(NASA-CR-184536) GENERAL SPECIFICATIONS FOR THE DEVELOPMENT OF A PC-BASED SIMULATOR OF THE NASA RECON SYSTEM Final Report, 1 Jul. 1985 - 31 Dec. (University of Southwestern Louisiana, Lafayette. Center for Advanced G3/82 Unclas 0183574 N89-14976

| N A S A | | N A S A |

U S L / D B M S N A S A / P C R & D

W O R K I N G P A P E R S E R I E S

Report Number

DBMS.NASA/PC R&D-4

The USL/DBMS NASA/PC R&D Working Paper Series contains a collection of formal and informal reports representing results of PC-based research and development activities being conducted by the Computer Science Department of the University of Southwestern Louisiana pursuant to the specifications of National Aeronautics and Space Administration Contract Number NASW-3846.

For more information, contact:

Wayne D. Dominick

Editor
USL/DBMS NASA/PC R&D Working Paper Series
Computer Science Department
University of Southwestern Louisiana
P. O. Box 44330
Lafayette, Louisiana 70504
(318) 231-6308

| N A S A |

GENERAL SPECIFICATIONS
FOR THE DEVELOPMENT OF
A PC-BASED SIMULATOR OF THE NASA RECON SYSTEM

Spiros Triantafyllopoulos

The University of Southwestern Louisiana
Computer Science Department
Lafayette, Louisiana

August 2, 1984

GENERAL SPACIFICATIONS
FOR THE DEVELOPMENT OF
A PC-BASED SIMULATOR OF THE NASA RECON SYSTEM

ABSTRACT

This document will describe the general specifications for a NASA/RECON simulator targeted for a Personal Computer. Information system simulation provides several advantages during system training, since it allows extensive use of the system without the typically high cost overhead of accessing large-scale, remote systems and also can provide a better user interface and assistance. This means less cost for the end-user, faster and more efficient training, all resulting in increased user productivity.

TABLE OF CONTENTS

GENERAL SPECIFICATIONS
FOR THE DEVELOPMENT OF
A PC-BASED SIMULATOR FOR THE NASA RECON SYSTEM

This is a working document. If you have any comments please contact the PC Simulator R&D team. Versions of this document will be circulated on a regular basis until finalization. The design and implementation of the simulator will require many changes suggested by individual reviewers; your feedback will be appreciated.

I. INTRODUCTION

This document will define the specifications for the IBM PC/XT-based NASA/RECON System Simulator. Input to this document has been from the USL NASA Contract Team and, up to this point, not from the NASA/RECON designers.

An information system simulator is defined as a program that behaves like a certain information system. The purpose of such a simulation is multiple: prototyping, CAI, reducing online costs, etc. In the NASA/RECON simulator, the main motive has been the development of a simulator as an educational tool to allow

instruction without paying the high ($50.00/hour or more) cost of long distance telephone charges, TELENET charges, and online host system charges. Also, CAI could be embedded in the implementation so that teaching a potential user can be highly automated and thus simplified.

In simulating an information system as large and as powerful as the NASA/RECON system, it should be of top priority to decide very early on the features that should be included as well as the features to be excluded, if any. The original NASA/RECON system is written in PL/IG and runs on an IBM 4341 computer (NASA/RECON Users Manual). Trying to simulate it on an IBM PC/XT would need very careful planning concerning what is to be included and what is to be excluded.

In such cases, where a program has to be migrated or simulated from one computer to another, it is obvious that the techniques applied in the original implementation may not be applicable to implement the model. In the following pages, some details will be highlighted.

II. GENERAL FILE DESIGN

The NASA/RECON system is a bibliographical Information Storage and Retrieval System. The system is based on a thesaurus, a list of words and related terms on which searches can be made. The thesaurus is the main part as it provides a basis for subsequent indexing of entries according to standard conventions and terms. In reality, the NASA/RECON thesaurus contains several thousand entries and approximately 2 million records in all file collections. It is obvious that the PC-based simulator can not contain more than a fraction of these records. Still, the thesaurus is needed as it is the main facility for indexing and retrieving documents and the PC-based thesaurus should be of a reasonable size to support simulating, if not replicating reality.

Record design should follow the NASA/RECON standard, with fields for the following items (as a minimum/first requirement):

* ACC: accession number
FST: financial support type
ISS: issue number
JAP: page numbers
* CAT: subject category
* RN: report number
* CN: contract number
* PDI: publication date
PAG: number of pages
LNG: document language
* UTI: unclassified title
* TLS: title supplement
* AU: personal author
* PAA: personal author affiliate
* CO: corporate source
* PUB: publisher

Note: in all entries, "*" denotes a directly searchable term which is indexed (keyed).

In addition, the following items must be included for text searches in the data base:

* MJS: major terms
* MNR: minor terms — both form ST (subject terms)
  ABS: abstract
  SUM: summary
* ANI: analytical terms
* ANN: analytical notes

Other terms and fields can be added as needed, if implementation constraints allow such expansions. However, file design depends on the file management system to be used.

File structures like the above would need disk space on the order of 1300-1500 characters per record so that all items are searchable, excluding keys. The file system to be acquired will support the Indexed Sequential Access Method (ISAM), based on multiple user-defined keys. Thus, an inverted file structure will be used, with all searchable items keyed, together with a record number to identify the entire record. If the number of records is kept small, then we can use sequential searches as well to avoid multiple indices and the associated space requirements as well.

On double sided, double density diskettes, up to 150 records can be included. A second diskette will contain the program, ISAM file management, user stored queries, scratch files and other supporting software. The thesaurus and dictionary/support files

will be included in the data disk. Thus, a two-disk based approach is neccesary if simulation is to be realistic. Program size should be around 128 to 256 Kbytes, possibly by chaining some parts of the program on and off the program disk. Program data size has to be limited if effective use of the system and compatability with others are desired.

An estimation of the disk space requirements (tentative) is as follows:

240K: data base (150 records x 1600 bytes/record)
45K : index lists (150 records x 15 items x 20 bytes/item)
40K : Thesaurus ((20 bytes/term + 20 ptrs) x 1000 terms)
16K : Linking (see next page)
24K : Free (workspace, etc.)

365K: Total disk space, double-sided, double-density

A brief explanation on the storage allocation estimates follows:

a. Data Base: The average NASA/RECON record fits in approximately one screen of a Cathod-Ray Terminal, which is 24 x 80. By limiting line length to approximately 60, as RECON does, we arrive at a length of 1600 bytes/record

approximately as an average.

b. Index lists: Assuming an indexed sequential file, then with 15 searchable indexed terms, allocating 20 bytes/record (this would be 17-18 for the term and 1-2 for the record number) makes a total of 45K. This depends on the characteristics of the ISAM file management package to be used; however, other similar packages have approximately the same space requirements.

c. Thesaurus: Assuming a maximum number of 1000 entries, including 20 pointers (max) for each term, 4 each to narrower, broader, related and used for, and 20 more bytes for the term and the associated record number, we need 40 bytes/record for a total of 40K bytes.

d. Linking list: Used for the thesaurus, it is an inverted list of thesaurus terms and record numbers, indexed on record numbers. This facilitates thesaurus cross-references and minimizes disk references. The space needed is 20 bytes/record for 1000 records giving a total of 20K bytes.

e. Free: This area can be used for any purpose that might arise, but as read-only. Temporary areas will reside on the program disk. This area will also cover unexpected changes in the design or other changes dictated by the design/implementation

cycle.

The design of the data file structures was mainly influenced by the need for fast access to data. Regular inverted file structure is not adequate in the case that the entire record needs to be displayed. Thus, in order to increase speed of access, the lists and the records themselves carry the information redundandtly.

As it stands, the original design does not include any Computer Aided Instruction improvements, with the exception of more specific error messages. If CAI is to be incorporated, then one more disk would be required for the CAI material.

III. NASA/RECON COMMAND LANGUAGE

The NASA/RECON system features a quite simple instruction format. Abbreviations for the commands can be used. All parameters follow the command after either a space or a slash ("/"). Some commands, while they are functional on NASA/RECON, will be reproduced in the simulator just for "simulation" purposes without doing anything functional. Others will perform tasks parallel to the corresponding mainframe NASA/RECON.

A list of no-operation commands and their corresponding actions follows:

BEGIN starts a search session; the user is asked to select a file collection and is placed in the RECON search environment. Then the RECON message is printed and the user commences the search. BEGIN can also function as a restart function to allow clean-up of previous searches and terminate current session, although END SEARCH should be used instead.

CANCEL will print a message but will not cancel a queued-up query, since there is only one user.

COMMAND STATUS will also function like CANCEL since no queing-up

of queries is permitted.

END SEARCH will terminate the current search session and display data relevant to the search, and ask the user some information.

END SEARCH BYPASS will terminate the session like END SEARCH but will not present the questionnaire.

HELP will provide some help to the user. The original NASA/RECON text can be used, or more explanatory text.

NEWS will display a login-like message, similar to RECON's.

ORDER will display a message similar to the one displayed when a document is ordered.

PAGE can count the number of lines and stop every 23 or so, and wait for response. It can be difficult to implement without the extra overhead.

PRINT will function like the ORDER, informing the user of his/her command. Again, output should be similar to NASA/RECON output.

SIGNOFF will terminate the session immediately, and return the user to operating system level in contrast with END SEARCH and END SEARCH BYPASS which leave the user in the simulator.

SIGNON will be used to start a user session and allow the user to enter into the RECON system simulator. The user has to enter USERID.

CURRENT will print all information about a user session, and let the user continue his/her session.

BEGIN BYPASS will start a search session without asking for user data.

NASA/RECON functional commands that are executed and interact with the database are as follows:

EXPAND will expand a term from the system dictionary. In practice, the word will be looked up and all terms that have similar spellings are printed along with the numbers of occurrences, like the real RECON system. This can be performed by storing words (terms) and occurrences in the same file, indexed by term in ascending order.

SELECT will select a set displayed by the EXPAND instruction. Linked lists can be used with a storage pool where the sets are represented by numbers (keys). When a set is to be displayed or otherwise manipulated, the linked list will recall the numbers (keys) and the ISAM will read in the

appropriate sets. Fields will be user specified.

The SELECT command has various forms:

TEXT search can be performed on user-specified words and phrases. Sequential searches will be performed unless the terms requested are indexed. Sequential searches will be performed in the case of abstracts, to avoid excessive storage for files.

RANGE search can be performed by using upper and lower values in the indexes that the terms are sorted upon. This can be performed by reading directly the lower bound and from then sequentially until the upper bound. Open-ended searches can also be supported in a similar manner.

ROOT search will have to be performed sequentially from the firstrecord in the range until the last one in the range. It does not seem to present major difficulties.

COMBINE will do the basic Boolean manipulation functions (AND, OR, NOT) plus combinations. For the simulator, a set expression interpreter will be required to perform these three operations on sets: AND is Intersection, OR is Union, NOT is set difference.

DISPLAY will display user-selected sets. The only complexity involved is user-defined formats. The standard specified formats can be used, allowing limited user-defined formats from a pre-defined subset.

FREQUENCY will definetely be complex enough to be placed in a deferred priority list. It can be implemented by keeping track of all MAJ and MIN terms, sorting the resulting list for a given set and producing frequencies of the resulting array (or file). The problem will be the time required to execute this instruction.

KEEP will just make a copy of the set requested, in the temporary set 99. It does not seem difficult, but depends on the complexity of the list manager.

LIMIT will remove items from the user specified set based on user specifications. It involves little more than set item deletion. LIMIT ALL will do this for all sets in the current process. LIMIT RELEASE will restore the sets into their previous state. File dumping every time a LIMIT is issued seems to be a reasonable implementation method. When a LIMIT RELEASE is issued, sets are reloaded from the user file(s) in the original condition. Space considerations should be addresed since keeping lists in file(s) would take up fair

amounts of storage.

SORT will sort a given set based on a user-defined key. The time required for execution can be a problem but otherwise it can be implemented with no major difficulties. Sort can be performed on certain keys only, in which case the program can be quite simple.

SEARCH will combine selection expressions. A reasonable way of implementing it would be to pass the appropriate arguments to the EXPAND, SELECT and COMBINE routines (provided that they are set up for such tasks) and then finally come up with one set. Again time required for execution will be the only major problem to be identified. A minor detail would be NASA/RECON's special requirements for blanks, stars ("*") and quotes, since the allowable characters are not common among terms searched.

SPECIFY FORMAT willallow the user to change the default format(s) used. It will relate to the DISPLAY and TYPE commands, since it will redefine output formats. A way of choosing different formats can be found easily since "C" allows formats to be formed at run time, and control characters (new-lines, line-feeds, and various format specifications). This can be quite complex, however if many items must be displayed. RECON

provides some standard formats (like /4, /2, etc) that can be used to start with and then allow user-defined formats later on.

TYPE will display on the screen just a certain subset of the entire record for the set specified. TYPE can be handled if the SPECIFY FORMAT provides for the formats allowed (in TYPE). Thus, a call to TYPE can be made as a call to SPECIFY FORMAT to set the format, a call to DISPLAY and a call to SPECIFY FORMAT again to restore the original format.

RELEASE will re-initiate the set node pool and will also erase any file(s) created by LIMIT commands. It should be used with all the search initiation and termination procedures and commands to ensure proper initialization.

IV. PROGRAM DESIGN ISSUES

The language that will be used for the implementation of the NASA/RECON simulator on the IBM PC/XT will be "C". In addition, an Indexed Sequential Access Method (ISAM) file management system will have to be obtained, since "C" does not provide for such support.

The file management system will be used for the support of all the inverted files to be mantained. Since all non-free-text searchable terms will be stored in inverted files, file design should be very carefull to avoid disk space overflow. The file management system comes in source code and the rights (licence) for the incorporation of it in user software. So the file management object code should also go on the simulator diskette.

System portability will have to be considered not only at the PC level but at the mini and mainframe level as well. The "C" compiler to be used is compatible with Unix (R) version 7 "C". Tests are being performed to determine the compatability between the PC/XT "C" and the "C" available on the VAX/VMS system. The primary goal is, of course, the PC/XT simulator, but the popularity of both Unix (R) and VAX/VMS can be a major factor if a production version of the simulator is made.

Stored queries will be a special issue to be considered. Most of the programs that are needed to support a stored query system are the same as the ones needed for the interactive interface. The query editor is not complicated and can be easily implemented.

Stored queries are advanced features which are not likely to be used by inexperienced simulator users. To provide, however, for a realistic environment that fully replicates RECON, and allows experienced users to practice more with the system, the stored search features must be implemented. Individual user groups should be allowed space on the data disk for their stored searches, and in general the whole stored search environment will be replicated. The problem with stored searches is that the entire user interface (RECON command level) will have to be tailored in batch/interactive environment. Such a command level would allow inputs from both file and terminal, and be able to store and edit queries as required. Again, complexity and efficiency are the main factors, but the result will be a much better simulation of the RECON system.

System logging in can be performed in a variety of ways: just enter RECON command level and BEGIN SEARCH, or simulate all the TELENET/NASA Host login procedures (!) for a more "realistic" environment. The same is also true for quiting the system and returning to operating system level.

Error recovery is also a major item to be considered. In case of error, NASA/RECON displays a cryptic message and nothing else, expecting the user to have a manual and see the error from the error code. This is definetely not going to help the users of the simulator. The proposed design can have two levels: a beginning level where all messages are self-explanatory with examples etc, and an advanced level where the same message structure as the RECON system is followed. Some messages can be extremely confusing (even to M.Sc. CMPS Students) and the help facility does not do much to explain. The help text can again be arranged in two levels as well. Then, the user can be asked if he wants the beginner or advanced level and proceed with the system.

User monitoring can help improve the simulator (and the NASA/RECON system as well). A facility similar to MADAM's can be incorporated. The number of queries, errors, and other measurable items can be recorded and used for both simulator and user evaluation. After a session the user can see his/her performance and improve based on the results. System evaluation can also be performed in a similar fashion. This is feasible on hard-disk based PC/XT's only as the additional space needed for Performance Measurement and Evaluation can be fairly large. Hardware monitoring functions will be rather difficult (and useless) to incorporate, since the simulator is only a model. User monitoring

is, however, much more important.

The data base will be populated mainly from records downloaded from the NASA/RECON system. The thesaurus, however, can not be downloaded and mainly depends on the records downloaded. If this is not possible, then MADAM records can be easily downloaded and used. Again a thesaurus and dictionary will have to be formulated and this can ba a major task.

5.4

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
IN-82	~~44792~~ 183574	

4. Title and Subtitle	5. Report Date
USL/NGT-19-010-900: GENERAL SPECIFICATIONS FOR THE DEVELOPMENT OF A PC-BASED SIMULATOR OF THE NASA RECON SYSTEM	August 2, 1984
	6. Performing Organization Code

7. Author(s)	8. Performing Organization Report No.
SPIROS TRIANTAFYLLOPOULOS	
	10. Work Unit No.

9. Performing Organization Name and Address	11. Contract or Grant No.
University of Southwestern Louisiana The Center for Advanced Computer Studies P.O. Box 44330 Lafayette, LA 70504-4330	NGT-19-010-900
	13. Type of Report and Period Covered
12. Sponsoring Agency Name and Address	FINAL; 07/01/85 - 12/31/87
	14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

The general specifications for the design and implementation of an IBM PC/XT-based simulator of the NASA RECON system, including record designs, file structure designs, command language analysis, program design issues, error recovery considerations, and usage monitoring facilities. Once implemented, such a simulator will be utilized to evaluate the effectiveness of simulated information system access in addition to actual system usage as part of the total educational programs being developed within the NASA contract.

This report represents one of the 72 attachment reports to the University of Southwestern Louisiana's Final Report on NASA Grant NGT-19-010-900. Accordingly, appropriate care should be taken in using this report out of the context of the full Final Report.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
NASA RECON Simulator, PC-Based Simulator Specifications, PC-Based Research and Development	

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	21	

*For sale by the National Technical Information Service, Springfield, Virginia 22161