the beginning of a row in the variable length list.
- Number of entities in list--the number of items in Section 2.3.23.
- #.P.$_0$--list of geopolitical entities which appear (in whole or in part) on the map sheet.

Conceptual Functions and Sets Involved with this Structure. None.

Remarks. Rather than search the whole file on position, this allows one to restrict the search to those countries that appear on the map product.

2.3.24 JOG Sheet Number Function

Type of Structure. This is not a table, but is a function that uses various small tables. The function is called using two arguments (latitude, longitude) and returns an alphanumeric code.

Size of Structure. This depends on how the function is computed. Using a 15' grid structure worldwide gives 259,000 squares, but this is not an effective way to handle the function. Using a list of longitudes by latitude gives under 1000 entries.

2.3.25 UTM Function

Type of Structure. This is a function, similar to the JOG function (para. 2.3.24). It returns the Universal Transverse Mercator Grid zone.

2.3.26 Other Database Structures

The above list of structures covers the principal tables used for storing the data. There are a number of other tables that the DBMS would use. These are not significant for timing estimates, so they will not be considered in any detail. This set includes:

Update Journal Database. Given the complexity of the database, and the multiple user environment, the safest approach to handling updates is to collect the updates in a separate file, and later, when no users are accessing the database, perform all the updates. Since the average data item in this base does not change very often, such an updating procedure is quite reasonable. The Update Journal Database is used to hold the updates prior to entry into the base.

Audit Trail Database. For the toponymist, it may be important on occasion to determine where certain information came from. The Audit Trail Database allows one to determine who put a data item in the base, when it was put in the base and the source of the information. This database, for obvious reasons, would be very large (33N.K.entries) and rather slow to access, but it would not be accessed as heavily as the database structures listed above. This database may have an overflow area for situations when the toponymist wishes to enter more than one reference, but multiple entries would be the exception rather than the rule.

Analyst Geoname Files. These are temporary files created by queries extracting data from the primary data files. The users

would then be able to review the files and edit them prior to sending the file on to the next process (i.e., either printing for a gazetteer or names placement for a map product). These working files would be protected so that the analyst who created the file could control the access to it.

Cartographer's Feedback File. There will be many users of this database, but the ability to modify the base will be restricted to a subset of users, such as the toponymists. Other users (e.g., cartographers) may modify their analyst geonames files if they do not like certain data items extracted from the base, but could not alter the database itself. When such a user finds a data item that he believes needs correction, he may enter it in the Cartographer's Feedback File as well as his Analyst Geoname File. The toponymist may periodically review this file, either adopting or rejecting the suggested changes to the database.

MIS File. Management Information Statistics (MIS) should be collected by every DBMS, not only to determine the amount of use of the base, but to determine the types of use. A data accessing strategy that is optimal for a particular mix of queries may be poor several years later if the mix of queries has dramatically changed.

2.4 OPERATION OF DATABASE

The database structure described above will support the types of queries given in Section 2.5 of this report. For searching on geonames, the Geopolitical Entity table and the Hash tables are used to access the Geoname Key table and, thus, the Primary Geoname file, Alias file, Boundary table, Larger Area table, and Non-Romanized Name table. For gazetteers, the user will query on feature designator and feature attribute, and this is handled by "And-ing" the pointers from the INV.D and INV.Att tables, then building subfiles of pointers for each logical clause, and using a sorted merge on the subfiles to handle the "Or-ing" of query clauses. The resulting set of pointers is then used to extract the data in alphabetical order from the Primary Geoname file and Alias file. UTM and JOG sheet designators are computed using internal functions. For queries in support of map products, the Geopolitical Entity by Map table is used to determine which countries are involved in the data request. For each of these countries, the inverted pointer tables are used to get to the Geoname Key table, which then accesses the various tables containing the requested data. The Map Sheet Limit table is used to limit the extract from each country to just those geonames which are located within the map sheet limits. The output is a subfile for the user which is then accessed, by appropriate applications programs, to edit as needed. Figure 12 shows the data flow.

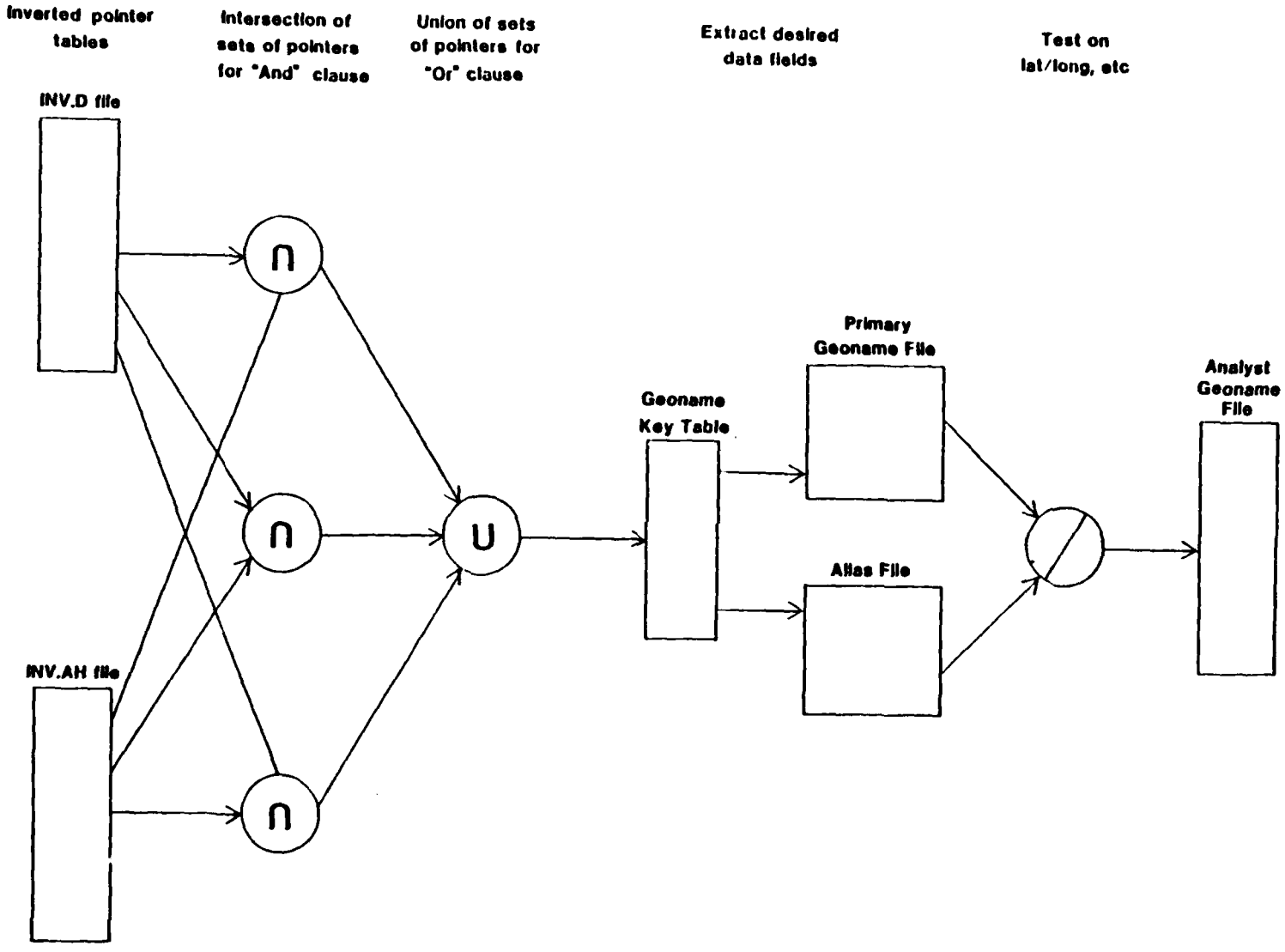


FIGURE 12. Data flow for typical query (query #1)

2.5 DISC ACCESSES

Assume the smaller files are kept in core and the larger ones are kept on disc. Specifically, assume the Adjacency table, INV.D file, INV.Att file, Geoname Key table, Primary Geoname file, Alias file, Non-Romanized Name table, Overflow file, Boundary table, Larger Area table, Map Sheet Ident table, Map Sheet Limits table, Audit Trail database, and Analyst Geonames files are on disc. Each of the queries listed in Chapter 2, Section E of this report can be analyzed in terms of disc accesses needed.

Estimates of Variables

To obtain numbers for disc accesses, it is necessary to assign numeric values to variables such as N.K., and to specify some numbers for the queries. Thus, we assume some typical values:

- Number of Geoname Keys, N.K. = 6×10^7.
- Number of Geopolitical Entities, $N.P_O = 1.5 \times 10^2$.
- Number of Geoname Keys in an average country, $N.K.(\#.P_O) = 4 \times 10^5$.
- Number of Countries Adjacent to an Average Country = 3.
- Number of territories administered by an average country = 1.
- Number of countries on a typical JOG sheet = 1.
- Number of names in a country equal to a typical given name = 2.
- Number of names in a country similar to a typical given name = 6.
- Number of non-Romanized names in the database = 10^6.
- Number of cities with boundary information = 5×10^4.
- Number of aliases, N.K.alias = 1.5×10^7.
- Number of geonames in an average gazetteer = 3×10^4.
- Number of geopolitical entities in an average geoarea = 6.
- Number of updates on a magnetic tape (e.g., from digitizing maps = 10^4.
- Number of geonames on average map = 3×10^3.

- Number of country names, counting aliases = 5×10^2.
- Number of geonames from a particular reference = 5×10^5.
- Number of countries for which a particular type of Romanization of names is used = 1.

2. Average Number of Disc Accesses Per Query

For each query listed in Section 2.5 of this report, the number of accesses is estimated. Since one-digit precision is adequate for this, only the major contributors in each case are counted. For example, in the first query, the DBMS would access the INV.D and INV.Att files for about 4×10^3 pointers, yet because the pointers are grouped together, very few disc reads are needed (six disc reads should be adequate). Using the figures from paragraph 1 (preceding) gives the following table.

QUERY	NUMBER OF ACCESSES
1	7×10^3
2	6×10^4
3	1×10^1
4	4×10^1
5	8×10^1
6	5×10^2
7	7×10^4
8	6×10^4
9	5×10^1
10	2×10^5
11	1×10^6
12	8×10^4
13	1×10^1
14	1×10^1
15	5×10^4

It is seen that disc access vary widely: from ten to a million for different types of queries. The accesses needed for the commonly used queries, however, are low (in terms of the amount of information requested).

2.6 Timing

For fiing estimates of the SDBM, one may assume that the I/O time is greater than the CPU time. This is reasonable because there are few in-core functions (such as sorting) which require a "long" time. With machine cycle times measured in nanoseconds and disc accesses in milliseconds, the timing estimates may be made

using I/O times only. Conversion from number of accesses to seconds depends upon the equipment used. Taking a rather arbitrary figure of 20 msec per access shows that Queries 3, 4, 5, 6, 9, 13, and 14 are measured in seconds (i.e., 0.2 seconds to 10 seconds); Query 1 requires about 2 minutes; Queries 2, 7, 8, 12, and 15 require about 20 minutes; Query 10 requires about an hour; and Query 11 requires 6 hours. Obviously, using another value for disc access time will give much different times; however, this indicates the relative speeds of different query types.

2.7 SIZE

In Subsection 2.6 it was implicitly assumed that all the files associated with the Geonames Database would fit on on-line disc units. Using the sizes given in Section 2.7, one obtains the sizes for the larger files given in the following table.

TABLE	SIZE (10^6) BYTES
Hash Table	364
INV.D	243
INV.Att	243
Geoname Key	203
Primary Geoname	1642
Alias	324
Non-Romanized Name	20
Overflow	48
Boundary	3
Larger Area	1
Audit Trail	1155

Excluding the Audit Trail Database (which is not used for most queries) gives a total size of 3.09×10^9 bytes. With the Audit Trail Database, the total is 4.25×10^9 bytes. In many databases, Audit Trail information is kept off-line, on slower media, such as magnetic tape. In such a case, the timing estimate for query 11 given in Section 2.6 would be longer, for many tapes would have to be read.

DISTRIBUTION LIST

Department of the Navy
Asst Secretary of the Navy
(Research Engineering &
System)
Washington DC 20350 (1)

Project Manager
ASW Systems Project (PM-4)
Department of the Navy
Washington DC 20360 (1)

Department of the Navy
Chief of Naval Material
Washington DC 20360 (1)

Department of the Navy
Chief of Naval Operations
ATTN: OP 951
Washington DC 20350 (1)

Department of the Navy
Chief of Naval Operations
ATTN: OP 952
Washington DC 20350 (1)

Department of the Navy
Chief of Naval Operations
ATTN: OP 980
Washington DC 20350 (1)

Director
Defense Technology Info
Cen
Cameron Station
Alexandria, VA 22314 (12)

Department of the Navy
Director of Navy
Laboratories
Rm 1062 Crystal Plaza
Bldg 5
Washington DC 20360 (1)

Commander
DWTaylor Naval Ship
R & D Cen
Bethesda MD 20084 (1)

Commanding Officer
Fleet Numerical Ocean Cen
Monterey CA 93940 (1)

Commander
Naval Air Development Cen
Warminster PA 18974 (1)

Commander
Naval Air Systems Command
Headquarters
Washington DC 20361 (1)

Commanding Officer
Naval Coastal Systems Cen
Panama City FL 32407 (1)

Commander
Naval Electronic Sys Com
Headquarters
Washington DC 20360 (1)

Commanding Officer
Naval Environmental
Prediction
Research Facility
Monterey CA 93940 (1)

Commander
Naval Facilities Eng Com
Headquarters
200 Stovall St.
Alexandria VA 22332 (1)

Commanding Officer
Naval Oceanographic Office
NSTL Station
Bay St. Louis, MS 39522 (1)

Commander
Naval Oceanography Command
NSTL Station MS 39522
Bay St. Louis, MS 39522 (1)

Director, Liaison Office
Naval Ocean R&D Activity
800 N. Quincy Street
502 Ballston Tower #1
Arlington VA 22217 (1)

Commander
Naval Ocean Systems Center
San Diego CA 92152 (1)

Superintendent
Naval Postgraduate School
Monterey CA 93940 (1)

Commanding Officer
Naval Research Laboratory
Washington DC 20375 (1)

Commander
Naval Sea System Command
Headquarters
Washington DC 20362 (1)

Commander
Naval Surface Weapons Cen
Dahlgren VA 22448 (1)

Commanding Officer
Naval Underwater Systems
Cen
ATTN: New London Lab
Newport RI 02840 (1)

Department of the Navy
Office of Naval Research
ATTN: Code 102
800 N. Quincy St.
Arlington VA 22217 (1)

Officer in Charge
Office of Naval Research
Detachment, Boston
Barnes Building
495 Summer St.
Boston, MA 02210 (1)

Commanding Officer
ONR Branch Office LONDON
Box 39
FPO New York 09510 (1)

Commanding Officer
ONR Western Regional Ofs
1030 E. Green Street
Pasadena CA 91106 (1)

Director
Scripps Inst of Oceanography
Univ of Southern California
La Jolla CA 92093 (1)

Working Collection
Texas A & M University
Department of Oceanography
College Station, TX 77843 (1)

President
Woods Hole Oceanographic Inst
Woods Hole, MA 20543 (1)

Director
Defense Mapping Agency
Washington, DC 20305 (5)

Director
Defense Mapping Agency
Hydrographic/Topographic Cen
6500 Brooke Lane
Washington, DC 20315 (8)

Director
Defense Mapping Agency
Aerospace Cen
St. Louis Air Force Station,
MO 63118 (5)

Director
Defense Mapping AGency
Special Program Office of
Exploitation and
Modernization
8301 Greensboro Drive,
Suite 1100
McLean, VA 22102 (2)

Commanding Officer
Naval Ocean R & D Activity
NSTL Station, MS 39529 (4)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NORDA Technical Note 189	2. GOVT ACCESSION NO. AD-A126 241	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* Advanced Type Placement and Geonames Database: Comprehensive Coordination Plan		5. TYPE OF REPORT & PERIOD COVERED Final
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) A.E. Barnes and E.C. Gough (both of PSI) R.M. Brown and A. Zied (both of NORDA)		8. CONTRACT OR GRANT NUMBER(s) PE63701B
9. PERFORMING ORGANIZATION NAME AND ADDRESS Planning Systems Inc. Slidell, La. 70458		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS PE63701B
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Ocean Research and Development Activity Pattern Analysis Branch, Code 371 NSTL Station, Miss., 39529		12. REPORT DATE January 1983 13. NUMBER OF PAGES 131
14. MONITORING AGENCY NAME & ADDRESS*(if different from Controlling Office)*		15. SECURITY CLASS. *(of this report)* Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Distribution unlimited

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Alphanumeric Data Entry System	DMA/HTC
Geonames Database System	OMA/AC
Type Composition and Placement System	Automated Cartography System
Database Management System	
Advanced Symbol Processing	

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

This paper presents a preliminary comprehensive coordination plan for the technical development and integration of an automated names database capture and management of an overall automated cartographic system, for the Defense Mapping Agency. It broadly covers the technical issues associated with system and associated subsystems' functional requirements, inter-subsystem interaction common technologices in hardware, software, database, and artificial intelligence. The 3-phase R&D cycles for each and all subsystems are also outlined.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*


4-8
DTI