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

DESIGN OF A JOINT OPERATIONS PLANNING SIMULATOR

James R. Jansen
Mark A. Roth

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

ABSTRACT

The Joint Operations Planning System (JOPS) is a Department of Defense system for the conduct of the joint planning process. We describe the design of microcomputer implementation of a JOPS simulator, called JPLAN. We present some background on JOPS and JPLAN, problems with the current simulation, and our approach for a new microcomputer implementation which eliminates the "user hostile" interface for users and maintainers of the simulation.

1. INTRODUCTION

To be effective the four military services must take a joint, or team approach to attaining military objectives. This means that we must plan our operations together. As one can imagine the job of fulfilling the requirements of each service is in itself a monumental task. And the process of integrating these operation plans throughout the four services is even more complex. The Joint Operation Planning System (JOPS) provides a standardized approach for planning and integrating these joint military operations. The following is an outline of the JOPS: (Joint Operation Planning and Execution (1986))

1. INITIATION - The need for a plan is identified.
2. CONCEPT DEVELOPMENT
   (a) Analysis of the Missions and Tasks
   (b) Preliminary Planning Guidance
   (c) Preparation of Staff Estimates
   (d) Preparation of the Commanders Estimate
   (e) Preparation of the Concept of Operations
3. PLAN DEVELOPMENT
   (a) Force Planning
   (b) Support Planning
   (c) Chemical and Nuclear Planning
   (d) Transportation Planning
   (e) Force, Movement and Support Shortfall Identification
   (f) Transportation Feasibility Analysis
   (g) Concept Approval and Refinement
   (h) Plan Documentation and Refinement
4. PLAN REVIEW
5. SUPPORTING PLANS

In this paper, we describe a microcomputer design and implementation of a JOPS simulator for use in the training and education of Air Force officers and logistics planners. This simulator will be used by the Air Force Wargaming Center at Maxwell AFB, Alabama in support of the curriculum of the Air Force's intermediate and senior service schools. It will also be used as part of the Combat Logistics graduate and professional continuing education curriculum at the Air Force Institute of Technology.

The organization of this paper is as follows. We first present some background on the JPLAN simulation, describing its current use and problems with its implementation. We present an overview of the simulation and our approach to designing the database, user interface, and simulation software for the microcomputer.

2. BACKGROUND

The Joint Planning Exercise (JPLAN) is a simulation of the Joint Operation Planning System (JOPS) used by the unified and specified commands to develop operation plans. The Rapid Deployment Exercise (RADEX) is a variation of JPLAN which adds sealift capability and is played at the Air War College. Throughout this paper the term JPLAN will be used to refer to the exercise in general. JPLAN allows student teams to identify needed combat, combat support, and combat service support forces; to build force lists; and simulate deployment of these forces in support of an operation plan. Students first act in the capacity of logistics planners at the component level to select forces and build force lists. Then, they act in the capacity of unified command planners for airlift simulation.

JPLAN was originally designed for the Air Command and Staff College (ACSC) as part of their Theater Warfare Studies program. In that role, JPLAN supports up to 80 teams simultaneously plus a monitor program for instructors and system operators. The exercise currently runs on the Honeywell 6000 series computer with student teams making inputs with TI Silent 700 modem terminals.

JPLAN is also used at the Air Force Institute of Technology (AFIT) as part of the Combat Logistics graduate and professional continuing education curriculum. The exercise runs on the Air Force Logistics Command (AFLC) Honeywell computer using the CREATE system dial up capability. Only four to six teams play at a time and no instructor monitor capability is needed. The AFIT course is also presented at many Air Force installations worldwide. To meet this requirement the exercise is run on a Honeywell computer at the installation, if one is available, or through a telephone link to the CREATE system at Wright- Patterson AFB, Ohio.

Neither of these alternatives is ideal. Running the exercise on a computer at the installation presents many problems, including allocating sufficient computer time, the availability of computer personnel to install the program, compatibility problems involved in getting the program up and running, and availability of terminals to use the exercise. The other alternative of using a telephone link to the CREATE system also has problems of finding sufficient terminals to use, losing the telephone link due to technical problems, and the high expense of time over telephone links.

The current implementation of the exercise is described as "user hostile" because both the hardware and software can be very frustrating to use. The exercise uses hardcopy thermal printer terminals (i.e., TI Silent 700) operating at 300 BAUD. Printing long listings of data on this device is quite common and consumes a large amount of the users' time, which could be utilized much more effectively with a screen oriented system and a dot matrix printer for reports. On the software side, all inputs are line oriented since a screen doesn't exist. Since the implementation is closely coupled to FORTRAN conventions, the user is forced to enter lines of data with commas and spaces placed exactly as required or he must start over when an input error occurs.

The difficulties encountered with running the JPLAN exercise on a mainframe computer system is being solved by redesigning JPLAN to a microcomputer system (specifically, the Zenith Z-158 and Z-248). This solution will allow the exercise to operate independently of a large computer system and eliminate the use of terminals for calling the CREATE system. This will allow the exercise to be used anywhere a compatible microcomputer can be found, which today means almost any government office or organization. Also, the availability of portable microcomputers will allow the instructor to take all the necessary equipment with him to the installation. The end product will be a JPLAN exercise running on a microcomputer which maintains the overall integrity and operating accuracy of the current mainframe version, with minor changes to allow the exercise to mirror JOPS more closely .

3. OBJECTIVES

The following objectives will be accomplished through the course of our project:

1. A database system will be designed to handle all of the JPLAN information necessary to run the simulation.

2. The Dialog Component (user interface) will be designed to allow users to interact with the system and run the simulation. Every effort will be made to eliminate the current "user hostile" interface. PC INGRES will be used as the database system and the 4th generation language (4GL) and forms generator provided will be used for implementing the user interface.

3. The simulation, which is presently written in FORTRAN IV, will be rewritten in Embedded SQL and Microsoft C and upgraded to more closely simulate the current JOPS system. The simulation should operate on any IBM PC compatible microcomputer system, and will be developed on a Zenith Z-248.

4. User documentation and instruction manuals will also be developed to aid in the operation of the system.

5. Maintenance and installation documentation for the system will also be provided to allow continued improvement and system maintenance.

4. THE DIALOG COMPONENT

To design the screens for the Dialog Component (user interface) a technique called storyboarding was used. Storyboards are used to represent the user's requirements using some type of graphics or form design tool to develop a sequence of displays which will represent a possible solution to the user's problem (Sprague and Carlson (1982)). A set of storyboards was designed for JPLAN using a graphics tool to illustrate the screens, system flow, and reports necessary to translate the FORTRAN based exercise to a PC based implementation. These storyboards were then reviewed by the users for accuracy and any changes were incorporated.

The Dialog component was determined to be best represented as a Menu Dialog because this type of dialog is effective for inexperienced and infrequent users (Sprague and Carlson (1986)). Since the users of this system are professional continuing education (PCE) students, Logistic's School students, and ACSC students, each of these groups (e.g., civilians, NCO's, and Officers) represent users who may be unfamiliar with computers in general. Therefore, the use of a highly structured and sequential menu system is appropriate. Also, this system mirrors the actual JOPS more closely by using the same module scheme.

The users will have a very limited exposure to the system (one day training and two days usage), so they will not gain a great deal of familiarity with it. To help them use and understand the system, each user is given a user's guide to what JPLAN is and the scenario of the exercise. The user's guide provides the users a basic understanding of what the acronyms stand for, their definition, and purpose. To answer any additional questions, the course director presents a more detailed explanation of JPLAN and answers any questions the users may have during the training session. The high degree of structure used in the dialog, plus the help screens will further assist the users in learning and using the system.

Storyboards were then implemented using the 4GL and Visual Forms Editor provided by PC INGRES. These tools allow the screens to be created quickly and are easily modified after the user reviews the initial prototypes. Before an sample screen can be discussed the data for the screen must be explained.

When a user develops his Force List, which is a list of all the units he wishes to deploy, the following data items are used to create the list:

- Line Number - uniquely identifies a particular deployment of a unit. A unit may be deployed more than once and to more than one POD. Since the line number uniquely identifies an instance of a force deployment it may not be reused if that deployment is deleted. This is because in "real life" line numbers are used for cross-referencing between commands and services (e.g., Air Force and Army, or Military Airlift Command (MAC) and Strategic Air Command (SAC)).
- UTC - Unit Type Code, uniquely identifies each unit available for deployment and has the attributes of DES, SVC, PERS, BPERS, PAX, STONS, and OSIZE.
- POD - Point of Debarkation, an abbreviation for the base to which the unit is being deployed.
- EMD - Earliest Movement Date, a date (−10 to 20) chosen by the user for the earliest departure of the unit. NOTE: Dates are in reference to D-Day, which is the day an operation begins on.
- LAD - Latest Arrival Date, a date (EMD to 20) chosen by the user for latest arrival of the unit.
- PRI - Priority, the relative priority of this force deployment with other deployments.
- DES - Description of Force Requirement, gives the user a short narrative on the units purpose.
- SVC - Service the unit belongs to: A (Army) and F (Air Force).
- PERS - The number of personnel assigned to that unit.
- BPERS - The number of personnel that will require base support. For example, for an army unit none of the personnel will stay on base because they will deploy to the surrounding area.
- PAX - The number of personnel that must be airlifted to the POD. This is usually the same as PERS, but could be different depending on the type of unit (e.g., some of the personnel may be flying aircraft to that base).
- STONS - The short tons (a unit of measure for generic cargo) of cargo belonging to the unit to be transported.
- OSIZE - the portion of STONS that must be transported by C-5A because of size requirements (e.g., Tanks, Trucks, and other heavy equipment).

In the old FORTRAN based implementation all user entries to the system were line oriented, which means that all entries for editing data were limited to one line. FORTRAN uses commas as delimiters for fields, therefore, users also had to insert commas between multiple fields on a line. To help illustrate the improvement of the new screen oriented user interface over the old FORTRAN based line entry system the two will be compared. For example, to change Line Number 119 to have a UTC of 4F9F9, a POD of PRM, a EMD of −7, and a LAD of −3 the following line entry in the old system would be required:

119,UTC,4F9F9,POD,PRM,EMD,−7,LAD,−3

If an error occurred in the line the entire line would have to be repeated. Also, the user had to remember the abbreviations for the fields he wanted to change, because as shown the abbreviation had to precede the new data value. Now one of the new screens (see Figure 1) will be used to illustrate the improvement obtained by the user interface we designed.

When the user brings up the "F40 - EDIT FORCE LIST" screen he will input the line number he wishes to edit and the corresponding information will be presented. The only information the user is allowed to edit are the UTC, POD, EMD, LAD, and PRI because this information affects each deployment of a particular force. Where the other information is constant regardless of the deployment date or location.

The information across the top of the screen (SAVE, QUIT, etc.) indicates the operations provided by this screen. Two modes of selection are provided, the user can either press the function key associated with the operation or press function key "F2" (predefined in PC INGRES) which will give the user a ring menu to select from. When activated the ring menu provides a highlighted bar which can be moved with the arrow keys to alternate choices. When the bar is placed on a choice (e.g., SAVE) an explanation is automatically given below the choice (e.g., Save the record to the Database).

```
[F3]=SAVE  [F7]=DELETE  [F1]=HELP  [F10]=QUIT

      F40 - EDIT FORCE LIST

      ENTER LINE NUMBER: 119

   UTC: 4F9FA            POD: PRM

   EMD: -7               LAD: -3

   PRI: 1                DES: FIRE/RESCU

   SVC: F                PERS: 15

   BPERS: 15             PAX: 15

   STONS: 1037           OSIZE: 165
```

Figure 1: Sample Menu F40 - EDIT FORCE LIST

Error checking is provided on each field the user must input to ensure the data is valid. If invalid data is entered an error message will be displayed and the user is returned to the field, the user will not be allowed to go to the next field until a valid input is received. Also, if the user tries to quit without saving the data he will be prompted to verify his intentions. Clearly this method is much more "user friendly" than the equivalent entry in the old FORTRAN implementation.

5. THE JOPS SIMULATOR

The model base structure and how the simulations, database, dialog, and user interact are shown in Figure 2. Note that the user does not interact directly with the models but uses the models as an extension of the database. This restriction exists because of the classroom environment the model is used in. To insure each group is using the same criteria, no group is allowed to manipulate the model in any way. The only interaction the users have with the model is through the dialog and any changes in the input parameters they make.

The simulation portion of the JPLAN exercise is the Transportation Feasibility Estimator (TFE). A TFE exists for both airlift (JPLAN and RADEX) and sealift (RADEX only), with the following description being oriented towards airlift because sealift is essentially the same (i.e., identical methodology and

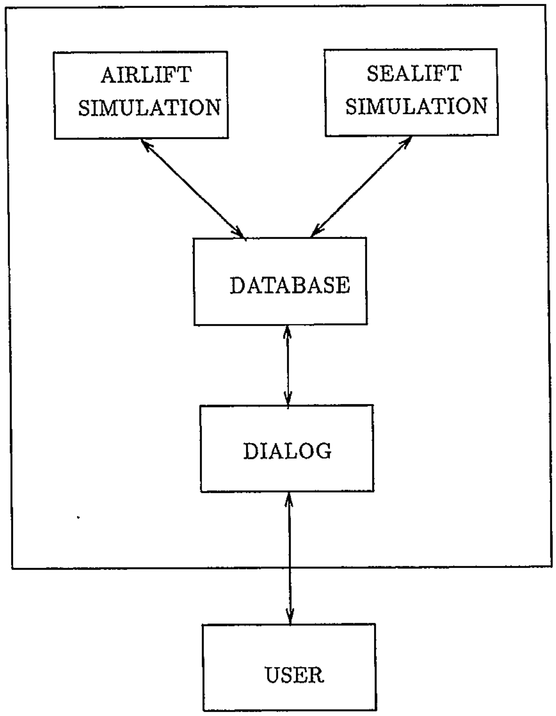

Figure 2: Data Flow - System Integration Diagram

purpose). The TFE takes the various unit logistics data (i.e., personnel and materials) and simulates the movement of those forces from their Point of Embarkation (POE) to the Point of Debarkation (POD) by using several types of airlift and sealift resources made available. Each UTC (Unit Type Code) is assigned an Earliest Movement Date (EMD), Latest Arrival Date (LAD), and Priority. This data along with POD and transportation data are used to simulate deployment and compute a Feasible Arrival Date (FAD). The FAD is a key result of the model because it indicates the feasibility of the current transportation plan. If all units arrive on time (i.e., each FAD on or before its LAD), the plan is considered transportation feasible (Joint Staff Officers Guide, AFSC PUB 1 (1986)).

Any late arrivals are shortfalls and can be resolved by the user adjusting the EMD, LAD, Priority, moving the UTC to another POD, or changing the POD Priority. Other considerations include plan requirements for force deployment and available resources (i.e., available aircraft, ships, ramp space for aircraft, and port capacity for ships) at each POE and POD.

More specifically the JPLAN simulation must perform the following steps one day at a time between the start and end days specified in the T10 Module: (Joint Operations Planning and Execution (1986))

1. Sort the Force List by LAD, UTC priority within LAD, and POD priority within UTC priorities.
2. If the force has Tactical Aircraft, ramp space is deleted at the POD (if available). This is because TAC aircraft fly to the base themselves, so ramp space is deleted up front.
3. Assign C-5As to carry outsized cargo.
4. If these C-5As are not full do the following in order:
   (a) Move outsized cargo from other UTCs with the same POD.
   (b) Move short tons of cargo from UTCs with the same POD.
5. Assign C-141s to carry UTC's short tons of cargo.
6. If all C-141s assigned are not full do the following in order:
   (a) Move short tons of cargo from UTCs with the same POD.
   (b) Move passengers from UTCs with the same POD.
7. After the preceding steps have been accomplished additional attempts are made to use any available aircraft resources which have not yet been assigned to move cargo and passengers left. The previous steps concentrated on moving cargo, and passengers only on those C-141s not filled with cargo. Now these steps are taken to move any remaining cargo that is not outsized by C-5A and remaining passengers:
   (a) Move short tons of cargo (other than outsize) with C-5As. Note: This differs from 4(b) because these C-5As do not carry any outsized cargo.
   (b) Move passengers with C-141s.
   (c) Move passengers with C-5As.
8. If any passengers remain use CRAF aircraft (any other available aircraft) to move passengers until CRAF resources expire.

This completes the simulation of one day's transportation of forces, this process is repeated for each day of movement. Upon assigning a load for a UTC to an aircraft the following checks are performed against data in the database:

1. A load cannot be assigned to an aircraft prior to the UTC's EMD.
2. The type of aircraft must be allowed at that POD.
3. The aircraft resources must be available.
4. Ramp space at the POD must be available.
5. The daily POD capacities for cargo and passengers must not be exceeded.

The following considerations apply to ramp space at a POD:

1. Once a unit with Tactical Aircraft deploy to a POD the ramp space required by those aircraft is decreased for remainder of the exercise.
2. When an airlift aircraft (e.g. C-5A, C-141, or CRAF) lands the ramp space required by that aircraft is taken for the entire day.

In summary, the simulation allocates transportation resources (i.e. aircraft and ramp space) to move selected forces into the theater of operation according to the base and unit priorities selected by the student. The primary limiting factor in the simulation is the amount of ramp space available at each POD to park aircraft. Since the student normally has plenty of aircraft available, the solution lies in the arrival timing and priorities of the forces.

6. DATABASE DESIGN

In designing the database structure a decision had to be made as to the structure of the data. The current system is implemented using flat files which causes a large number of redundant data items, as well as, compounding integrity and consistency problems among the duplicated data. To eliminate the data redundancies and enhance the integrity, consistency, and usability of the data, a relational database management system (DBMS) was selected. After an evaluation, the user selected PC Ingres.

An object-based design using entity-relationship (E-R) diagrams is used first to develop a conceptual view of the database. This view shows the data items with their associated attributes and the interrelationships between the data items. Then a relational database design methodology is used in implementing the conceptual view of the database.

The first step in designing the database is to identify the objects (entities). The JPLAN simulation references the following objects.

- Airbase
- Airlift Aircraft
- Ship
- Seaport
- Tactical Aircraft
- Time Phase Force Deployment Data (TPFDD)
- Type Units Data (TUCHA)

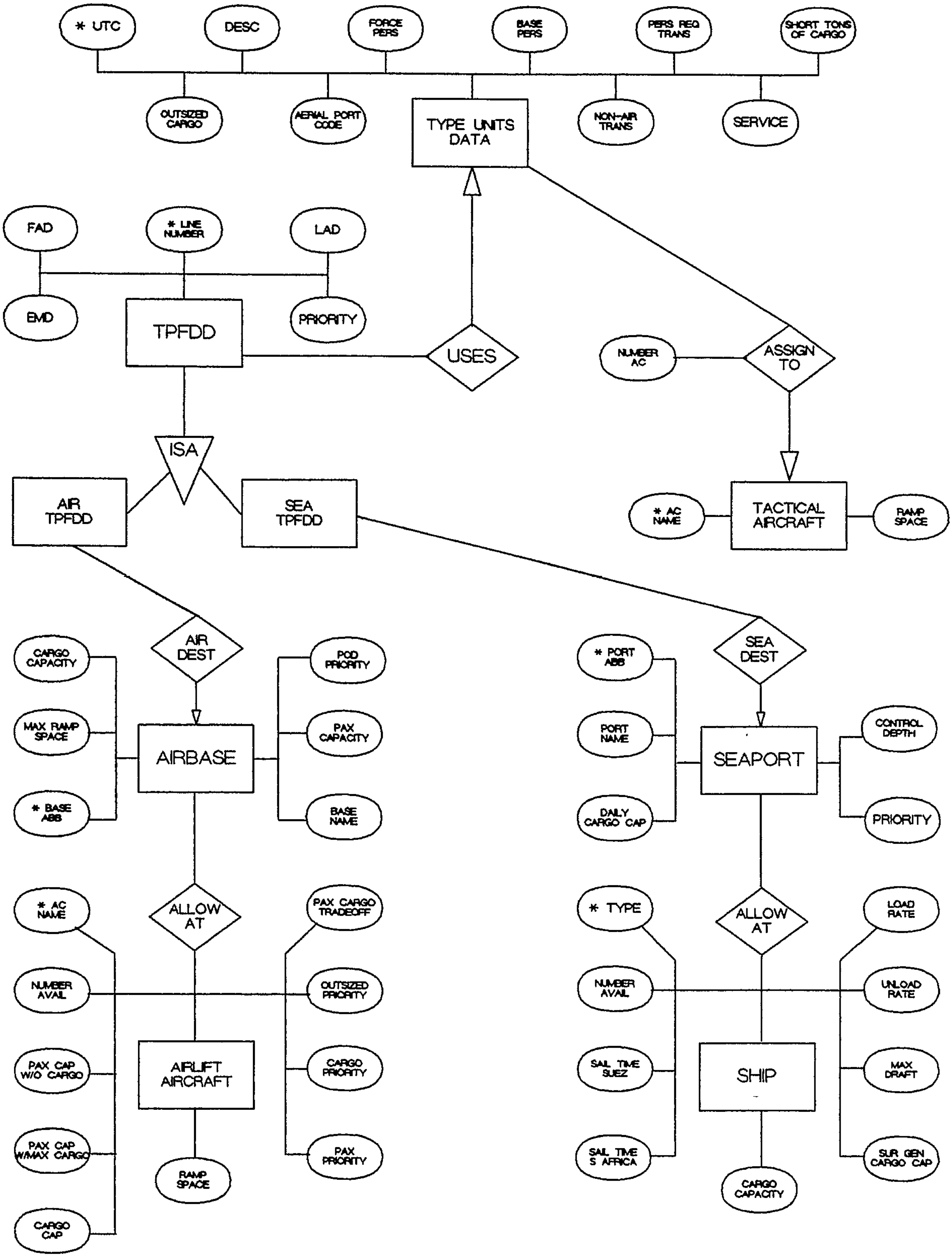

Figure 3: JPLAN Entity-Relationship (E-R) Diagram

After identifying the entities we identify the accompanying attributes. We also identify the variable type (e.g. integer, real, and character) at this step.

Now we determine the relationships between the various entities and draw the E-R diagrams. The following explanation applies to Figure 3. We were able to eliminate data redundancy in the FORTRAN flat file by eliminating duplicate attributes of (Description, UTC, Force Personnel, Service, Base Personnel, Personnel Requiring Transportation, Short Tons of Cargo, and Outsized Cargo) between the TUCHA and TPFDD entities since all of these attributes appear in the TUCHA entity. Therefore, we have the relationship of USES between the Type Units Data and the TPFDD because the TPFDD uses data contained in the TUCHA to form the Force List (see Figure 1). This illustrates one of the many benefits of using a relational database design to reduce data redundancy and enhance data integrity.

By using an ISA relationship to the Air and Sea TPFDD entity we can imply the attribute of Transportation Mode, thus saving space and increasing the logical structure of the data. We use the Air and Sea Destination relationships to imply the POD of the Force Deployment. Since not every UTC will consist of Tactical Aircraft, we elected to store this information in a separate entity to eliminate duplication of information and storage of empty fields (nulls) when the UTC doesn't have Tactical Aircraft assigned to it. Similarly, the Number of Aircraft is an attribute of the Assigned To relationship since a different number of aircraft may be assigned to UTCs with the same type of Aircraft.

We represent the PODs and transportation resources through the following entities and relationships. The Airlift Aircraft entity represents the various types of aircraft available to transport UTCs by air. The Airbase entity represents all the available Airbase Point of Debarkations (POD) available for UTC deployment. The Allowed At relation represents whether an Aircraft can land at that POD. UTCs from the Air TPFDD have a destination represented by the Air Destination relationship. The Seaport portion is essentially the same as the Aircraft section of the E-R diagram.

After all the entities are identified and the E-R diagram is completed, we determine the relationships between the objects and put them into database tables. Figure 4 illustrates two tables we constructed from the E-R Diagram, note that by using a relational database structure we can eliminate the unnecessary duplication of data described earlier. We use the UTC attribute to join the two tables so we are able to easily maintain the data shown in the F40 Edit Force List screen (see Figure 1) without keeping all of this information in one table and duplicating all of the TUCHA information for each instance of a unit deployment.

Type Units Data (TUCHA)

utc	pax	stons	...
3M1AC	159	154	
UFBDG	574	1005	
QFEBB	179	139	
7FVLB	304	449	

Time Phase Force Deployment Data (TPFDD)

line-number	utc	pod	emd	lad	...
1	3M1AC	BUN	-10	-7	
2	UFBDG	BAK	-10	-7	
3	QFEBB	MAX	-7	3	
4	7FVLB	PRM	-3	5	

Figure 4: Sample Database Tables Constructed From the E-R Diagram

7. Conclusion

We have described our approach to the design and implementation of a Joint Operations Planning Simulator for the microcomputer environment. Utilizing proven database design and software engineering techniques we have designed an exercise which will educate our future Air Force leaders in the complex activity of planning joint operations without burdening these students with out-of-date and inadequate computer resources.

ACKNOWLEDGEMENTS

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

REFERENCES

Joint Operation Planning and Execution (1986), Department of the Air Force, Air University, Maxwell AFB, Alabama.

Joint Staff Officers Guide, AFSC PUB 1 (1986), National Defense University, Armed Forces Staff College, Norfolk, Virginia.

Sprague, Ralph H. Jr. and Eric D. Carlson (1982), *Building Effective Decision Support Systems*, Prentice-Hall, Englewood Cliffs, New Jersey.

AUTHORS' BIOGRAPHIES

JAMES R. JANSEN 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 and Management Science from Metropolitan State College in 1981, and a M.S. in Systems Management from the University of

Southern California in 1984. His previous assignments included Instructor Crew Commander Minuteman ICBM for the 341st Strategic Missile Wing and Senior Crew Commander Squadron Command Post Minuteman ICBM for the 10th Strategic Missile Squadron.

James R. Jansen
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