Proceedings of the 1987 Winter Simulation Conference
A. Thesen, H. Grant, W. David Kelton (eds.)

RE-HOSTING A COMPUTER ASSISTED WARGAME EXERCISE FROM A MAINFRAME TO A MICRO: DATABASE AND USER-INTERFACE ISSUES

Michael D. Brooks
Mark S. Kross
Mark A. Roth

Air Force Institute of Technology
Department of Electrical and Computer Engineering
Wright-Patterson AFB, OH 45433-6583

ABSTRACT

This paper addresses the issues concerned with transporting a large-scale, computer assisted, warfare exercise program from a mainframe to a microcomputer environment. Its main focus is on replacing the current application dependent file system with a commercial DBMS and using a 4th generation language applications development tool to develop a more friendly user interface. Issues concerned with database and user interface design are discussed.

1. INTRODUCTION

The Theater War Exercise (TWX) (Theater Warfare (1986)) is a computer assisted, theater level, airpower employment exercise conducted at the Air Force Wargaming Center, Maxwell AFB, AL. TWX is designed to provide senior Air Force officers with a realistic, five day, European theater conflict which will test and evaluate their warfighting skills.

Player decisions required for this exercise are typical of those that an air component commander and staff would make during an actual war. These decisions are fed into the TWX air and land battle simulation programs which model the employment of the airpower strategy, doctrine, and warfighting principles inherent in those decisions. Once the simulation is run, the players receive their battle results, logistic status, weather forecast, and new air/land orders of battle in a computer run format. Using these results, the players revise their strategy as necessary and prepare the next day's battle plan.

In its present configuration, the exercise is a large, Fortran IV program which runs on a Honeywell H6000 mainframe computer. All inputs are made on slow, hard copy devices which severely limit the interaction by game controllers and exercise participants. Because of these limitations, players cannot receive on-line help from the computer and must use a very rigid input format to process exercise information.

This "user hostile" interface tends to frustrate the players because they must learn a complicated computer syntax rather than concentrating on playing the exercise. Also, due to processing constraints built into the software, (e.g., fixed amount of airbases) the exercise is not very flexible and is difficult to maintain.

TWX is also played each year at the Canadian Forces Command and Staff College in Canada and the Royal Air Force Staff College in England. This remote processing requires a compatible Honeywell H6000 computer facility to be leased at each foreign site and Air Force Wargaming Center personnel to setup and conduct each exercise at a great expense.

The main goals of this project are twofold. First, the game's user interface must be modified to incorporate good software engineering design principles. Here, we intend to make use of a commercial DBMS and 4th generation language (4GL) application development tool to design screens which incorporate help facilities, on-line editing, and real-time data validation.

Second, to provide better exercise portability, a combination of standard IBM PC/XT compatible microcomputers and a Digital Equipment Corporation MicroVAX II will be used. This setup provides an environment which is powerful enough to support up to 16 simultaneous users configured such that up to eight separate exercises could be running concurrently. By meeting these two goals, a new, user friendly Micro-TWX exercise can be played at any site using relatively inexpensive hardware and software.

The rest of this paper discusses the issues concerning hardware selection, database selection and design, user interface issues, and rehosting the air/land battle simulation.

2. HARDWARE SELECTION

In selecting the hardware for the new Micro-TWX system there were several factors to consider. First, we needed a multi-user system or a local area network (LAN) configuration to support a minimum of three users, i.e., blue players, red players, and a game controller. Second, a powerful central processor is required to run the actual game simulations in an acceptable amount of time. In most cases, the actual simulation is run at night so turnaround time is not critical. However, the exercise schedule currently requires a 4 hour maximum turnaround for the final day of the exercise.

Finally, a large amount of secondary disk storage is needed to store the TWX database and simulation programs. Depending

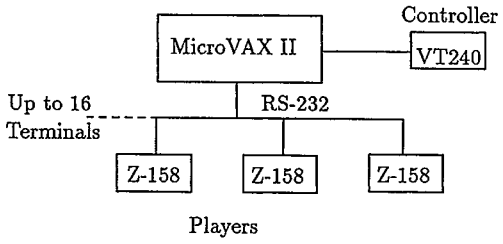


Figure 1: Micro-TWX Hardware Configuration

on the number of seminars the system must support, anywhere from 20 to 100 MB of storage will be needed.

To solve these requirements and still meet our goals of portability we decided to use a MicroVAX II, Z-158 PC/XT combination as depicted in Figure 1. By connecting the Z-158s and MicroVAX together, the MicroVAX can perform as a central file server for concurrent exercises, with the Z-158s running the user interface software. Also, the MicroVAX, using the MicroVMS operating system, has the capability to run batch jobs which would allow large report jobs to run in the background thus, freeing up the individual player's terminal for other processing requirements.

3. DATABASE ISSUES

The database function within the current TWX system is handled by a collection of independent application programs. These programs access multiple data files which reflect an emphasis on physical implementation rather than logical data organization. Consequently, a large amount of data redundancy exists and the data structures cannot be modified easily without extensive changes to the TWX system. Also, in order to re-host the TWX system, the current specialized Fortran IV, I/O subroutines would have to be rewritten for the new microcomputer version. Since the current file system contains these drawbacks and recoding the current system would require excessive time and resources, it is more advantageous to incorporate a vendor supplied DBMS in place of the current application system.

To incorporate a DBMS into TWX, the current user interface software would be removed from the system and replaced by the 4GL applications software. Also, the I/O calls within the actual simulation programs will be replaced by embedded calls to the DBMS. This will involve a major software coding effort and is the main challenge of the project.

3.1. DBMS Selection

During the DBMS selection process, we narrowed our choices down to three true database management systems which met most of the TWX DBMS requirements; MDBS III, an extended-network DBMS, Oracle, a relational DBMS, and Ingres, also a relational DBMS.

Overall, the best package with regard to the database function for the Micro-TWX implementation was MDBS III. MDBS III, using the extended-network model, was better suited to handle the intricate data relationships required for the Micro-TWX database. However, at the time of this evaluation, the MDBS III PC version did not support a 4GL development tool or a forms management generator. Without these features, the processing power of the Z-158 could not be utilized unless the user interface application software was developed in a high-order-language such as Pascal. Since a 4GL and forms management system were needed to rapid prototype the user interface, MDBS III was eliminated from consideration.

With MDBS out of the running, the selection process focused on Oracle and Ingres. Both database management systems were proven performers and each had a powerful 4GL application development tool which included a forms development package. Either system provided all the capabilities necessary to develop a professional Micro-TWX user interface incorporating on-line help, windowing, and referential integrity checking features. Also, both systems have a powerful report writer and support the Structured Query Language (SQL) which is the "industry" standard query language.

Although Oracle rated higher in the 4GL and forms management area and was our initial choice, Ingres was selected for this project due to one overriding factor. Since, the Air Force Wargaming Center is currently using Ingres for other in-house database applications and already has a trained support staff, Ingres was selected to preclude unnecessary training. This factor, plus significant cost reduction incentives, made Ingres the logical choice. Figure 2 depicts the proposed Ingres DBMS setup for the Micro-TWX project.

3.2. Micro-TWX Database Design

The database for the new Micro-TWX version serves as a repository for all the data necessary to conduct the exercise. This includes initialization data, player inputs, and simulation constants used throughout the game.

The technique used to design the database is described by Korth and Silberschatz (1986) and Teorey, et al. (1986). First, an entity-relationship (E-R) diagram is developed which will facilitate the database design by specifying a high-level, object-oriented scheme. This scheme represents the overall logical structure of the database and provides an intuitive view to the designer of how the database will function.

For our design, the current TWX file system is used as a guide to construct this diagram. Also, an analysis of the current game input and reporting requirements is conducted to ensure the necessary data is represented within the E-R diagram. The next step

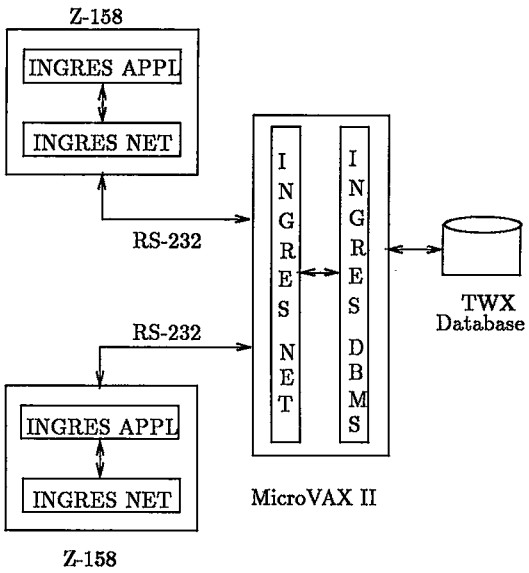
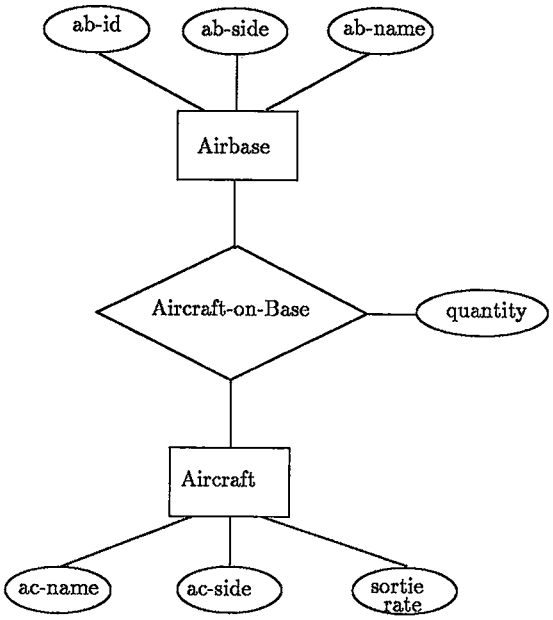


Figure 2: Ingres Z-158/MicroVAX Configuration

involves converting the E-R diagram into a relational database design. The relational model, utilizing a tabular approach for representing data, has numerous advantages (Korth and Silberschatz (1986)), and is the model used in Ingres. We will use relational database normalization techniques to achieve a design which minimizes data redundancy while maintaining the ability to efficiently test data integrity.

Although there are formal algorithms for designing relational systems given a set of data dependencies, we have decided to use a more practical design approach where each entity and relationship in the E-R diagram is reduced to a relational table using special transformation rules (Korth and Silberschatz (1986) and Teorey, et al. (1986)). Figure 3 shows a small section of the Micro-TWX E-R diagram along with the identified functional dependencies. Each functional dependency $X \rightarrow Y$ states that for each X-value there a unique Y-value associated with it. For example, the functional dependency ab-id ab-side $\rightarrow$ ab-name states that for each airbase identifier, airbase side there exists a unique airbase name. Figure 4 depicts the tables derived from the entities and relationships shown in Figure 3.

The E-R diagram for the Micro-TWX database consists of approximately 40 entities (e.g., airbases, aircraft, land units, etc) and over 30 different relationships between these entities. All together, the database contains well over 500 unique data attributes. To help organize the database, the entities and relationships were broken out into major functional areas consisting of airbases, aircraft, munitions, air missions, land units, and exercise constants. Also, in order to speed up the user interface query processing, we originally planned to horizontally partition



Functional Dependencies
ab-id ab-side $\rightarrow$ ab-name
ac-name ac-side $\rightarrow$ sortie-rate
ab-id ab-side ac-name ac-side $\rightarrow$ quantity

Figure 3: Partial Micro-TWX E-R Diagram

Airbase Entity

ab-id	ad-side	ab-name	...
50	blue	Ramstein	
63	blue	Bracknell	
75	red	SAFP AD	
52	red	EGAF AD	

Aircraft Entity

ac-name	ac-side	sortie-rate	...
F4G	blue	5	
F111E	blue	3	
B52G	blue	2	
U15	red	6	

Aircraft-on-Airbase Relationship

ab-id	ab-side	ac-name	quantity
50	blue	F4G	15
50	blue	B52G	5
75	red	U15	30
52	red	U15	20

Figure 4: Sample Table Constructed From E-R Diagram

tional partitioning of the airbase and air mission tables by Allied Tactical Air Force (2 and 4 ATAF). However, due to the number of additional tables this partitioning generated and conflicting query requirements with regard to the airbase and air mission tables, we decided only to partition by the side attribute. By using the E-R diagram to table design approach along with horizontally partitioning the data, we were able to develop a logical database scheme which enhanced query processing, minimized data redundancy, and provided an efficient means to enforce data integrity.

4. DEVELOPING NEW USER INTERFACES

The process of developing new user interfaces is a software engineering problem. The basic approach are taking is that of prototyping. Due to time constraints, we are limited on the number of prototypes we can develop. Therefore, we are using prototyping as a means of establishing user requirements rather than using prototyping throughout the entire development process. The steps involved in designing the new user interfaces are: requirements analysis, screen design, developing a prototype, final design and implementation, and testing and integration.

Requirements analysis includes finding and analyzing user requirements for both information and control flow. As far as the information requirements are concerned, they are largely the same as those of the current system. This is because the actual play of the game and the decision making processes involved need to remain the same. Since the use of hard copy devices for input is no longer be a constraint, many of the input formats have been modified in order to allow the exercise players to more efficiently make their logistics and mission planning decisions.

Analysis of control flow involves finding out the best sequence of screens to allow the user to input data easily. For instance, editing features which allow for changing the input on a previous screen are highly desirable for TWX players. In-person discussions with Air Force Wargaming Center personnel have allowed for the completion of the requirements analysis phase.

The screen design phase involves the drawing of each user input and output screen. These were first done by hand and then generated by the INGRES Visual Forms Editor program. This program will allows to generate an initial prototype without having to connect the user interface screens to any actual database implementation or simulation software.

Together with added control flow, the screens we design constitute an initial prototype which will be evaluated by Wargaming Center personnel. Their suggestions will be incorporated into a more final design which will then be implemented with the new database system. Developing the TWX user's guide is an ongoing process throughout the development of this project. The user interface software maintenance manual will be written only after the code itself is in a reasonably final form.

The final task in the user interface design effort is identical to the final tasks in designing the new database system and rehosting the simulation programs. This is the task of total system integration, validation, and testing. For the user interfaces themselves, the main testing criteria is Wargaming Center acceptance, and the verification of transactions to the database. The Wargaming Center will determine if the user interfaces are indeed correct and useful in terms of ease of use and the exercise's information requirements. By verifying transactions, we mean that it is necessary to run several checks to make sure that the input through the user interfaces has the correct effect on corresponding database entries. For example, we will verify that sorties entered against a target through the user interface actually get scheduled against the target, as shown from entries in the database.

The blue and red user interfaces are being developed separately, since one interface cannot serve for both the red and blue players. This is despite the fact that the two interfaces are largely identical. This two interface requirement is necessitated by a decision to horizontally partition the database by player side. Therefore, the table names used in the user interface software must be different for the red and blue sides. We are implementing the blue interface first, and then we will quickly produce the red interface, using the blue screens and interface code as a template for development. A few of the basic aspects of the interfaces must be different for the red side, but they should not present a problem during implementation.

Currently, the blue user interface is being implemented using the PC version of INGRES. When the networking hardware and software are in place, the user interfaces will be moved to a PC attached to the MicroVAX II. At that point, the user interface software must be modified in order to work with a remote database residing on the MicroVAX II.

An example of the type of improvements we are trying to make with the new user interfaces is in the area of logistics planning in TWX. Predirect rates are the daily airbase resupply rates for petroleum, spare parts, and munitions. In TWX, these rates require adjustment as the game progresses from day to day. With the current version of TWX, the base number and the desired predirect rates are entered over a hard copy device. No feedback or editing capability is provided to the user. For instance, if the user accidentally inputs the wrong base number before entering new predirect rates, then the input rates will be entered in the database against the specified (incorrect) airbase. With the new user interface, such a situation should never occur. A clear key exists which can be used to start the entry process over with a new airbase number. No transactions are actually entered against the database until a commit key is pressed. An online help screen is available. A column showing the current predirect rates for the entered base number serves as a memory aid for

AAFCE PREDIRECT RATE CHANGE

Day 1 Enter Airbase Number: 17

Logistics Item:	Current Predirect Rate:	New Predirect Rate:
POL	120	110
SPARES	50	25
AIMR	0	20
AIMI	0	0
GP1	12	12
GP2	12	12
CBU1	0	0
CBU2	0	0
MAVR	30	0
STDA	0	25
AIMX	12	12
GB	10	0

F1 for Help
F5 to Commit Changes
F7 to Delete All Entries, Including Airbase Number
F10 to Return to Overall AAFCE Menu

Figure 5: AAFCE Predirect Rates Change

the user. Figure 5 shows the layout of the predirect rates change screen for the new user interface.

The nature of the necessary PC networking, along with certain control aspects of the game, makes it necessary to implement locking procedures at both the data and application levels. When multiple users are allowed to access certain database tables at the same time, data locking must be used to ensure a consistent database. In addition we need application locking to restrict user access to certain functions of the user interface software, depending on what functions other users are currently using or have used with the user interface.

An example of application locking in TWX is the relationship between mission planning and aircraft rerole. In TWX, some types of aircraft may be reroled from one type of role to another. For instance, F4 aircraft may be reroled from an attack role to a defense role. Only certain aircraft with specific role designators are allowed to fly particular mission types. Once the missions are planned, it is obviously unacceptable to allow the aircraft flying those missions to be reroled. This must be handled by locking users out of the aircraft rerole function of the user interface after the mission planning function has been accomplished. The planned method of implementing application locking is the use of flags stored in a seminar control table in the database. Furthur, the method used to assure the consistency of these flags will be data level locking, which is a function available through the INGRES software.

An interesting aspect of the design of the user interfaces is the selection of a database query language. The INGRES package supports two different query languages: SQL (Structured Query Language), and QUEL (an INGRES-developed query language). The decision was made to use SQL since it is becoming the de facto standard in database retrieval languages. This selection will enhance the maintainability of the user interface software in two ways. First, future rewrites of the system which use some other DBMS package will be easier, since almost all commercially available DBMS offer SQL as a query language. Code conversion will, therefore, be relatively easy. Second, SQL is widely taught as a model database retrieval language in database design classes. Therefore, the future maintainers of the TWX system are likely to have been introduced to SQL at some point in their academic careers.

5. REHOSTING THE AIR/LAND BATTLE SIMULATION

The majority of the effort in rehosting the air/land battle simulation will be in the beginning, when we are trying to separate the simulation code from that which is part of the old user interface or old database system. This rehosting will be the last thing

done before total system integration and testing. A particularly troublesome problem here will be validation of the simulation software once it is rehosted. This is because no formal validation was ever done on the original system's simulations. The system was built with what were thought to be reasonable parameters for the air and land conflicts. The exercise was experimented with and these parameters were adjusted until the exercise outcome satisfied the original system builders. The judgement criteria used was field experience and a sense of what should happen under given circumstances, not on any absolute model.

Since no true validation was ever done on the simulations, we will have to define validation criteria of our own. Due to time constraints, we will not be able to validate the simulations and equations against standard conflict models, as should be done. This is an area requiring further research. Our validation criteria will have to be a comparative analysis of the current exercise with our new system. Our plan is to input identical data to both systems, along with identical random number streams if possible, and then to compare the results of the simulations from each system. General trends in terms of attrition to each side should then be the same in both cases.

6. CONCLUSION

This paper has discussed some of the issues involved with transporting the Theater War Exercise wargame simulation from a mainframe to a microcomputer environment. Particularly, we have addressed the need for incorporating a commercial DBMS and 4th generation applications tool in order to enhance the user interface and improve software maintainability.

ACKNOWLEDGEMENTS

This research was supported by a grant from the Air Force Wargaming Center, CADRE/WG, Maxwell AFB, AL, 36112.

REFERENCES

Theater Warfare Exercise Users' Handbook (1986), Air University Center for Aerospace Doctrine, Research and Education, CADRE/EDW, Maxwell AFB, Alabama.

Korth, H. F. and Silberschatz, A. (1986), *Database System Concepts*, McGraw-Hill, New York.

Teorey, T. J. and others (1986), A Logical Design Methodology for Relational Databases Using the Entity-Relationship Model, *ACM Computing Surveys.*

AUTHORS' BIOGRAPHIES

MICHAEL D. BROOKS is an M.S. student in the School of Engineering at the Air Force Institute of Technology and a Captain in the United States Air Force. He received a B.S. in Computer Science from the University of South Carolina in 1979. His previous assignments included systems analyst for the 552 Airborne Warning and Control Division in the area of E-3A (AWACS) command and control software and systems analyst for the Tactical Air Warfare Center, Information Systems Directorate in the area of Electronic Warfare computer systems support.

Michael D. Brooks
Air Force Institute of Technology
AFIT/ENA
Wright-Patterson AFB, OH 45433-6583

MARK S. KROSS is an M.S. student in the School of Engineering at the Air Force Institute of Technology and a Captain in the United States Air Force. He received a B.A. in mathematics from the University of Alabama in 1983. His previous assignment included systems analyst and programmer for Headquarters Strategic Air Command, Command and Control Directorate, in the area of Force Control computer support.

Mark S. Kross
Air Force Institute of Technology
AFIT/ENA
Wright-Patterson AFB, OH 45433-6583

MARK A. ROTH is an assistant professor of computer systems in the School of Engineering at the Air Force Institute of Technology and a Captain in the United States Air Force. He received a B.S. in computer science from Illinois Institute of Technology in 1978, an M.S. in computer systems from the Air Force Institute of Technology in 1979, and a Ph.D. in computer science from the University of Texas at Austin in 1986. His previous assignment included systems analyst and programmer for Headquarters Air University, Data Automation Directorate in the area of computer assisted wargaming exercises. His current research interests include wargaming simulation and database management systems. He is a member of ACM and IEEE Computer Society.

Mark A. Roth
Air Force Institute of Technology
AFIT/ENG
Wright-Patterson AFB, OH 45433-6583