DEPARTMENT OF COMPUTER SCIENCE
COLLEGE OF SCIENCES
OLD DOMINION UNIVERSITY
NORFOLK, VIRGINIA 23529

AUTOMATED DATABASE DESIGN TECHNOLOGY AND TOOLS

By

Stewart N.T. Shen, Principal Investigator

Final Report
For the period ended February 29, 1988

Prepared for the
National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia 23665

Under
Master Contract Agreement NAS-1-17993
Task Authorization No. 54
Floyd S. Shipman, Technical Monitor
ISD-Systems Architecture Branch

Submitted by the
Old Dominion University Research Foundation
P. O. Box 6369
Norfolk, Virginia 23508-0369

June 1988

Automated Database Design Technology and Tools
Final Report

Stewart N. T. Shen

Old Dominion University
Norfolk, VA 23529

Abstract:

We summarize our reserch project results in this final report. For more detailed reports of this project, please refer to Interim Reports 1 through 4[1,2,3,4]. In this final report, we provide our overall comments on the state of the art in various aspects of database design and our recommendations on further research for SNAP and NAVMASSO's future database applications.

I. Theoretical Efforts in Database Design

The major theoretical efforts in database design concentrate mostly on relational database design. They include Data Dependency Theory, Inference Rules for Dependencies, Normal Forms of Relations, and a number of algorithms for database design.

There are available now many theories and algorithms leading through designing of minimal FD-prserving, lossless-join 3NF/BCNF relations. Such theories and algorithms are ripe for computerization. In fact, tools are now available for just doing such tasks as discussed in a later section.

There are further theories for 4NF and even higher levels of normal form relations. But these higher level normal forms have their weaknesses and are not yet suitable for automation.

There are no formal theories in heirarchical or network database design. But the design of such databases can also take advantage of the theories and algorithms for relational databases to achieve better heirarchies or networks. Additionally, there are methodologies that allow the conversion between the different databases.

II. Database Design Methodology

It is generally accepted that database design should be performed according to a methodology such as the following. First start with the requirements analysis and specification. Next convert such a specification into a high level, graphic semantic model facilitating the understanding of the enterprise environment. Refinements and clarifications can be more effectively done on a semantic model. One of the most popular semantic modeling technique is to use the entity-relationship approach. Such a semantic model, called the entity-relationship model, is then converted to a relational, heirarchical, or network database design according to certain guidelines or algorithms. These conversion algorithms exist now in such a form that a significant amount of computerization is feasible. Some computerized tools are available as discussed in a later section.

The weakest link in database design may be in the requirements specification. This is because of the fact that requirements specification is a highly intellectual process that requires a thorough understanding of the problem environment and must be spelled out in an unambiguous and complete manner. It was demonstrated in the Sixth International Conference on Entity-Relationship Approach, N.Y., N.Y., November, 1987 that with an identical seemingly clear but actually ambiguous specification, different groups of professional designers came up with rather different semantic models.

Computerized tools are also available that can assist in requirements specification. However, they can not guaranttee total clarity, validity, or completeness. We assert that good tools for efficient, effective, and clear description and specification of problem environments are still lacking.

III. Computerized Tools for Database Design

There are many tools developed that may be useful for database design. We categorize them into two groups described below.

III.1. Tools for requirements specification

A tool for requirements specification typically has three parts. One part is a language that allows users to specify the requirements. The language often has a graphics component. Another part is the

processor that understands the language and can process it. There are often a set of utility programs, may be considered part of the processor, that can perform useful tasks such as generating different kinds of reports for certain purposes. Still another part is the database which stores the information specified in an efficient represnetation that can be utilized and manipulated by the processor.

A tool for requirements specification can reduce clerical efforts, provide better access to different parts of the specification, allow clearer and better specification, and may even produce a basis from which computerized conversion to semantic database models can be implemented. However, as good as they may sound, such tools still can not insure truely unambiguous, complete, and valid specifications. As mentioned above, and can not be overemphasized, that requirements specification remains to be the weak spot in database design.

We have looked into several of the tools basically from the literature. A well reported, sophisticated one is SREM, which is an extension of PSL/PSA. SREM is not available on micro or mini computers, which we believe to be more suitable environments for NAVMASSO. Most micro computer based systems use data flow diagrams. Some more popular ones are Excelerator (Index Technology, Massachusetts) and PCSA (StructSoft, New Jersey). Some relatively newer ones are Consoi-DFD (SystemOID, Quebec, Canada), Blue/20 (Advanced Logical Software, California), and PC-IASP (Control Data, Minnesota). These packages have price tags varying from hundreds of dollars to thousands of dollars. At this stage, we have not got either the time or the money to test all these packages. We performed some experiments on Excelerator and a demonstration package of PC-IASP. Both systems performed similarly for our purposes and did what they were supposed to, that is they allowed data flow diagram representation of specificaitons. One weakness is that the small screens on the micro computers do not provide a good view of the graphics. We feel that large-screen workstations would be more suitable for such systems utilizing graphics.

III.2. Tools for semantic modeling and schema generation

Several software packages have been developed for semantic modeling, mostly based on the entity-relationship approach and implemented on micro computers. Some vendors provide multiple modules that will not only allow using graphics to do the entity-relationship model design, but also do

further normalization and conversion to different schema definitions for various database management systems. They may also generate different types of useful reports. Some other vendors are relatively new into the market and have fewer modules that can do fewer things at this time. Since the necessary theories and algorithms are available, there is no question that as time passes by, more vendors will provide more modules that perform more tasks. On the other hand, all these packages are relatively new in the market and may take a little longer to smooth out some bumps and corners and to eventually mature.

Examples of the more complete software packages are MASTER packages (InfoDyne, Indiana), PC-IASP packages (Control Data, Minnesota), and ER-Designer packages (Chen and Associatess, Louisiana). The less complete packages include Consoi-ERM, Consoi-ERD packages (SystemOID, Quebec, Canada), and Blue/20, Blue/60 packages (Advanced Logical Software, California). We performed some experiments with the demonstration package of PC-IASP and a thorough study on the ER-Designer packages. We picked ER-Designer for a thorough study partly for its relative completeness, and partly because of the fact that it was produced by the inventor of the entity-relationship approach, namely P. Chen. This set of packages provides a wide spectrum of database design functions. Our experience with this set of packages is that it does a good job but is still at a garbage-in-garbage-out level. The packages are still a little rough and do not provide helpful warnings, advices, or explanations.

Our conclusion on the study of these packages is that they are very useful for database design in improving the quality of design and in saving manpower. But at the current state of development, there is still ample room for improvements for such packages. More user-friendliness, capability of providing adivices, warnings, and explanations, and large screen for graphic display would make such packages even more effective.

IV. The Graphic Knowledge Base Shell

The Center for Artificial Intelligence at Old Dominion University is developing the Graphic Knowledge Base Shell. This is going to be a generic system allowing the building of many kinds of

graphic representation systems very conveniently. Such graphic representations are supported by text files with excellent text editing facilities. Different software modules can be implemented on top of this system to allow processing the representations in different ways, including a data flow diagram processor and a database designer. The system is intended to be a super-hypertext system. It is designed to be a tool for powerful, effective, and efficient knowledge representation and retrieval thus should be very useful for NAVMASSO's SNAP and future database and other applications.

We give an illustration of the Graphic Knowledge Base Shell in its current preliminary form by describing the SNAP forms. Figure 1 shows the relationships of these forms in a graphic representation.

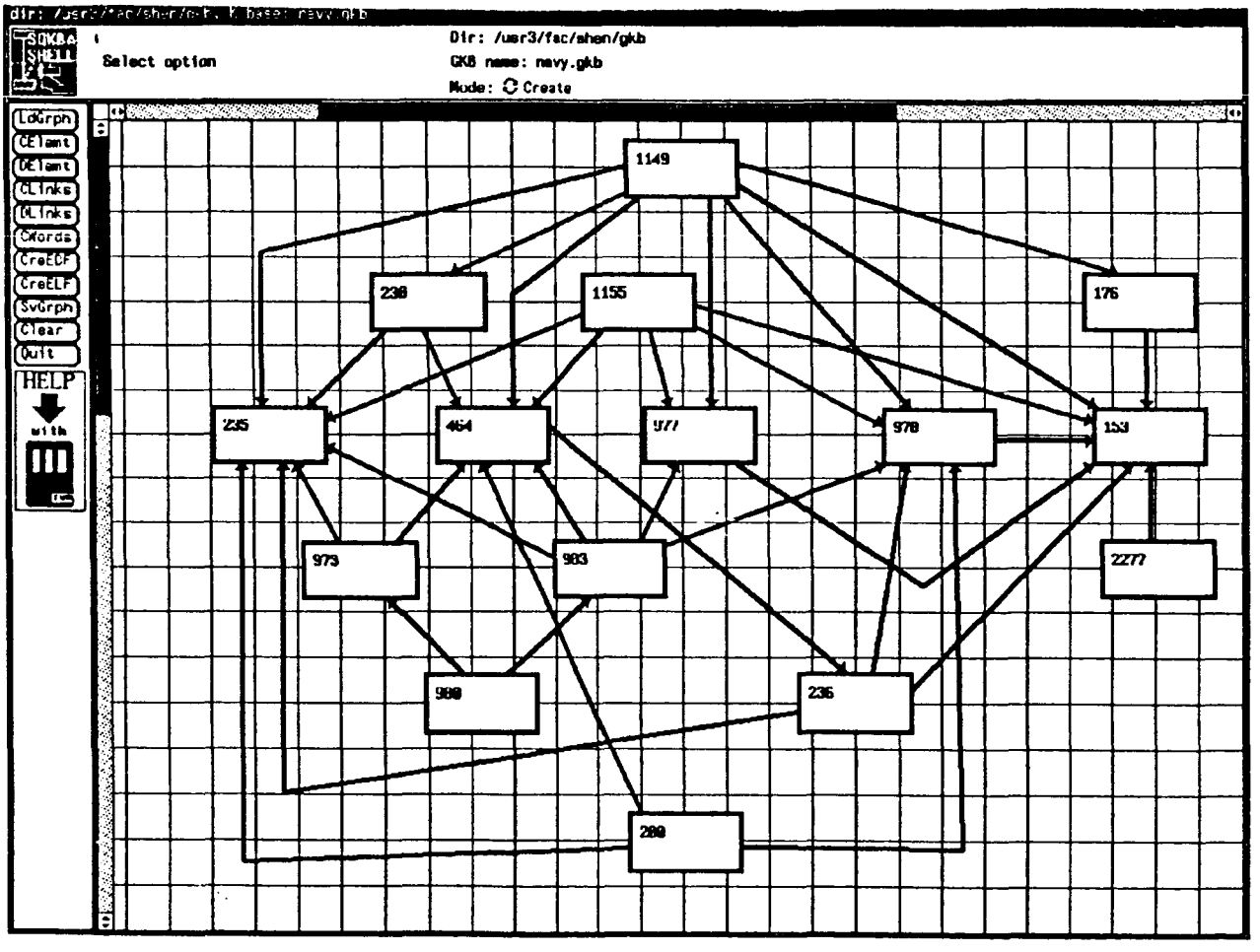
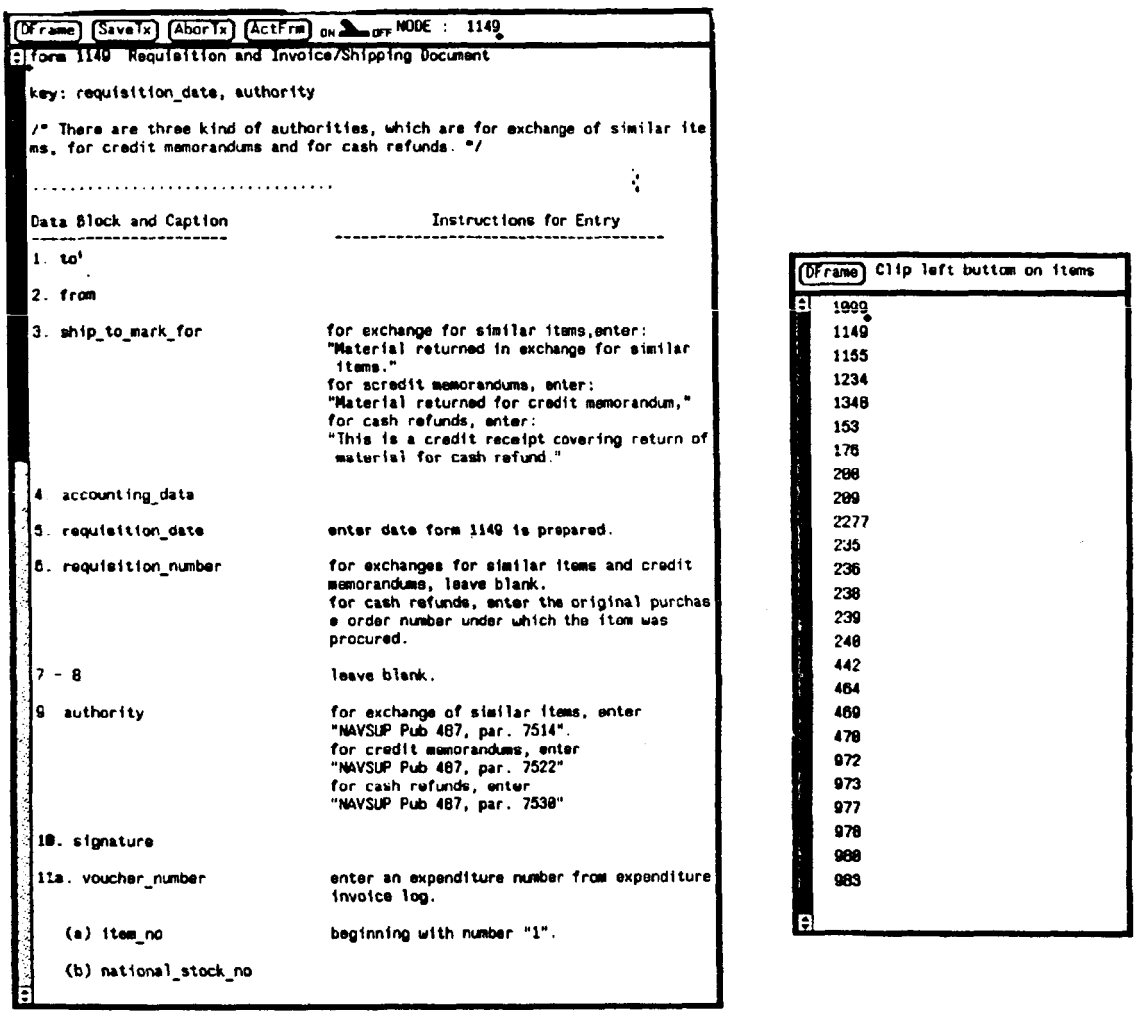


Figure 1. The SNAP Forms Relationships

Figure 2 (a) shows an example of part of the textual description of Form 1140. The text screen can be scrolled at the clicking of a mouse button to show the entire text file. Figure 2 (b) shows a list of all the SNAP forms. With the clicking of a mouse button, a particular form can be automatically selected for highlighting and the textfile displayed. Multiple screens of text displays as in Figure 2 (a) can be shown on the same CRT screen simultaneously. This capability is particularly conducive to a better understanding for certain situations.



(a) (b)

Figure 2. A Text File and the List of All SNAP Forms

The Graphic Knowledge Base Shell should allow convenient and effective analysis of the SNAP forms. When fully developed, it can be a good database design tool and a design and analysis tool for almost any system.

The Graphic Knowledge Base Shell is being developed on the Sun workstations, which have large screens excellent for graphic display. Since the Sun workstations use the Unix operating system and have the powerful Sun Tool facilities, the entire working environment is very pleasant, very convenient, and very powerful. Multiple files can be accessed simultaneously and multiple processes can be executed simultaneously too.

V. Conclusion and Recommendations on Future Research

We have found that research and development efforts in database design and its computerization have been intensive. Available theories, algorithms, and methodologies have made computerization of a significant portion of the database design process a feasible task. Many computer software tools are in fact already available in the commercial market. However, all of such software packages are still somewhat rough and most are at the garbage-in-garbage-out level. They can not provide advices, warnings, and explanations beneficial to the designer. Some of the tools are availble only on the mainframe machines and most others are available only on micro computers. Systems on micro computers are good for portability but do not provide large display screens which are better for graphics, while most representations do include graphics for ease of understanding.

We feel that systems implemented on multi-tasking, large-screen workstations can be most convenient and effective. In particular, we strongly believe that the Graphic Knowledge Base Shell described earlier in its full form can be a very effective tool for analyzing the NAVMASSO data processing environment and for any system design activities including database design.

For future research, we recommend that NAVMASSO support further development of the Graphic Knowledge Base Shell and use it for some analysis and design activities. We also recommend to keep abreast of the current developments in the computerization of database design process, since they are continuously changing. Such a familiarity is needed before an intelligent policy decision can be made. A study of SNAP to get familiarized with its operation and to recognize any of its defects or inadequacies may result in some recommendations on revisions to achieve significant improvements before conversion to a database system. Such a study will also be beneficial for future design of a superior

database system good for an integrated information management system for NAVMASSO.

References:

1. Shen, S. N. T. "Automated Database Design Technology and Tools, Interim Project Report 1", April 15, 1987, 19 pages.

2. Shen, S. N. T. "Automated Database Design Technology and Tools, Interim Project Report 2", June 30, 1987, 100 pages.

3. Shen, S. N. T. "Automated Database Design Technology and Tools, Interim Project Report 3", Spetember 29, 1987, 30 pages.

4. Shen, S. N. T. "Automated Database Design Technology and Tools, Interim Project Report 4", February 29, 1988, 30 pages.