FINAL REPORT

# THE ROLE OF AVIATION TECHNOLOGY IN THE CARIBBEAN BASIN

Robert W. Smith
University of Puerto Rico

## SUMMARY

Aviation technology can play a significant role in the development of the countries of the Caribbean Basin. This report focuses on a variety of applications of rotorcraft in the region and the benefits, both economic and political which can result from utilization of rotorcraft in both a public service and commercial role. A computer simulation of rotorcraft used as emergency medical vehicles is applied to compare and evaluate the advantages of using rotorcraft technology such as civil derivatives of the new tiltrotor. We conclude that by using a civil derivative of the tiltrotor significant improvements can be obtained in the level of health service in the region. We are currently engaged in an investigation of the potential for cargo and passenger transport applications for rotorcraft in the region and are developing a second computer simulation to be used to evaluate the potential benefits to be derived from such applications.

(NASA-CR-183069) THE ROLE OF AVIATION TECHNOLOGY IN THE CARIBBEAN BASIN Final Report, Sep. 1987 - Jul. 1988 (Puerto Rico Univ.) 99 p CSCL 01A

N88-26289

Unclas
G3/01 0149522

CARIBBEAN REGION
DEVELOPING NATIONS
ROTORCRAFT AIRCRAFT
COMPUTERIZED SIMULATION
TILT ROTOR AIRCRAFT

POLITICS
ECONOMICS
EVACUATING (TRANSPORTATION)
CIVIL AVIATION
PUBLIC HEALTH
MEDICAL SERVICE

INTRODUCTION

This study will concentrate on the islands and countries comprising an expanded definition of the Caribbean Basin (see figure 1 ); in addition to the islands of the Caribbean we will consider Central America, Mexico and the northern portion of South America. This area includes 25 independent states and four groups of dependent territories. This is the same region considered in a study of aviation in the Caribbean Basin carried out recently by the Center for Strategic and International Studies ( CSIS )(1). The population of this region is approximately 175 million with a total GNP of about 325 billion dollars and an average per capita income of approximately 1800 dollars ( see figure 2 ) . As mentioned in the CSIS study this region is important from both an economic and geopolitical perspective. This importance stems in part from the fact that 45% of all U.S. imports/exports pass through the Caribbean Sea and Gulf of Mexico as do 60% of NATO re-supplies and 55% of U.S. crude oil imports. (see figure 3) It is clearly important to the U.S. that this region improve its level of economic development and consequent political stability. We belong to a growing community who believe that an important impetus to growth in the region can be found in new developments in aviation technology. It is our contention that new developments in STOL and VTOL technology can play an important role in the economic development of the region. Rotorcraft, both conventional helicopters and new hybrid technologies now being developed such as variants of the U.S. Department of Defense V-22 Osprey tiltrotor can play a significant part in the development of an improved transportation infrastructure for the region. This report will concentrate on a review of promising applications of rotorcraft in the region with particular consideration of possible applications of the derivatives of the tiltrotor technology.

Before considering specific applications of rotorcraft in the region it is instructive to consider the general categories of Civil helicopter applications in the U.S. ( see figure 2 ). While public service applications enjoy a relatively high profile due to publicity and media interest in rescue operations, it is interesting to note that approximately 65% of Civil Helicopters in the U.S. are engaged in commercial applications. Careful analysis of potential commercial applications is clearly required. This fact is underscored further by the fact that whereas public service applications benefit and should by paid for by general segments of the population, commercial applications are more closely tied to concerns of immediate return on investment. While this report will consider general considerations and guidelines for commercial applications, the necessary more detailed analysis of cargo and passenger transport is an ongoing research project.

Figure 5 shows the geographic distribution and density of helicopters compared to population densities in the United States in 1980, (NASA contractor report NAS2-10411)(3). At that time the majority of the states had between 1 and 4 helicopters per million people. However these states were the most populous and industrially developed with the best system of roads. The more sparsely populated and mountainous states of the Rocky Mountains had between 11 and 32 helicopters per million people. Alaska is in a category of its own with over 200 helicopters per million people. The helicopter has many applications in such areas including medical evacuation, resource development and search and rescue operations. While no conclusion can be drawn from such a rough comparison, it is encouraging from the standpoint of potential rotorcraft applications that many of the distinguishing features of these high helicopter application regions are shared by the countries of the Caribbean Basin.

The geographical characteristics of the Caribbean Basin; numerous small islands, inaccessible areas, underdeveloped sea transportation and mountainous terrain with poor roads are all factors which make the region well suited to rotorcraft applications. Yet until recently the region has been slow to acquire helicopters in appreciable numbers. Though there have been considerable acquisitions in the last few years the region only operates approximately 1.5% of the world's helicopter fleet(1) (CSIS 1986). The apparent discrepancy between actual and potential use suggests the possibility of rapid growth of the rotorcraft fleet in the region.

In addition to basic commuter transport and air freight operations there are numerous applications for which rotorcraft technology is particularly well suited in the region. These applications include:

1) Search and rescue operations

2) Emergency medical services (EMS)

3) Border patrol and customs services

4) Drug interdiction

5) Disaster Relief

6) Rural electrification programs

7) Natural resource development ( servicing oil platforms)

8) Tourist transport to remote islands

9) Heavy construction projects in remote regions

Additional public service missions are listed in figure 6 from a NASA study carried out by Bell Helicopter Textron Inc.(4). These applications are distinguished from other commercial applications by the fact that the general public is the benefactor and the cost is assumed by the government agencies involved. Therefore in order to evaluate the cost benefit of these operations it is appropriate to consider the benefits and costs to society in as broad a sense as possible. Many of these applications are particularly pertinent for the Caribbean Basin region and underscore the strong market potential for a vertical takeoff and landing vehicle in the region.

The major hurdle of expanded utilization of rotorcraft in the region is operating cost. If an application can be carried out equally well by traditional fixed wing operations the rotorcraft will not be a viable head-to-head competitor in terms of operating cost. There are however a significant number of situations in which the vertical lift ability of rotorcraft provides the needed advantage and assures a place for rotorcraft.

## Passenger Services

In order for rotorcraft to be a viable alternative to traditional fixed wing air transport there must be a significant advantage provided by vertical take off and landing ability on at least one end of the trip. Island hopping and the ferrying of tourists to remote locations are obvious situations in which the VTOL ability of rotorcraft can provide the extra advantage. In high density congested areas the time advantage provided by using rotorcraft for inter-city rapid transit can provide the added advantage. In

ORIGINAL PAGE IS
OF POOR QUALITY

a current NASA report on civil tiltrotor applications (CR 177452), prepared by Boeing Commercial Aircraft Corporation, Bell Textron and Boeing Vertol, it is concluded that for travel distances between approximately 200 and 600 miles the Tiltrotor with its speed advantage over conventional helicopters and its VTOL advantage over conventional fixed wing aircraft will be able to capture a portion of the passenger service market. A significant number of passenger routes within the Caribbean Basin fall within this distance ( see figures 7,8 )(2). The above mentioned study concludes that approximately 25 commercial tiltrotors is a viable estimate for the passenger service market in the region by 1995. This estimate is based on a five percent penetration of a passenger service market which has grown at a 5% rate.

## Air Freight

Efficient economical transport of cargo to from and between the countries of the region is a necessity for economic growth in the region. Because of the vast number of islands and other natural geographic considerations air transport is well suited to serve this transportation need and to stimulate economic development. As concluded in the CSIS study(1) "..air transport, by furnishing the region with a low-cost, flexible and rapid method of transporting cargo, can play a crucial role in surmounting one of the most persistent obstacles to investment and growth in the Caribbean Basin - high transportation costs." The costs of air cargo in the region have been estimated at 17% of the value of the goods shipped. The region currently has a significant level of trade with the U.S., a large percentage of which is transported by air. See figures 9 and 10 for trade data(14).

The region has a significant number of secondary airports which, while not able to accommodate large aircraft, could service many newer aircraft which incorporate advances in airframe, engine and wing technology to provide relatively large cargo capacity and STOL (short takeoff and landing) ability. The remote areas of the region could be opened up to light manufacturing and agricultural production by the development of a transportation network incorporating local feeder networks using STOL or VTOL technology. The key economic questions hinge on a cost benefit analysis of the possible configurations and technologies involved. Some possible

ORIGINAL PAGE IS
OF POOR QUALITY

approaches include:

(1) Current turboprop aircraft servicing improved airport facilities coupled with ground transport connections.

(2) A feeder network based on the newer generation of high cargo capacity STOL aircraft.

(3) A system based on advanced technology such as the civil tiltroter providing point to point delivery between production facilities and major airports or other manufacturing facilities in the region.

In the tiltrotor applications study by Boeing Commercial Aircraft Corporation, Bell Textron and Boeing Vertol(2) it was concluded that based on projections of cargo transport requirements in the region ( excluding Mexico ) and probable percentage of penetration by versions of the tiltrotor into the high value cargo transport market it is unlikely that a significant number of tiltrotors will be required to service this market. Further analysis is needed in this area, particularly into the feasibility and probable market penetration of configurations based on the concept of using the tiltrotor to transport passengers by day and freight by night. A key question which must be considered is the cost benefit to air transport customers and subsequent increase in market penetration provided by the ability to forgo the time, expense, inconvenience and added probability of damage involved in ground transport to local airport facilities.

Natural Resource Development

There are numerous applications of rotorcraft in natural resource development including mining and logging operations, the fisheries industry, agricultural uses and servicing of offshore oil installations. The increased range and cargo capacity of the tiltrotor insures that it will play a key role in such applications. In a NASA report on civil tiltrotor applications, prepared by Boeing Commercial Aircraft Corporation, Bell Textron and Boeing Vertol(2), it is estimated that the 221 offshore oil platforms in the Gulf of Mexico and Caribbean may eventually require up to 170 tiltrotors. While this may be the major application in this category the advantages of the tiltrotor will also insure its place in other areas of resource development.

## Public Service

We will now narrow our discussion of applications of aviation technology in the Caribbean region to the public service sector with emphasis on rotorcraft applications. The relative inaccessibility of much of the region again is a key factor supporting rotorcraft applications in this area. As stated previously most applications in the public service area tend to benefit broad segments of the population and should therefore be supported by the governments or agencies involved where possible. It seems reasonable therefore to assume as global a view as possible when analyzing the costs and benefits of a given application. One of the current issues facing Emergency Medical rotorcraft operations in the United States is the question of payment for what is unquestionably an expensive service. According to an ASHBEAMS survey in Rotor and Wing International, Nov. 1982 the average direct cost of an Emergency Medical Helicopter is $630,000 per year compared to net Revenues from patient charges of $162,000, or 25.7% of the direct operating cost. There is an added benefit to the hospital of increased utilization which may or may not make up the difference. In this as well as other public service applications it is necessary to arrive at some consensus concerning who should pay and who benefits from the given application. This is critical in a less affluent region such as the Caribbean Basin.

In the NASA report on civil tiltrotor applications, prepared by Boeing Commercial Aircraft Corporation, Bell Textron and Boeing Vertol(2), the following operational benefits and requirements were noted for public service applications:

(1) Law Enforcement: For general law enforcement a 6-10 passenger tiltrotor is needed, for drug interdiction a larger 12-16 passenger tiltrotor would be optional. Law enforcement applications would benefit from the greater speed, range and operational flexibility of the tiltrotor.

(2) Coast Guard search and rescue: For search and rescue operations a 6-10 passenger tiltrotor configuration would be generally most suitable. The tiltrotors advantages over conventional helicopters in this application include longer range, greater speed, a more stable platform, and milder

downwash.

(3) Medical transport: For medical transport operations a 6-10 passenger tiltrotor configuration is needed. Medical transport operations would benefit from time savings and reduced transfers.

## Law Enforcement Operations

There are many advantages to the use of rotorcraft in law enforcement operations. In the last few years the number of helicopters used by law enforcement agencies and particularly drug enforcement units has grown markedly. A new drug enforcement unit in Puerto Rico has recently acquired several helicopters. For general law enforcement operations in the region the tiltrotor would be advantages for long-range prisoner transfer, high-priority personnel movements and other applications where the speed and range would prove most beneficial. The versatility of the tiltrotor would be important because of the diverse applications required by law enforcement agencies. While the tiltrotor may not replace the helicopter in many applications such as intra-city patrols and ground unit support it could play a major role where its larger capacity, speed and range could be utilized. Perhaps the most important area of law enforcement in which the tiltrotor should prove effective is drug interdiction. The larger size, greater speed, range and endurance of the tiltrotor would be sufficiently important in this type of operation to make the tiltrotor the ideal vehicle for drug enforcement agencies of the region.

## Search and Rescue

The United States Coast Guard operates bases located on the Northwestern coast of Puerto Rico and in Florida. The primary missions are providing protection for vessels operating within the 200 mile offshore coastal region surrounding the U.S. and drug interdiction operations. Currently the missions are accomplished using a combination of HU-25 Falcon jets for long range reconnaissance and HH-65 and HH-3 Helicopters and Coast Guard Cutters for rescue operations and interdiction.

The tiltrotor with its longer range and greater speed is an ideal vehicle for sea rescue operations. In addition the tiltrotor offers a more stable platform and milder downwash than the helicopters currently in use. Currently the 300 nmi radius limitation of the HH-3 helicopter results in only 73% of the incidents being within range. The longer range missions must be reached by cutters and other long-range craft. The longer range tiltrotor could significantly increase the number of incidents which can be reached quickly and could consolidate some multiple aircraft requirements into a single unit, thus resulting in possible cost savings.

Disaster Relief

Disaster relief includes most of the applications involved in search and rescue, emergency medical services and law enforcement operations. In the advent of a natural or man-made disaster, people and relief supplies must be transported into and from the affected area. The tiltrotor with its speed, range and versatility is ideally suited to a major role in relief operations. In a major disaster tiltrotors could be drawn from police, military and even commercial applications for service during the relief operations.

Emergency Medical Services

Hospital based Emergency rotorcraft can play a significant role in improving the level of health care available to the population. An efficient system of HEMS centers can reduce response time and provide more effective health care to the region. The obvious benefits include the reduction of the number of deaths, permanent disabilities, and length of hospital stays.

In the US the accident death rate in rural areas where quick access to large trauma centers is not available is four times the urban death rate. The number of persons killed per year in the U.S. by trauma is approximately 115,000. Traditionally a large percentage of EMS missions involve trauma cases. Trauma affects primarily young productive persons. The cost to society in the US is estimated at $41 billion annually. For most trauma cases the probability that death will occur within the first 25 minutes is 67%. It is estimated that the average response time can be reduced by between 30% to 80% by helicopter rescue services, and that mortality can be reduced by approximately 50% if patients are rapidly transported to a trauma center. These facts underscore the importance of a well developed system of emergency medical transfer of critically ill and trauma patients.

The countries of the region have varying levels of health services but there is a universal need to improve the level of health care. In the least developed countries such as Haiti, the most urgent need is for elementary health care facilities and the corresponding infrastructure to provide basic health care to the population. Other countries including parts of Central America, Venezuela, Jamaica, and the Dominican Republic have a more developed system of health care, at least in the more populous regions. These larger countries have the population and economic base to support a system of helicopter based emergency medical centers within their borders. We feel that once economic and political barriers are overcome, a rapid growth of such centers will parallel the growth of such centers in the United States. The first successful center will serve as an example of the feasibility of the concept and should act as a catalyst for further development in the region. An attractive location for such a center is the Lesser Antilles.

## NEED ANALYSIS

In the United States trauma is the major cause of death for persons under 40 years of age. In addition to accidental death caused by motor vehicle accidents, drownings, falls, burns, poisonings and firearms, many thousands die prematurely from critical medical illnesses that did not receive urgently needed medical attention. The two most critical factors determining a trauma victim's chance of survival are rapid transport to a medical facility and the ability of the medical facility to provide a high level of expert emergency health care. "Survival is directly proportional to the ability of the trauma system to respond to the accident with adequate and appropriate care, and is inversely proportional to the severity of the initial injury and to the square of the time lapse between the injury and stabilization of the unstable patient"(5). Small emergency rooms in community hospitals are seldom well equipped and unless the attending physicians have the experience born of a relatively high volume of trauma cases they will not be able to provide optimal care. It is estimated that 40 cases per month are required to keep the necessary skills.

There has recently been much improvement in emergency health care in the United States. The number of states served by HEMS has nearly doubled in the last five years. This rapid growth has been due to the recognition of the inadequacy of the trauma care provided by small emergency rooms in which the staff have neither the experience nor the facilities to provide adequate service. The countries of the Caribbean Basin are characterized by inadequate transportation of trauma victims and small ill-equipped emergency facilities.

The existing level of services on the majority of the islands in the region indicates a need for improvement. However, in order to optimize resource allocation and insure fiscal integrity, it is important to analyze the potential demand for HEMS in the region. A common concern of all new HEMS centers is the prediction of the number of patients who will require helicopter transfer. This is important from both an economic as well as professional viewpoint. A high volume of patients is necessary to maintain the required level of experience and expertise of the attending staff. Several methods of predicting usage have been utilized. The most common methods range from simple guidelines based on population served, to detailed analysis of emergency room data. No method has proved completely satisfactory since estimates based solely on population ignore many important factors and complete emergency room data is a rarity even in the United States. A separate analysis will be carried out for the island of Puerto Rico since more data is available for Puerto Rico and the island is demographically more similar to the United States than the rest of the region.

The simplest method of estimating helicopter utilization mentioned in the literature is based solely on the size of the population served. The average number of helicopter transports per 100,000 population is approximately 31 per year in the United States. See table 11 for an average HEMS profile. There has traditionally been a learning period during which the population is educated as to the availability of HEMS service and the primary responders to emergencies are trained to recognize how and when to request helicopter transfer. Consequently, during the first year of operation the call rate is often as much as 50% less than the rate of a more mature center. Table 12 indicates the diversity of experience of several HEMS centers during their first year of operation and underscores the approximate nature of any estimate based solely on size of population served.

It is generally true that a center located in an isolated area will experience a greater demand rate for helicopter transfer than a center located in a densely populated metropolitan area. The apparent reason being that in a metropolitan area in which much of the population is located relatively close to an emergency center, rapid transport can usually be provided by ground vehicles. In the more isolated regions ground transport is often not a viable option if speed and smooth ride are important factors, as they are in most trauma situations. This observation is clearly pertinent when considering the numerous small islands of the Caribbean Basin and would tend to suggest that a higher percentage of trauma patients would benefit from helicopter transfer in this region than in most areas of the United States. Another important factor which would tend to differentiate the Caribbean Islands from the United States is the fact that a high percentage of HEMS requests in the United States arise from motor vehicle accidents. In the estimate of helicopter transfer requests by Rhee et Al.(6), approximately 50% of the requests were related to traffic accidents. While the motor vehicle accident rate in Puerto Rico is similar to the United States(7), this may not be the case for the rest of the countries. In an attempt to counteract this difference, we will base motor vehicle accident estimates for the Lesser Antilles on the number of motor vehicles rather than population size.

## Benefits of HEMS Centers

The obvious benefit provided by HEMS centers to the region will be the reduction of loss of life and injuries due to trauma and major medical emergencies. While it is difficult to place a dollar value on a human life it is important to estimate the savings to society in order to put the considerable cost of establishing such a system in perspective. It is necessary to consider the savings from the point of view of society as a whole and not just the profit or loss of the individual operator involved.

In the US the National Health and Traffic Safety Administration estimates the value to society of a human life to be over $200,000. This estimate was for 1980 when the per capita GNP of the United States was approximately $12,000. Using the average per capita GNP of the Caribbean of approximately $1,800 to scale the National Health and Traffic Safety administration's estimate to the region yields an estimate of $30,000 per life saved. While clearly a crude estimate, the indication is that even in this relatively poor region an emergency medical rotorcraft with an operating cost of $630,000 would have to save 21 lives per year in order for its operating cost to be offset by its benefit to society. In the United States it is estimated that an EMS rotorcraft saves approximately 9 lives per 100,000 population served. So for a service region containing as little as 1 million people an EMS rotorcraft would save the society as a whole more than four times as much as its operating cost. This would vary according to the relative wealth of the country but on the average there is clear economic justification from the viewpoint of savings to the economy of the region.

## HEMS Utilization Estimate for Puerto Rico

In order to estimate the potential utilization of HEMS for Puerto Rico, we rely on motor vehicle accident data for Puerto Rico provided by the U.S. Department of Transportation(7). We assume other causes of trauma are similar to the United States. As in the United States and the majority of the Caribbean islands, heart disease is the major cause of death(8). In the United States a major source of emergencies after traffic accidents is myocardial infarction with a very high deaths-per-incident rate on the order of 40%(9). We will estimate the number of myocardial infarctions by applying the U.S. average of .0075 heart attacks per person per year(10). The following analysis is for the first year of operation, consequently after an HEMS center has matured it would be reasonable to expect the rate of utilization to double. In our analysis we follow the method of estimation and apply some of the frequency and utilization rates presented by Rhee et al.(6) in which an analysis of HEMS demand was developed for southern Michigan. The corresponding utilization rate may be different for Puerto Rico but the resulting estimate should serve as a first approximation.

In the United States motor vehicle accident (MVA) trauma is one of the major causes of accidental death. While between 30% and 50% of MVA deaths occur almost instantaneously(11), approximately 20% of the victims are potentially salvageable if transported quickly to an adequate medical facility(12). Approximately 50% of those victims who eventually die are taken to an emergency facility(6) This information along with records of the number of motor vehicle accident deaths in Puerto Rico can be used to estimate the number of patients who would benefit from HEMS transfer. The number of traffic fatalities in Puerto Rico for the period between 1978 and 1983 has averaged 516 per year(7). We estimate that 70% of the population of Puerto Rico is served by basic or less than basic emergency services and the remaining 30% are located in areas served by a major medical facility. Following the analysis presented by Rhee et al.(6), we estimate that 50% of the fatal accident victims located in areas with basic or less than basic emergency services needed HEMS and 5% of those located in the major service area would have benefitted from HEMS. This analysis indicates that 516*0.70*0.50=180.6 of those accident victims in areas with basic or less than basic medical service needed helicopter transfer. Similarly 7.74 victims in the major service areas needed HEMS.

Continuing the analysis presented by Rhee et al.(6) in a system in which patients are taken to the nearest hospital and then transferred, actual transfer might occur in 25% of the cases. This leads to an estimate of 47.09 HEMS requests for those motor vehicle accident victims who would otherwise die. A similar analysis is carried out for HEMS calls for an estimate of the number of traffic accident victims who are not likely to die as well as for victims of falls, spinal cord injuries, burns, myocardial infarction, cerebrovascular accidents and pediatric and other emergencies. The results are summarized in table 13 in which we estimate approximately 442 HEMS calls for the first year of operation of a HEMS center located in Puerto Rico. This compares favorably with the national average of 370 calls per year for the first year of operation of centers within the U.S. mainland. However, this is not our final estimate for such a center since we have not yet included several sources of HEMS calls. The above analysis is based on a service area comprising only the island of Puerto Rico. If a vehicle with an effective range of 300 miles is used, the population served is approximately three times that considered in the above analysis. Our initial estimates clearly indicate that there will be sufficient demand to support a HEMS center on the island of Puerto Rico.

## HEMS Utilization Estimate for the Lesser Antilles

In order to estimate HEMS utilization for the islands of the Lesser Antilles, the islands between Puerto Rico and Venezuela, we will use much the same techniques as for Puerto Rico. However, for these islands we do not have such complete motor vehicle accident data and will have to estimate the number of fatal traffic accidents as well as the number of motor vehicle accident survivors. We do this by interpolating from the number of motor vehicles on the islands rather than from the population. Once again, this will only provide a rough estimate but we feel any error is on the conservative side given the congestion and poor road conditions found in most of these islands. Another difference is that since we envision the establishment of only one HEMS center in the southernmost islands, approximately 98% of the total population of the Lesser Antilles will be considered to be located in an area with basic or less than basic emergency service. The 98% estimate is derived by assuming that the HEMS center will be located on the island of Grenada and approximately 50% of the population of Grenada is sufficiently remote from the capital to be considered in a basic or less than basic medical service area. After making these assumptions and applying an analysis parallel to that for Puerto Rico, we estimate that during the first year of operation there will be 342 calls for HEMS from the islands of the Lesser Antilles. See table 14 for a summary of these calculations. As stated previously, the number of HEMS calls will increase as the center becomes better known and the communication system improves.

The above estimates of demand are sufficient to support a second HEMS center located in one of the southern islands. That our estimates of HEMS demand are probably somewhat conservative can be seen by considering that use of the rule of thumb estimate of 31 calls per 100,000 population served would lead to an estimate of 600 calls per year for a center located in Grenada with an operating range of 150 miles. Note from table 15 that there are approximately 2 million people within a 150 mile radius of Grenada.

## HEMS Utilization Estimates for the larger islands and countries of the Caribbean Basin

For the remaining countries of the Caribbean Basin a similar analysis was carried out using estimates based on population size and the density of urban populations. Table 16 gives a summary of the results of these estimates. As in the previous cases these estimates are based on percentages derived from experience in the United States and therefore to be taken as only rough approximations. As in the case of the estimate for the Lower Antilles the errors are most likely on the conservative side. It is rather clear that most of the countries would have sufficient demand to support one or more HEMS center. The real issue is the affordability of establishing such centers. We feel however that as the benefits of such centers become better appreciated the necessary economic and political pressure will result in the eventual establishment of a system of HEMS centers throughout the region. We will now consider the relative advantages that new technology such as the Tilt-Rotor could contribute to such a system.

## The Tilt Rotor as an Emergency Medical Vehicle

Bell Helicopter Textron has developed a new aircraft which we feel is destined to constitute a milestone in emergency medical rescue technology. The Tilt Rotor is an aircraft which can be flown either as a helicopter or as a fixed wing, combining the versatility of one with the speed and range of the other. The transition from one mode to the other is done smoothly in a matter of twelve seconds. The implications for medical rescue operations are obvious.

Since its use in the Korean War, the helicopter has shown the importance of quick response time in emergency rescue operations. But the helicopter has neither the speed nor the range of a fixed wing aircraft. The Tilt Rotor combines the speed and range of a fixed wing aircraft with the ability to fly directly to the scene of an accident and back to the medical facility. This new technology is not just in the initial planning stages; two XV-15 Tilt Rotor prototypes have been flying since 1977 and a commercial version of the Tilt Rotor should be on the market in less than ten years.

The advantages of the XV-15 are illustrated with a speed envelope comparison. In figure 11 we see that the XV-15 is capable of flying 300 MPH, fully twice the speed of current emergency medical helicopters. The range of the XV-15 is equally impressive with an effective range of approximately 700 miles. (See figure 17). The cabin dimensions of the XV-15 are roughly 1.5 * 1.5 * 4 meters. This spacious cabin can be fitted for a multi-patient, casualty evacuation role or it can be tailored to serve as a fully equipped flying emergency medical facility.( See figure 18 ).

The initial cost of the Tilt Rotor will be higher than either a helicopter or fixed wing aircraft; the current cost projections are approximately 10% more than a helicopter of the same passenger size. There will be many situations where the Tilt Rotor will be very cost effective. This is due mainly to the fact that the Tilt Rotor can go twice as far and twice as fast as a helicopter on the same amount of fuel(13). The Tilt Rotor will be exceptionally useful and cost effective in situations where it is necessary to cover large sparsely populated areas. The higher initial cost must then be measured against the benefit of serving a large region with a single medical facility. This situation is found in the Caribbean Basin as well as in many sparsely populated areas of the world.

## Computer Simulation of HEMS

In an attempt to analyze the results of locating HEMS centers in the Caribbean Basin, a finite event Monte Carlo simulation of HEMS centers was developed and run for various configurations of locations and helicopter types. The program generates accidents in a given region according to criteria input by the user and then calculates the time required to provide assistance to each accident using the closest available aircraft. The availability of an aircraft depends on its location, range, speed, and whether or not it is currently being used for a previous rescue or is out of service due to repairs or bad weather. The program accepts as input the locations and categories of hospitals, the number and capacity of helicopters at each hospital, the region in which a given percentage of accidents will occur, and other data relating to rescue time, response time, etc. The program was initially developed on a Macintosh micro-computer and makes extensive use of graphics for both input of data and illustration of the simulation. The output of the program includes the average wait time before an accident victim is reached, the average rescue time before the victim is taken to the nearest appropriate hospital, the number of accident victims which were not rescued due to the lack of an available helicopter, and the number of hours per week that each helicopter spends flying. Figures 19, 20 and 21 illustrate the results of three simulations.

In one case, the trauma centers were located on the islands of Puerto Rico and Grenada. A helicopter was located at each of these trauma centers and an additional helicopter was located on the island of Guadeloupe to provide transport from the middle islands to either of the two trauma centers. The helicopters were assumed to have a speed of 150 MPH and an effective range of 150 miles. This speed and range correspond to the limitations of the HEMS helicopters currently available. The third helicopter located in the middle islands was necessary in order to cover the entire region. We did not assume a third trauma center in the middle islands, however, since the population of the region would not currently support a full trauma center; at this site we located a level two center and assumed that transfer to one of the full trauma centers would be necessary for the majority of the rescue operations.

In the second case we simulated two trauma centers, one located in Puerto Rico and the other in Grenada. In this case we assigned one helicopter to each center with a speed of 300 MPH and an effective range of 300 miles. This speed and range correspond to the capacity of a proposed version of the new XV-15 Tilt Rotor. In this case a third helicopter is not necessary since the increased range of the Tilt Rotor enables two vehicles to cover the entire region.

A comparison of the results of the two simulations underscores the advantage of using the two Tilt Rotors rather than three helicopters. In the simulation using two Tilt Rotors the average rescue time was 100 minutes verses 125 minutes for the system of three standard helicopters. The average time before arrival of the rescue vehicle was 34 minutes using Tilt Rotors and 32 minutes using helicopters. The percentage of out of range calls was also reduced by using the Tilt Rotors. While these results are tentative and no firm conclusions should be inferred from this initial comparison, it does point out some of the potential advantages to be derived from the new Tilt Rotor technology.

In figure 21 we show the results of running a simulation for the entire region. In this case we located tiltrotors in the same locations as the previous simulation and located conventional helicopters in major population centers of the remaining countries of the Caribbean Basin. According to this simulation good coverage of the entire region is possible using twenty HEMS centers. While the average rescue times and percentage of missed calls is not as good as for the previous simulation, with a relatively small number of centers the majority of the population could be in range of helicopter rescue service.

## CONCLUSION

It is clear that there is a need for the level of medical service which could be provided by strategically located HEMS centers in the Caribbean Basin. In addition to improving the quality of health care available for residents and tourists in the area, a system of helicopter based medical centers would foster a sense of cooperation between the islands of the region as well as with the United States. Such a system will also serve as an example and training site for other countries in the region which have both the resources and the need for such medical services. If appropriate sources of funding and guidance are provided by the United States a lasting and highly visible source of goodwill will have been set in motion. We feel that the benefits of such a system warrant further serious consideration.

## REFERENCES


1. Center For Strategic and International Studies, Latin American Program: On The Wings Of Development: Aviation And The Caribbean Basin, (final draft). December 1987.

2. Boeing Commercial Airplane Co.; Bell Textron, Boeing Vertol: Civil Tiltrotor Missions and Applications: A Research Study. NASA ARC; July 1987 ( NASA CR 177452 ).

3. Newman, R.B.; Adams, R.J.: Study of Helicopters In Public Service, Vol II, Expanded assesment of Emergency Air Medical Services. NASA ARC; August 1981 (NASA CR NAS2-10411).

4. Williams, L.N.; Koehn, P.D.; Sugarek, S.E.: The Emergency Medical Service Helicopter Design Study. NASA ARC; July 1985 (NASA CR 699-099-179).

5. Symbas, P.: The Principles and Practice of Emergency Medicine. Schwarts, G.R.; Safer, P.; Stone, J.H.; Story, P.B.; Wagner, D.K.; and W.B., eds. Saunders, Co., Philadelphia, 1978.

6. Rhee, Kenneth J.; Gurfield, R; Sidel, V.W.; Action, J.; Lown, B.; Stephany, S.J.; Frey, C.; Huel, K.D.; Symbas, P.; and Borland, Robert R.: Predicting the Utilization of Helicopter Emergency Medical Services: An Approach Based on Need. Annals of Emergency Medicine, vol. 13, no. 10, Oct 1984, pp. 916-923.

7. U.S. Department of Transportation: Fatal Accident Reporting System 1983. Washington, D. C. 1983.

8. U.S. Agency for International Development, Office of Foreign Disaster Assistance: Countries of the Caribbean Community, A Regional Profile. Washington, D. C., 1982.

9. Gurfield, R.: Expected Medical Emergencies, an Index of Need for a Region. JACEP, vol. 6, 1977, pp. 94-99.

10. Sidel, V. W.; Action, J.; and Lown, B.: Models for the Valuation of Prehospital Coronary Care. Am. J. Cardiology, 1968, p. 679.

11. Stephany, S. J.: An Evaluation Methodology for Multi-element Emergency Medical Services Systems. IEEE Transaction on Vehicular Technology, vol. 6, 1976, pp. 128-141.

12. Frey, C.; and Huel, K.D.: Resuscitation and Survival in Motor Vehicle Accidents. J. Trauma, vol. 9, 1969, pp. 292-310.

13. Borland, Robert R.: The Tilt Rotor Aircraft in the Medical Role. Bell Helicopter Textron, Inc. staff report, 1985.

14. International Civil Aviation Organization: International Air Passenger and Freight Transport; Latin America and the Caribbean. Circular 175-AT/65, 1983.

FIGURE 1. COUNTRIES COMPRISING THE CARIBBEAN BASIN


ORIGINAL PAGE IS
OF POOR QUALITY

FIGURE 2. SUMMARY DATA FOR COUNTRIES OF THE CARIBBEAN BASIN

| Country | Population | GDP/GNP Billions | Per Capita | Exports (Billions) | Imports (Billions) |
|---|---|---|---|---|---|
| Antigua/Barbuda | 82,000 | $ 0.171 | $2,085.990 | $ 0.044 | $ 0.159 |
| Bahamas | 235,000 | 1.949 | 8,292.275 | 2.490 | 3.248 |
| Barbados | 253,000 | 1.247 | 4,929.479 | 0.422 | 0.710 |
| Belize | 168,000 | 0.201 | 1,199.300 | 0.101 | 0.136 |
| British Virgin Islands | 12,000 | 0.087 | 7,210.548 | 0.003 | 0.074 |
| Cayman Islands | 22,000 | 0.087 | 3,933.026 | 0.027 | 0.157 |
| Costa Rica | 2,714,000 | 3.681 | 1,356.245 | 1.035 | 1.192 |
| Dominica | 74,000 | 0.092 | 1,249.382 | 0.028 | 0.061 |
| Dominican Republic | 6,785.000 | 11.909 | 1,755.141 | 0.938 | 1.516 |
| El Salvador | 5,105,000 | 4.552 | 891.601 | 0.824 | 1.002 |
| Grenada | 86,000 | 0.094 | 1,095.191 | 0.021 | 0.063 |
| Guadeloupe | 334,000 | 1.626 | 4,868.983 | 0.112 | 0.702 |
| Gautemala | 8,600,000 | 9.604 | 1,116.784 | 1.236 | 1.407 |
| Guyana | 771,000 | 0.432 | 560.257 | 0.230 | 0.240 |
| Haiti | 5,870,000 | 1.949 | 331.974 | 0.189 | 0.332 |
| Honduras | 4,648,000 | 3.464 | 745.337 | 0.759 | 0.792 |
| Jamaica | 2,288,000 | 2.165 | 946.331 | 0.764 | 1.191 |
| Martinique | 328,000 | 1.902 | 5,798.397 | 0.154 | 0.881 |
| Montserrat | 12,000 | 0.036 | 2,996.591 | 0.002 | 0.022 |
| Netherlands Antilles | 236,000 | 1.517 | 6,428.069 | 4.944 | 5.057 |
| Nicaragua | 3,342,000 | 3.027 | 905.881 | 0.334 | 0.887 |
| Panama | 2,227,000 | 4.763 | 2,138.954 | 0.454 | 1.451 |
| Puerto Rico | 3,300,000 | 14.462 | 4,382.515 | 10.226 | 10.675 |
| St. Christopher, Nevis | 40,000 | 0.070 | 1,741.769 | 0.035 | 0.053 |
| St. Lucia | 123,000 | 0.160 | 1,302.644 | 0.056 | 0.120 |
| St. Vincent & the Grenadines | 103,000 | 0.100 | 970.983 | 0.047 | 0.080 |
| Suriname | 381,000 | 1.191 | 3,125.624 | 0.385 | 0.375 |
| Trinidad & Tobago | 1,204,000 | 9.310 | 7,732.876 | 2.382 | 2.057 |
| Turks & Caicos Island | 7,000 | 0.021 | 2,953.221 | 0.004 | 0.025 |
| US Virgin Islands | 102,000 | 0.773 | 7,579.934 | 4.045 | 5.281 |
| Total | 49,452,000 | $80.643 | $1,627.093 | $32.289 (16.8 US) | $39.946 (16.2 US) |

FIGURE 3. MOVEMENT OF GOODS THROUGH THE CARIBBEAN BASIN

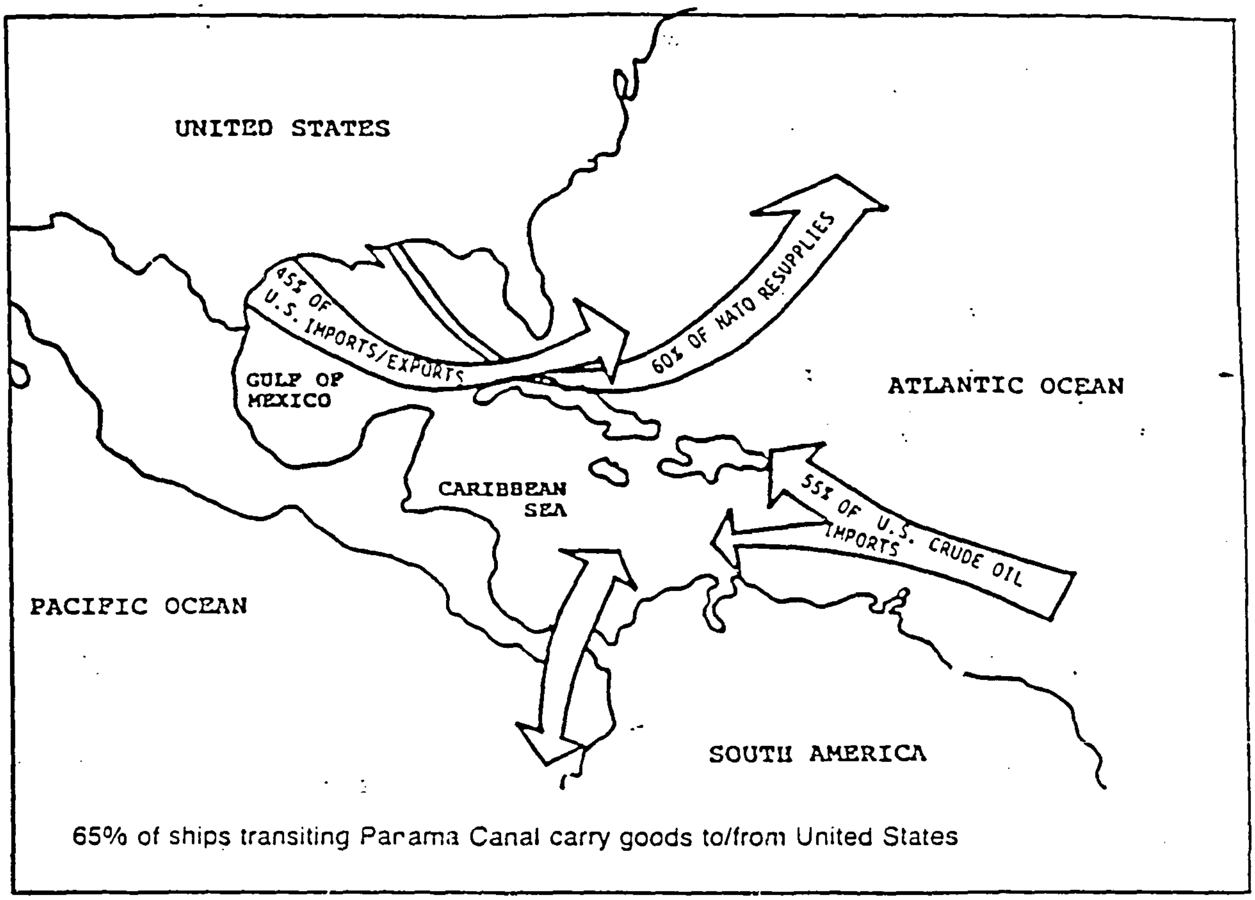
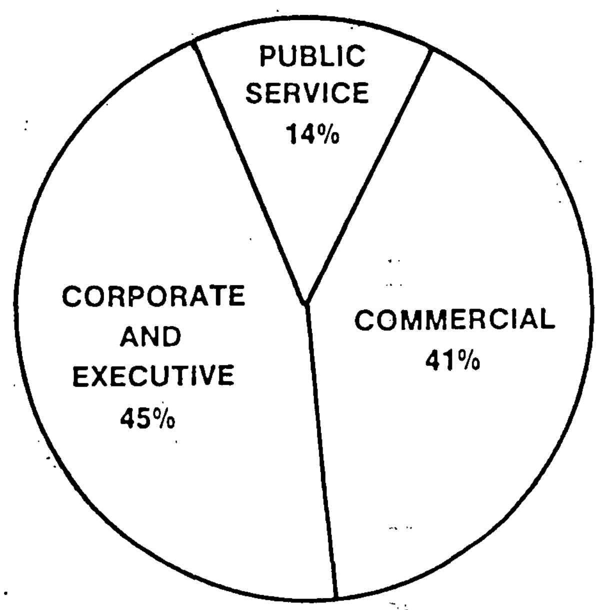
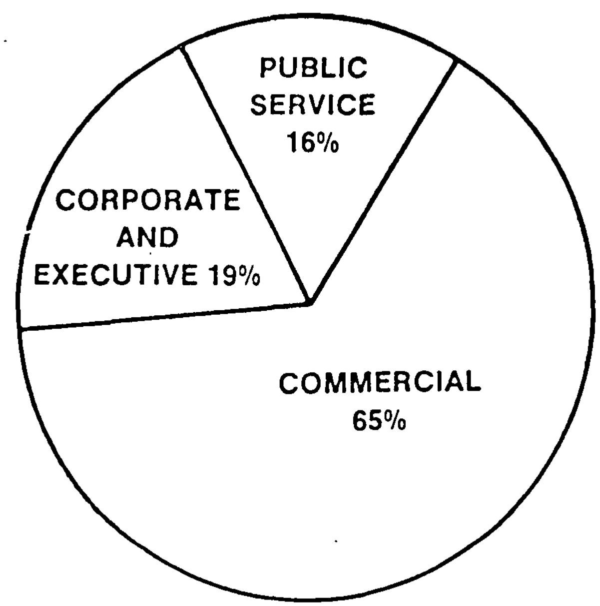


FIGURE 4. HELICOPTER DISTRIBUTION BY PRIMARY MISSION




Distribution of Civil Helicopter Operators by Primary Mission




Distribution of Civil Helicopters (No. of Aircraft) by Primary Mission

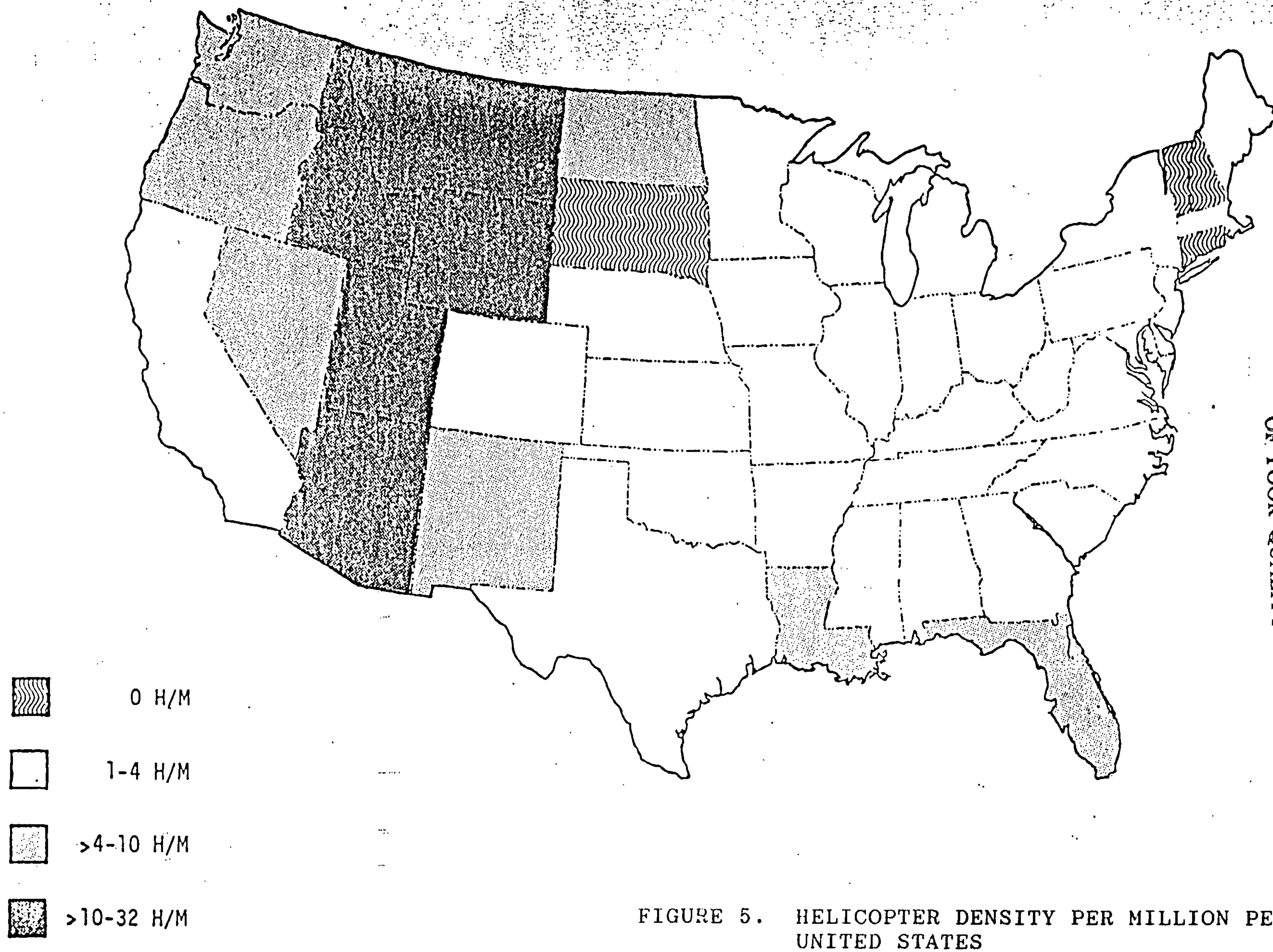



FIGURE 5. HELICOPTER DENSITY PER MILLION PEOPLE IN THE UNITED STATES

H/M = helicopters per million people

ORIGINAL PAGE IS
OF POOR QUALITY

FIGURE 6. PUBLIC SERVICE HELICOPTER MISSIONS

## PUBLIC SAFETY

### SEARCH AND RESCUE

1. Mountain Remote Site Rescue
2. Ocean/River Rescue
3. Missing or Late Vessels
4. Ship Collisions and Groundings
5. Missing Persons
6. Aircraft Accident
7. Endangered Fire Fighting Equipment

### LAW ENFORCEMENT

1. Drug Enforcement & Detection
2. Security (Building & VIPs)
3. Surveillance (General & Covert)
4. Search (Fugitives & Vehicles)
5. Patrol
6. Observation Post
7. High Speed Pursuit
8. Command Post
9. Crowd Control (Traffic & Riots)
10. Pollution Control
11. Transport (VIPs & Crime Specialists)
12. Stolen Property Recovery
13. Ambulance Escort
14. Disaster Warning & Relief
15. Emergency Cargo Transport
16. Fire Detection
17. Rescue
18. Search (People Lost)
19. Traffic (Emergency)
20. Water Area Patrol
21. Aerial Photography

### EMERGENCY MEDICAL SERVICES

1. At the Scene Accident Pick-Ups
   - A. Traffic
   - B. Occupational
   - C. Residential
   - D. Recreational
2. Interhospital Transfers
   - A. Critical Patient Transfer
   - B. Neonatal Transfer
   - C. Burn Patient Transfer
   - D. Organ/Blood Transport
   - E. Medical Supply Transport
   - F. Medical Equipment Transport

### FIRE FIGHTING

1. Transport Personnel
   - A. Fire Crews
   - B. Command Post
   - C. Firefighting Tools, Hardware & Supplies
   - D. Suspended Maneuvering System
2. Retardant Applications
3. Reconnaissance
   - A. Mapping
   - B. IR Sensing
   - C. Dry Season Surveillance
   - 4. Backfiring

### DISASTER RELIEF

1. Lifesaving People Transport
2. Life Sustaining Supply Transport
3. Evacuations
4. Early Warning & Response
5. Command Post
6. Post-Disaster Clean-Up

## QUALITY OF LIFE

### WILDLIFE MANAGEMENT

1. Herding Animals
2. Tagging Animals
3. Relocating Animals
4. Damage Control
5. Fish Stocking
6. Fish Management

### SURVEYS

1. Animal & Fish Population
2. Inspect Oil Platforms
3. Inspect Strip Mines
4. Inspect Powerlines
5. Inspect Dams & Reservoirs
6. Aerial Photography
7. Factory Pollution Monitoring
8. Wetlands Inspection.

### EXTERNAL LOADS

1. Tower & Pole Setting
2. Wire Stringing
3. Pipeline Laying
4. Liming Lakes
5. Seeding Forests
6. Remote Site Construction
7. Remote Site Supply
8. Snooding

### LAND MANAGEMENT

1. Fire Control
   - A. Bureau of Land Management
   - B. U.S. Forest Service
   - C. Bureau of Indian Affairs
2. Geological Studies
   - A. Exploration
   - B. Earthquake Research
   - C. Volcano Research
   - D. Channel Monitoring
3. Cadastral Surveys
4. Electronic Surveys
5. Resource Management

### TRANSPORTATION

1. Inspection
2. Work Crews
3. Survey Equipment
4. Survey Personnel
5. Resupply
6. Search & Rescue

**Ref: Morrison, 1982**

Public service helicopter missions.

FIGURE 7. POTENTIAL TILTROTOR CARIBBEAN PASSENGER ROUTES

| From | To | Passengers /Year | Distance (Miles) |
|---|---|---|---|
| St. John | St. Martin | 79,475 | 108 |
| St. Martin | St. John | 79,475 | 108 |
| Port of Spain | Bridgetown, Barbados | 77,143 | 208 |
| Bridgetown, Barbados | Port of Spain | 77,143 | 208 |
| Santo Domingo, Dom. Rep. | San Juan, Puerto Rico | 64,536 | 234 |
| San Juan, Puerto Rico | Santo Domingo | 64,536 | 234 |
| St. Lucia | Bridgetown, Barbados | 44,676 | 120 |
| Bridgetown, Barbados | St. Lucia | 42,132 | 120 |
| Bridgetown, Barbados | St. Vincent | 36,599 | 120 |
| Panama | San Jose, Costa Rica | 34,658 | 350 |
| St. Vincent | Bridgetown, Barbados | 31,938 | 120 |
| San Jose, Costa Rica | Panama | 31,126 | 350 |
| Port Au Prince | Santo Domingo | 31,730 | 170 |
| San Salvador, El Salvador | San Jose, Costa Rica | 30,939 | 440 |
| San Jose, Costa Rica | San Salvador, El Salvador | 29,546 | 440 |
| Santo Domingo, Dom. Rep. | Port Au Prince, Haiti | 25,931 | 170 |

Source: Summary Final Report (Civil Tiltrotor Missions and Applications: A Research Study), Boeing Commercial Airplane Co.; Bell Textron, Boeing Vertol, NASA ARC; July 1987 (NASA CR 177452)

FIGURE 8. AIR ROUTES BETWEEN ISLANDS OF THE CARIBBEAN BASIN

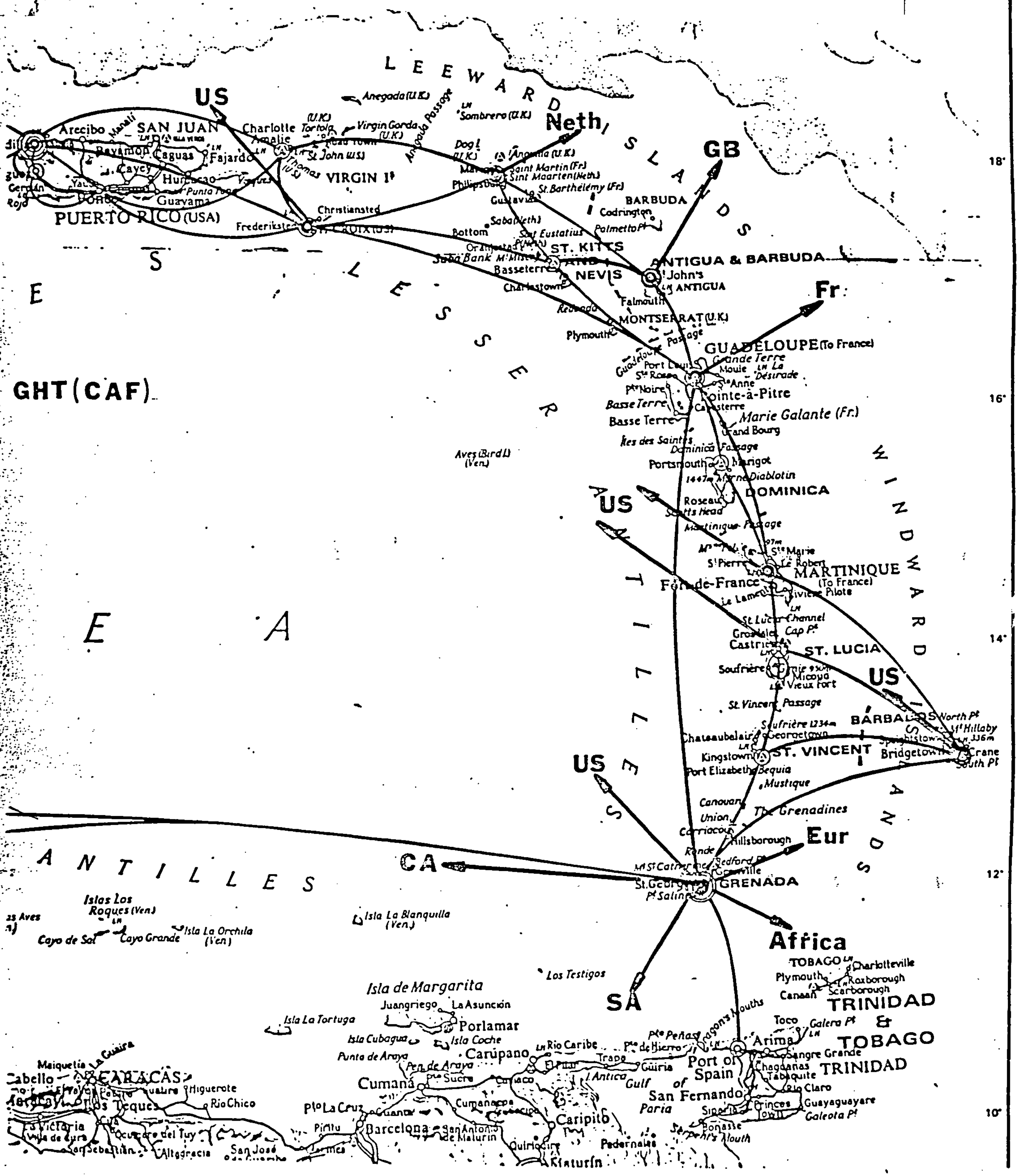


FIGURE 9. 1985 US – Caribbean Trade ($ in Millions)

| Item | Value | Value Transported By Air* | Air Transport Cost (17%) |
|---|---|---|---|
| Agricultural Foods | | | |
| Coffee | $641 | $641 | $109 |
| Bananas | 423 | 423 | 72 |
| Sugars | 263 | 263 | 45 |
| Shellfish | 207 | 207 | 35 |
| Beef/Veal | 106 | 106 | 18 |
| Electronic Components | | | |
| Integrated circuits | $170 | $170 | $29 |
| Switches | 66 | 66 | 11 |
| Capacitors | 28 | 28 | 5 |
| Sewn Products | | | |
| Lace Garments | $82 | $62 | $11 |
| Women's Apparel | 76 | 57 | 10 |
| Men's Apparel | 67 | 50 | 9 |
| Body Supporting Garments | 38 | 29 | 5 |
| Total | $2,167 | $2,102 | $359 |

*100% for agricultural foods and electronic components; 75% for sewn products.

FIGURE 10.

INTERNATIONAL TRADE BY AIR BETWEEN THE REGION AND THE UNITED STATES

| COUNTRY | Air as % of Total by Value IMPORTS (%) | EXPORTS (%) |
|---|---|---|
| CARIBBEAN | | |
| Bahamas | 14 | 4 (50) |
| Barbados | 25 | 50 |
| Bermuda | 26 | 40 |
| Cayman Islands | 12 | 4 (4) |
| Cuba | 80 | -- |
| Dominican Republic | 15 | 21 |
| French Antilles | 24 | 63 (64) |
| Haiti | 31 | 49 |
| Jamaica | 9 | 4 |
| Leeward and Winward Islands | 25 | 41 (52) |
| Netherlands Antilles | 21 | 0 (14) |
| Trinidad & Tobago | 14 | 0 (6) |
| Turks & Caicos Islands | 16 | 98 |
| SUB-TOTAL | 31 | 6 (25) |

FIGURE 11. SUMMARY DATA FOR HELICOPTER BASED EMERGENCY MEDICAL CENTERS IN THE UNITED STATES. (DATA PROVIDED BY ROCKY MOUNTAIN HELICOPTERS, INC.)

| | | |
|---|---|---|
| Population with 50 miles of sponsoring hospital | | 1,177,850 |
| Number of beds of sponsoring hospital | | 675 |
| Annual transports: | | |
| | first year | 370 |
| | second year | 462 |
| Annual transports per 100,000 | | 31 |
| Percentage of transports to sponsor hospital | | 60% |
| Length of stay for helicopter transported patient | | 16 days |
| Percentage of transports within 50 mile radius | | 75% |
| Type of helicopter response: | | |
| | Hospital transfers | 75% |
| | Scene pickups | 25% |
| Patient diagnosis: | | |
| | Trauma/Surgical | 45% |
| | Cardiac | 15% |
| | Other medical | 25% |
| | High risk mother/infant | 10% |
| | Burns | 5% |
| Helicopter response to requests: | | |
| | Completed | 90% |
| Not completed: | | |
| | Bad weather | 5% |
| | Other | 5% |

TABLE 12.- COMPARATIVE ANALYSIS OF THE FIRST YEAR OF OPERATION OF EIGHT HOSPITAL-BASED REMS SYSTEMS

| Hospital/location | Flights per 100,000 population |
|---|---|
| John Lincoln/Phoenix, AZ | 41 |
| Emanual/Portland, Oregon or Maine? | 8 |
| University/San Diego, CA | 24 |
| Hermann/Houston, TX | 12 |
| St. Vincent/Toledo, OH | 25 |
| Baptist/Pensacola, FL | 71 |
| Latter Day Saints/Salt Lake City, UT | 63 |
| St. Anthony's/Denver, CO | 48 |
| Mean | 36.5 |

TABLE 13.- SUMMARY CALCULATIONS FOR PREDICTING REMS REQUEST FOR CENTER LOCATED IN PUERTO RICO

| | Total cases | Pop. (%) | Need (%) | HEMS needed | Demand (%) | Predicted REMS calls |
|---|---|---|---|---|---|---|
| MVA trauma, death likely | 516 | | | | | |
| Basic service area | | 70 | 50 | 180.6 | | |
| Major service area | | 30 | 5 | 7.74 | | |
| Sub Total | | | | 188.3 | 25 | 47.09 |
| MVA accident survivors | 38000 | | | | | |
| Basic service area | | 70 | 5 | 1330 | | |
| Major service area | | 30 | 2.5 | 285 | | |
| Sub Total | | | | 1615 | 7.5 | 121.13 |
| Falls and spinal cord inj. | 280 | | | | | |
| Basic service area | | 70 | 75 | 147 | | |
| Major service area | | 30 | 10 | 8.4 | | |
| Sub Total | | | | 155.4 | 33 | 91.28 |
| Burn victims | 1088 | | | | | |
| Basic service area | | 70 | 10 | 76.16 | | |
| Major service area | | 30 | 5 | 16.32 | | |
| Sub Total | | | | 92.48 | 40 | 36.99 |
| Myocardial infarction | 5984 | | | | | |
| Basic service area | | 70 | 10 | 418.9 | | |
| Major service area | | 30 | 5 | 89.76 | | |
| Sub Total | | | | 508.6 | 7.5 | 38.15 |
| Cerebrovascular accid. | 5542 | | | | | |
| Basic service area | | 70 | 5 | 194 | | |
| Major service area | | 30 | 2.5 | 41.57 | | |
| Sub Total | | | | 235.5 | 7.5 | 17.67 |
| Pediatric and other emer. | 10200 | | | | | |
| Basic service area | | 70 | 10 | 714 | | |
| Major service area | | 30 | 5 | 153 | | |
| Sub Total | | | | 667 | 15 | 130.05 |
| TOTAL REQUESTS | | | | | | 442.35 |

TABLE 14.- SUMMARY CALCULATIONS FOR PREDICTING HEMS REQUEST FOR CENTER LOCATED IN GRENADA

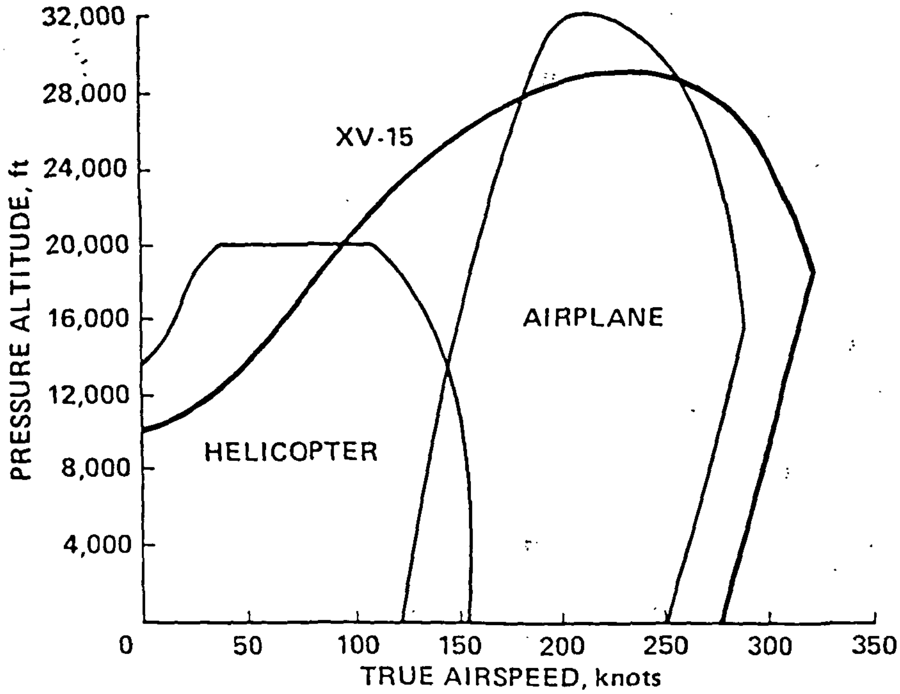
| | Total cases | Pop. (%) | Need (%) | HEMS needed | Demand (%) | Predicted REMS calls |
|---|---|---|---|---|---|---|
| MVA trauma, death likely | 127 | | | | | |
| Basic service area | | 98 | 50 | 62.23 | | |
| Major service area | | 2 | 5 | 0.127 | | |
| Sub Total | | | | 62.36 | 25 | 15.59 |
| MVA accident survivors | 10088 | | | | | |
| Basic service area | | 98 | 5 | 494.3 | | |
| Major service area | | 2 | 2.5 | 5.044 | | |
| Sub Total | | | | 499.4 | 7.5 | 37.45 |
| Falls and spinal cord inj. | 249 | | | | | |
| Basic service area | | 98 | 75 | 183 | | |
| Major service area | | 2 | 10 | 0.498 | | |
| Sub Total | | | | 183.5 | 33 | 60.56 |
| Burn victims | 960 | | | | | |
| Basic service area | | 98 | 10 | 94.08 | | |
| Major service area | | 2 | 5 | 0.96 | | |
| Sub Total | | | | 95.04 | 40 | 38.02 |
| Myocardial infarction | 5280 | | | | | |
| Basic service area | | 98 | 10 | 517.4 | | |
| Major service area | | 2 | 5 | 5.28 | | |
| Sub Total | | | | 522.7 | 7.5 | 39.20 |
| Cerebrovascular accid. | 4890 | | | | | |
| Basic service area | | 98 | 5 | 239.6 | | |
| Major service area | | 2 | 2.5 | 2.445 | | |
| Sub Total | | | | 242.1 | 7.5 | 18.15 |
| Pediatric and other emer. | 9000 | | | | | |
| Basic service area | | 98 | 10 | 882 | | |
| Major service area | | 2 | 5 | 9 | | |
| Sub Total | | | | 891 | 15 | 133.65 |
| TOTAL REQUESTS | | | | | | 342.62 |

TABLE 15.- POPULATION WITHIN RANGE OF SELECTED ISLANDS OF THE CARIBBEAN BASIN

| | Population in 1982 Est. | Population within 150 miles | Population within 200 miles | Population within 300 miles |
|---|---|---|---|---|
| Antigua and Barbuda | 77000 | 825000 | 1076000 | 4846000 |
| Barbados | 260000 | 924000 | 2164000 | 2614000 |
| Br. Virgin Islands | 11000 | 3438000 | 3527000 | 9582000 |
| Dominica | 75000 | 1077000 | 1449000 | 2741000 |
| Dominica Republic | 5660000 | 10805000 | 10805000 | 14202000 |
| Grenada | 112000 | 2089000 | 2164000 | 2573000 |
| Guadeloupe | 320000 | 949000 | 1193000 | 4846000 |
| Haiti | 5145000 | 10805000 | 10805000 | 13040000 |
| Jamaica | 2235000 | 2235000 | 2235000 | 7380000 |
| Martinique | 300000 | 1408000 | 1449000 | 2730000 |
| Montserrat | 12000 | 825000 | 1076000 | 1576000 |
| Puerto Rico | 3270000 | 3397000 | 3438000 | 95070000 |
| St. Kitts, Nevis, Angila | 41000 | 652000 | 4222000 | 4734000 |
| St. Lucia | 124000 | 1319000 | 2573000 | 2614000 |
| St. Vincent, Grenadines | 128000 | 2164000 | 2484000 | 2614000 |
| Trinidad, Tobago | 1165000 | 1405000 | 1789000 | 2164000 |
| U. S. Virgin Islands | 116000 | 3438000 | 3847000 | 9880000 |

FIGURE 16. SUMMARY ESTIMATES FOR PREDICTED HEMS REQUESTS FOR COUNTRIES OF THE CARIBBEAN BASIN

| Country: | Estimate of HEMS Calls/year |
|---|---|
| Bahamas | 20.6 |
| Colombia | 2858 |
| Costa Rica | 249 |
| Cuba | 825 |
| Dominican Republic | 577.5 |
| El Salvador | 488.6 |
| Guatemala | 826.9 |
| Grenada | 342.6 |
| Haiti | 613.4 |
| Honduras | 443.7 |
| Jamaica | 216.95 |
| Mexico | 7776.3 |
| Nicaragua | 275.5 |
| Panama | 191.2 |
| Puerto Rico | 442.4 |
| Venezuela | 1797.9 |
| Total | 17945.55 |




Speed envelope comparison of XV-15 Tilt Rotor.

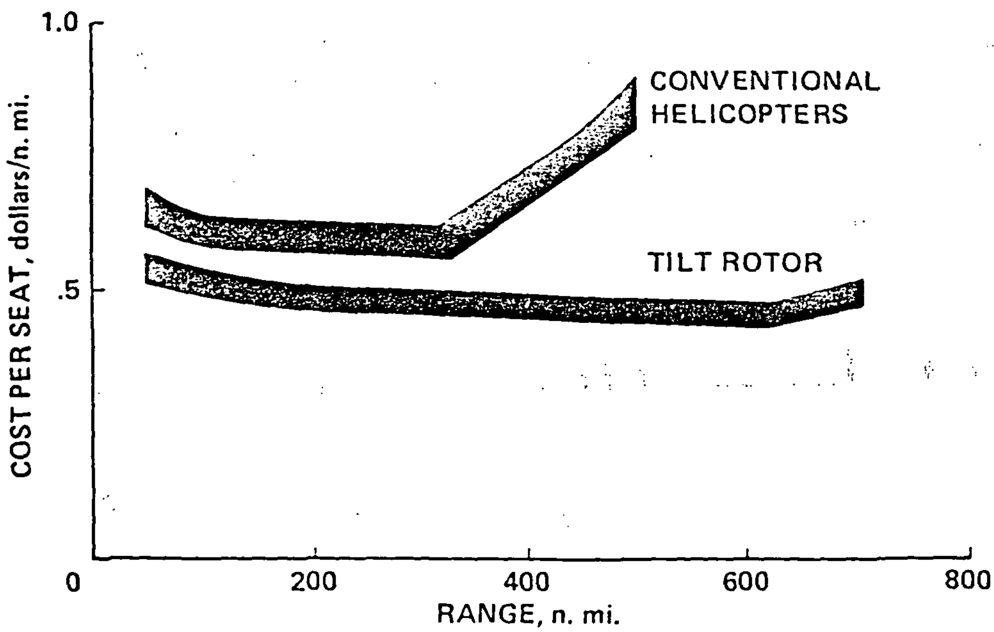
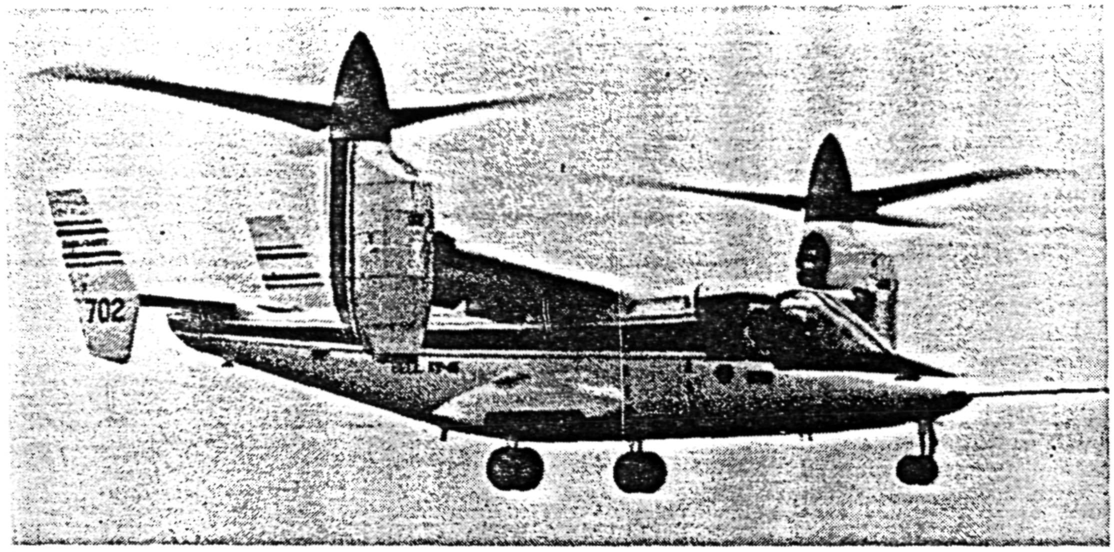
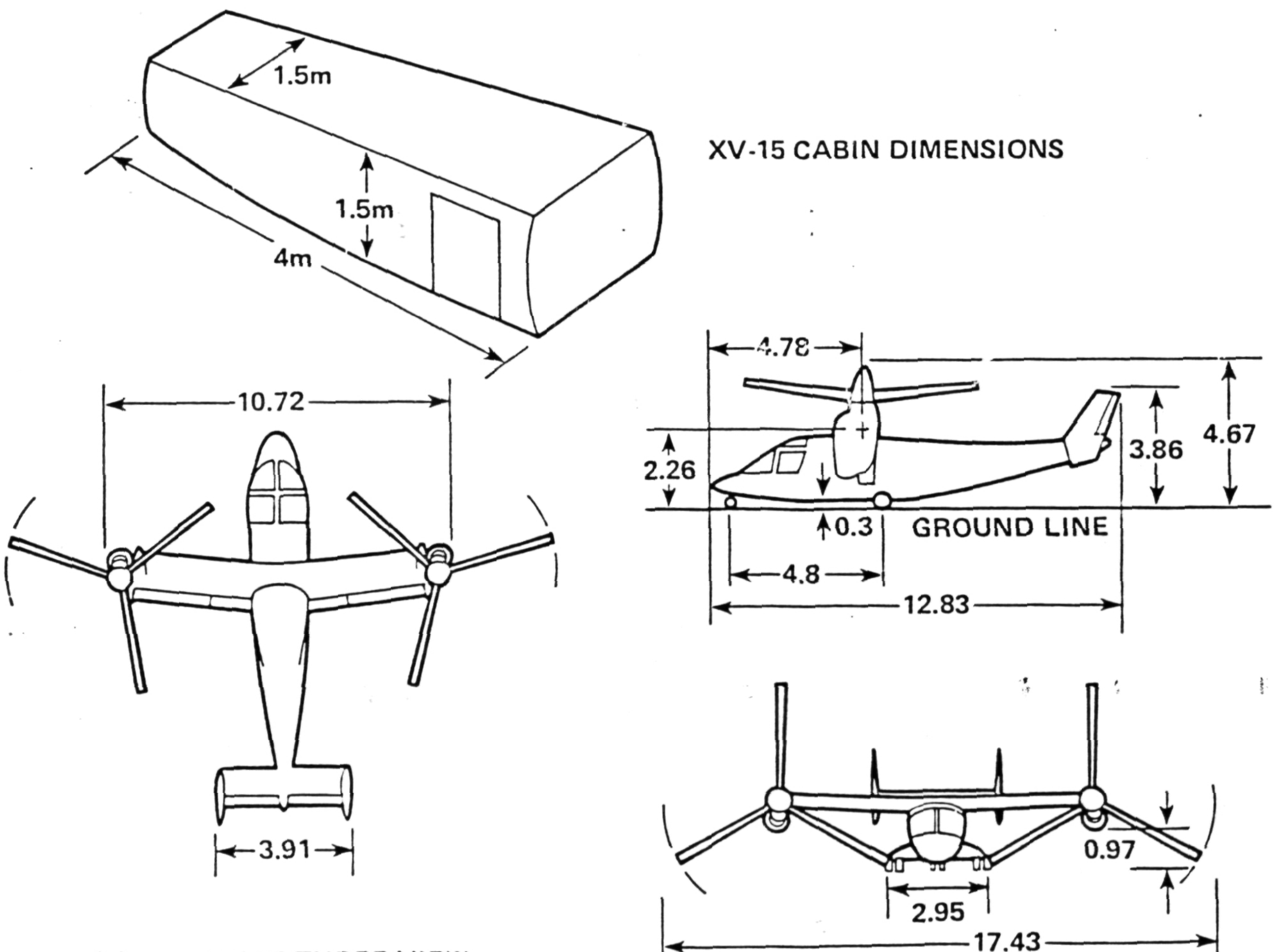



Figure 17.- Seat/mile costs of XV-15 Tilt Rotor.




(a)



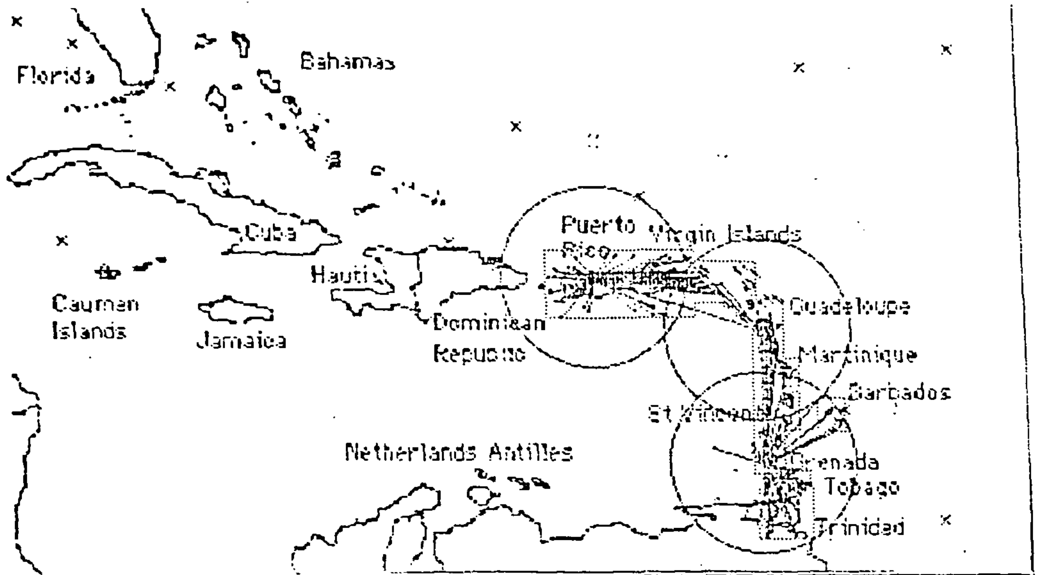

(b) XV-15 THREE-VIEW

(a) XV-15 cabin dimensions.

(b) XV-15 three-view.

Figure 18.- Dimensions of XV-15 Tilt Rotor in meters.

FIGURE 19. RESULTS OF SIMULATION: THREE STANDARD HELICOPTERS




8 Week Simulation Time

| | | | |
|---|---|---|---|
| Number of Rotorcraft Flights | 166 | Missed (out of Range) | 12 |
| Avg Distance to Scene | 54 | Missed (Unavailable) | 15 |
| Avg Wait for Rotorcraft arrival | 32 | Re-Simulate | |
| Avg Total Transfer Time | 125 | [illegible] PRINT OK | |

San Juan Status = 3 Flights Received = 50

| craft | Rnge | N-Flts | Av Rsc (min.) | Av wait (min.) | Flt Hrs /week | Av Dst |
|---|---|---|---|---|---|---|
| 1 | 150 | 51 | 118 | 39 | 6 | 73 |
| | | 51 | 118 | 39 | | |

Guadelot Status = 2 Flights Received = 1

| craft | Rnge | N-Flts | Av Rsc (min.) | Av wait (min.) | Flt Hrs /week | Av Dst |
|---|---|---|---|---|---|---|
| 1 | 150 | 71 | 130 | 20 | 25 | 25 |
| | | 71 | 130 | 20 | | |

Grenada Status = 3 Flights Received = 105

| craft | Rnge | N-Flts | Av Rsc (min.) | Av wait (min.) | Flt Hrs /week | Av Dst |
|---|---|---|---|---|---|---|
| 1 | 150 | 44 | 125 | 42 | 6 | 79 |
| | | 44 | 125 | 42 | | |

| ntr | Stat | Rcvd | Bird num. | Rnge | total flts | Av. resc. (min.) | Av. wait (min.) | Flt Hrs. /week | Av. Dist |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 3 | 60 | 1 | 150 | 51 | 117.8 | 39.3 | 6.2 | 73.2 |
| 2 | 2 | 1 | 1 | 150 | 71 | 130.5 | 20.0 | 26.3 | 24.9 |
| 3 | 3 | 105 | 1 | 150 | 44 | 124.9 | 41.6 | 5.8 | 79.0 |
| | | 166 | | | | 125.1 | 31.6 | | |

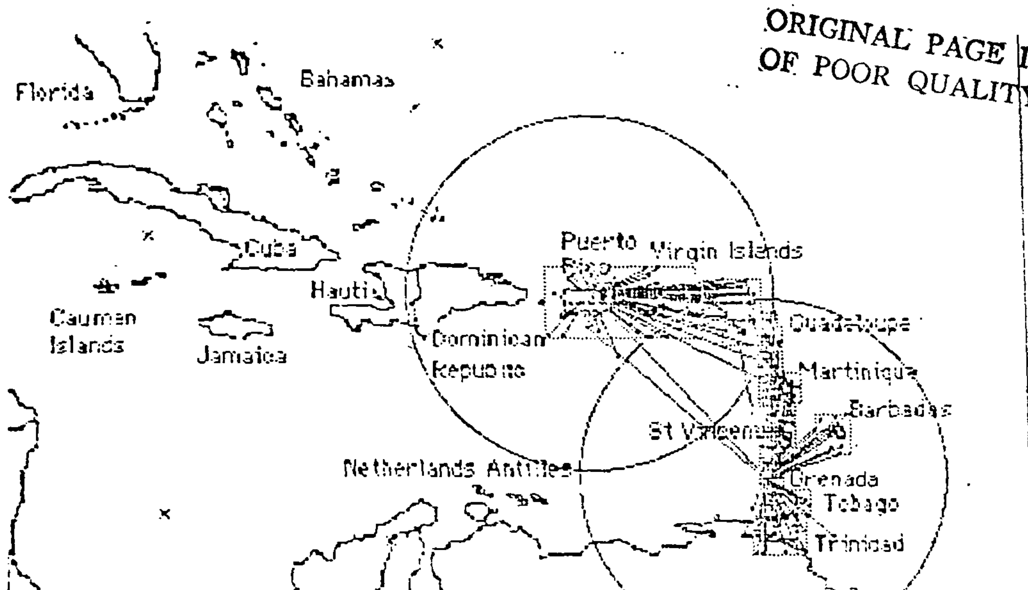
here were 12 "out of range" calls and 15 non-responses due to "in-service" or "in-repair"



ORIGINAL PAGE IS OF POOR QUALITY

FIGURE 20. RESULTS OF SIMULATION: TWO TILTROTORS






8 Week Simulation Time

| | | | |
|---|---|---|---|
| Number of Roturcraft Flights | 165 | Missed (out of Range) | 2 |
| Avg Distance to Scene | 120 | Missed (Unavailable) | 2 |
| Avg Wait for Rotorcraft arrival | 34 | Re-Simulate | |
| Avg Total Transfer Time | 100 | PRINT | OK |

San Juan Status = 3 Flights Received = 92

| craft | Rnge | N-Flts | Av Rsc (min.) | Av wait (min.) | Flt Hrs /week | Av Dst |
|---|---|---|---|---|---|---|
| 1 | 300 | 96 | 101 | 34 | 10 | 122 |
| | | 96 | 101 | 34 | | |

Guadelou Status = 2 Flights Received = 6

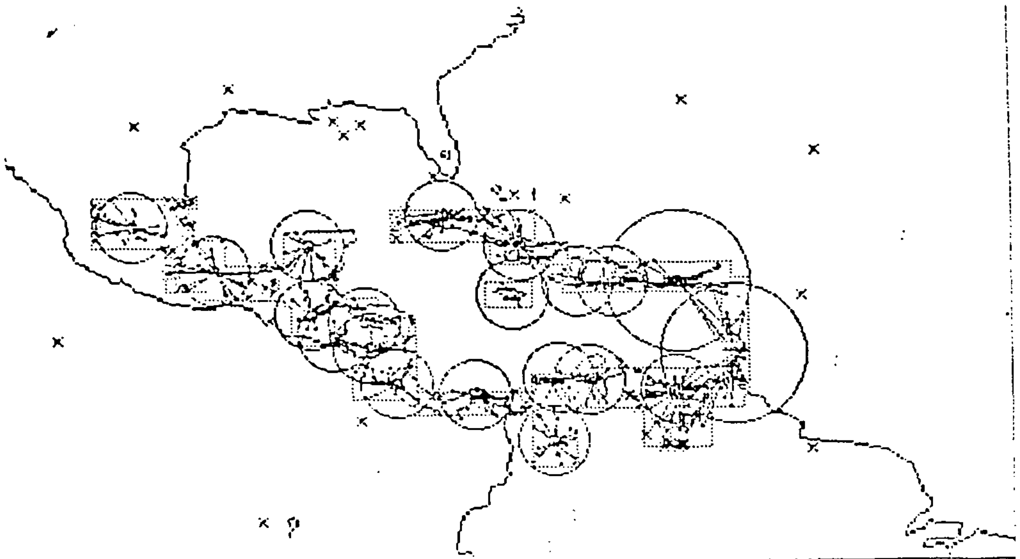
Grenada Status = 3 Flights Received = 67

| craft | Rnge | N-Flts | Av Rsc (min.) | Av wait (min.) | Flt Hrs /Week | Av Dst |
|---|---|---|---|---|---|---|
| 1 | 300 | 69 | 98 | 33 | 7 | 117 |
| | | 69 | 98 | 33 | | |

| Cntr | Stat | Rcvd | Bird num. | Rnge | total flts | Av. resc. (min.) | Av. wait (min.) | Flt Hrs. /week | Av. Dist |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 3 | 92 | 1 | 300 | 96 | 100.9 | 34.4 | 10.1 | 121.9 |
| 2 | 2 | 6 | | | | | | | |
| 3 | 3 | 67 | 1 | 300 | 69 | 98.1 | 33.3 | 6.9 | 116.5 |
| | | 165 | | | | 99.8 | 33.9 | | |

There were 2 " out of range " calls and 2 non-responses due to "in-service" or "in-repair"

FIGURE 21. RESULTS OF SIMULATION: ENTIRE REGION



4 Week Simulation Time

| | | | |
|---|---|---|---|
| Number of Rotorcraft Flights | 289 | Missed (out of Range) | 84 |
| Avg Distance to Scene | 104 | Missed (Unavailable) | 8 |
| Avg Wait for Rotorcraft arrival | 45 | | Re-Simulate |
| Avg Total Transfer Time | 134 | PRINT | OK |

ST Status = 3 Flights Received = 16

| craft | Rnge | N-Flts | Av Rsc (min.) | Av wait (min.) | Flt Hrs /week | Av Dst |
|---|---|---|---|---|---|---|
| 1 | 300 | 17 | 120 | 43 | 5 | 164 |
| | | 17 | 120 | 43 | | |

DR Status = 3 Flights Received = 19

| craft | Rnge | N-Flts | Av Rsc (min.) | Av wait (min.) | Flt Hrs /week | Av Dst |
|---|---|---|---|---|---|---|
| 1 | 150 | 18 | 98 | 33 | 3 | 57 |
| | | 18 | 98 | 33 | | |

Haiti Status = 3 Flights Received = 6

| craft | Rnge | N-Flts | Av Rsc (min.) | Av wait (min.) | Flt Hrs /week | Av Dst |
|---|---|---|---|---|---|---|
| 1 | 150 | 7 | 106 | 35 | 2 | 63 |
| | | 7 | 106 | 35 | | |

ORIGINAL PAGE IS
OF POOR QUALITY

# BIRDS

A program to simulate rotorcraft rescue operations

Robert W. Smith
University of Puerto Rico
Mayaguez, Puerto Rico

# BIRDS
## A program to simulate rotorcraft rescue operations

Robert W. Smith
University of Puerto Rico

### Problem/Function Definition

In order to evaluate rotorcraft rescue operations and compare the effectiveness of different aircraft and center locations there are several key questions which must be considered. For a given configuration of rotorcraft and center locations what percentage of the incidents are within range? What is the average rescue time? How long must the victims wait before arrival of the rotorcraft? What is the average distance of the rescue flight? How many calls are missed due to downtime caused by repair or weather conditions? How important is speed in making a significant reduction in response time? What is the optimal location of the rotorcraft bases to reach the most calls in the least time? This simulation attempts to answer some of these questions in order to help to make a comparison between different aircraft and center locations.

### Method of Solution

This program is a finite event Monte Carlo simulation of aircraft operations. After accepting information relating to aircraft capacities, center locations and regions in which a user supplied percentage of accidents occurs the program generates accidents and simulates rescue operations in order to collect pertinent data for comparison of different configurations of center locations and aircraft capacity.

While the initial application was for rotorcraft rescue operations the simulation can be used for any operation in which incidents occur in specified regions which must be responded to by aircraft located at fixed centers. The program can generate a map of any region of the world or maps previously created by other programs may be imported for use by the simulation. The program generates accidents in a given region according to criteria input by the user and then calculates the time required for an aircraft to reach the location at which the accident occurred. The availability of an aircraft depends on its location, range, speed, and whether or not it is currently being used for a previous call or is out of service due to repairs or bad weather. The program accepts as input the locations and categories of centers, the number and capacity of aircraft at each center, the region in which a given percentage of accidents will occur, and other data relating to rescue time, response time, etc.

The program was developed on a Macintosh micro-computer using the Turbo Pascal language and makes extensive use of graphics for both input of data and illustration of the simulation. The output of the program includes the average wait time before an accident is reached by an aircraft, the average rescue time before the victim is taken to the nearest appropriate hospital, the number of accident victims which were not rescued due to the lack of an available helicopter, and the number of hours per week that each aircraft spends flying. Figure 1 illustrates the results of a simulation.

## Implementation Instructions

The program was compiled using Turbo Pascal. The only files necessary in order to run the program are the compiled program called 'BIRDS' and a map file 'World.dat' which contains a data representation of a map of the world. The program can be run on any Macintosh computer with as little as 512K memory for the small map data file version. To run the program simply click on the file 'BIRDS'.

## User Instructions

The program follows the guidelines for a standard Macintosh application with few exceptions and is therefore quite self explanatory. After the initial 'about' display the user is presented with the option of selecting between several pull down menus. A description of each item follows:

FILE:

NEW: This selection will import the picture currently in the scrap area to be used as a map for a simulation. You must have used the copy command to place a map
on the 'scrap' before using this command.

OPEN: This selection will produce a standard file dialog and ask the user to select a previously saved file to re-run a simulation.

SAVE: This selection is used to save a file in order to re-run a simulation.

QUIT: This selection will terminate the program and return the user to the desktop.

EDIT:

COPY: This selection will place the current map on the 'scrap' so that it may be altered using a drawing program such as MacPaint.

SIM:

RUN: This selection will begin the simulation once a map has been selected.

MAP:

NewMap: This selection will cause a map of the entire world to be displayed on the screen and direct the user to select a sub-region by using the mouse to enclose it with a rectangle. The sub-region will then be re-displayed in greater detail in order to be used for the simulation.

OpenMap: This selection is used to select a previously saved map file to be used for the current simulation.

FromScrap: This selection will import the picture currently in the scrap area to be used as a map for a simulation. You must have used the copy command to place a map on the 'scrap' before using this command.

SaveMap: This selection is used in order to save the current map file for later use.

After a map has been created or loaded from disk the user should select the RUN command. At this point the map will be re-drawn and the user will be asked to supply information for the simulation. The following information is required:

(1) The user will be asked to use the mouse to indicate a distance of one hundred miles. This distance will be used to calculate the range and speed of aircraft in the simulation.

(2) The user will then be asked to indicate the locations of the hospitals by clicking with the mouse. After all hospitals have been located the user must click the mouse in the 'OK' rectangle.

(3) For each hospital the user will be asked to indicate the number and capacity of all aircraft located at this center. In addition the user is asked to indicate the 'status' of this center. A status of three is used to indicate a full trauma center which can handle all accidents. A level two center is a secondary level center which can not service the most serious emergencies.

(4) After all hospitals have been located and information supplied about the aircraft located at each center the user is asked to indicate the 'high accident regions'. These regions are indicated by using the mouse to enclose a series of rectangles. After these regions have been created the user must click the mouse in the 'OK' box. At this point the program will ask for the percentage of accidents which will occur in the 'high accident region'.

(5) After the user has indicated what percentage of accidents are to occur in the 'high accident region' the program will present a dialog box with nine questions along with default answers about the current simulation: The user can use the default values to change any or all of the values. An item can be selected either by using the mouse to move to another item or by using the 'TAB' key to move on to the next item. Pressing the 'ENTER' key signals the program to accept the current values and continue with the simulation. The information to be supplied is as follows:

(a) Number of accidents/year requiring Rotorcraft.

(b) Percentage of accidents which must be taken to a level 3 Trauma Center.

(c) Percentage of ground transport ambulance accidents which will later need air transfer. This is used to estimate the number of air transfers from centers without aircraft to another center.

(d) Percentage of transfer which must go to a level 3 trauma center.

(e) Hours/week that aircraft are out of service due to maintenance.

(f) Average duration of Maintenance (in minutes).

(g) Average duration of weather downtime ( in minutes).

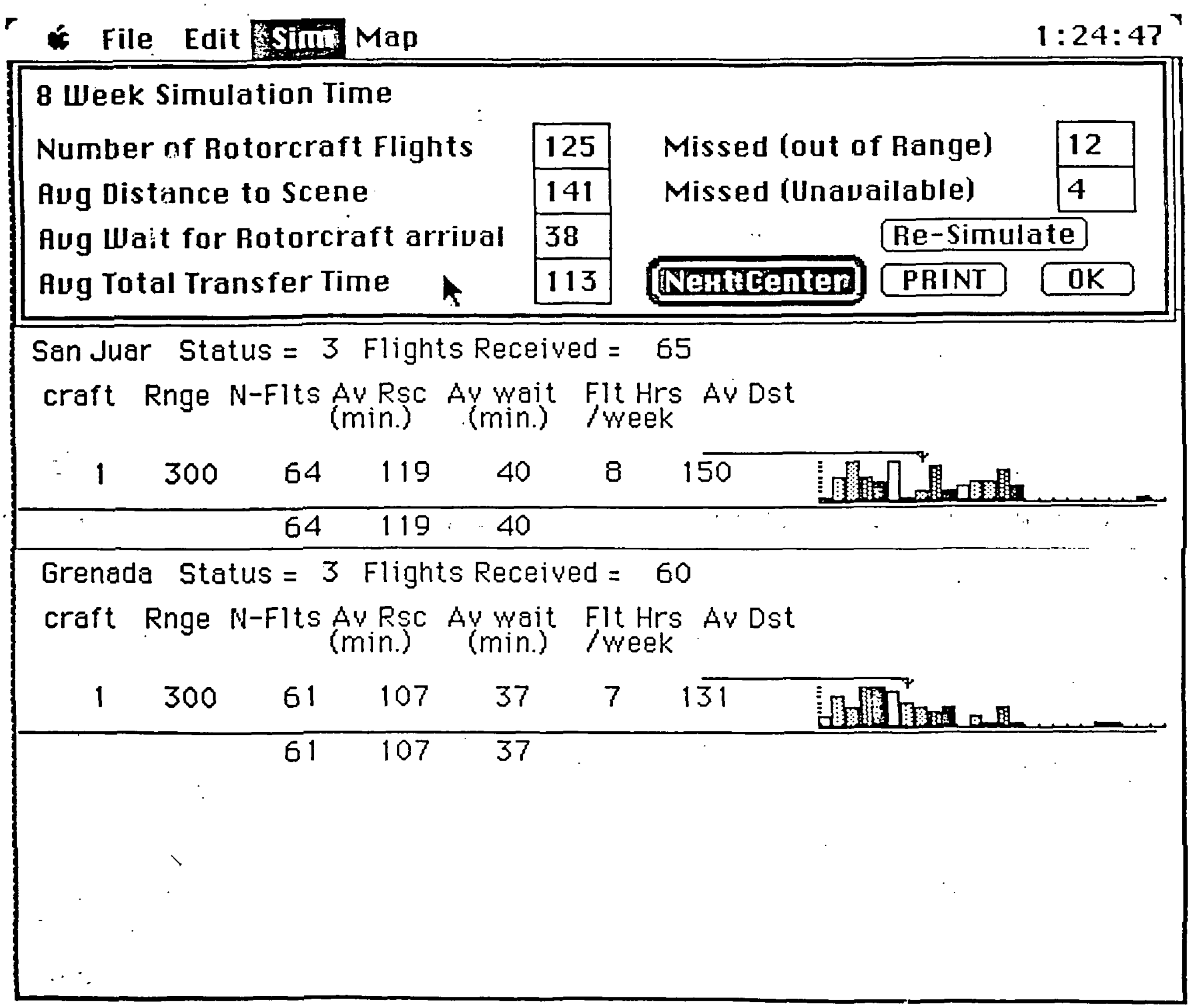
(h) Average time on ground to pick up a patient.

(i) Average time to respond to call.

(6) After the user has pressed 'ENTER' or used the mouse to click in the 'OK' button to indicate his acceptance of the displayed values to the above questions the program will ask if he wants to save the file with the current information for future use.

(7) Finally the program will ask for the simulation time in weeks.

At this point the simulation will begin and the rescue operations will be indicated on the screen as the simulation progresses. After the simulation is finished the user will be asked to press the 'ENTER' key in order to see the data for the completed simulation. before pressing the 'ENTER' key the user might want to press the key combination ( 'Shift'/'Command'/'3' ) in order to save the current display on disk or ( 'Shift'/'Command'/'4' ) in order to send the display to the printer. After the user presses the 'ENTER' key the data indicating the results of the simulation is displayed on the screen. An example of this data follows:



The top section includes global information for all of the centers in the simulation. The bottom section includes information for one center. The user can cause information for the next center to be displayed by pressing the 'ENTER' key or by clicking in the 'NEXT CENTER' button with the mouse.

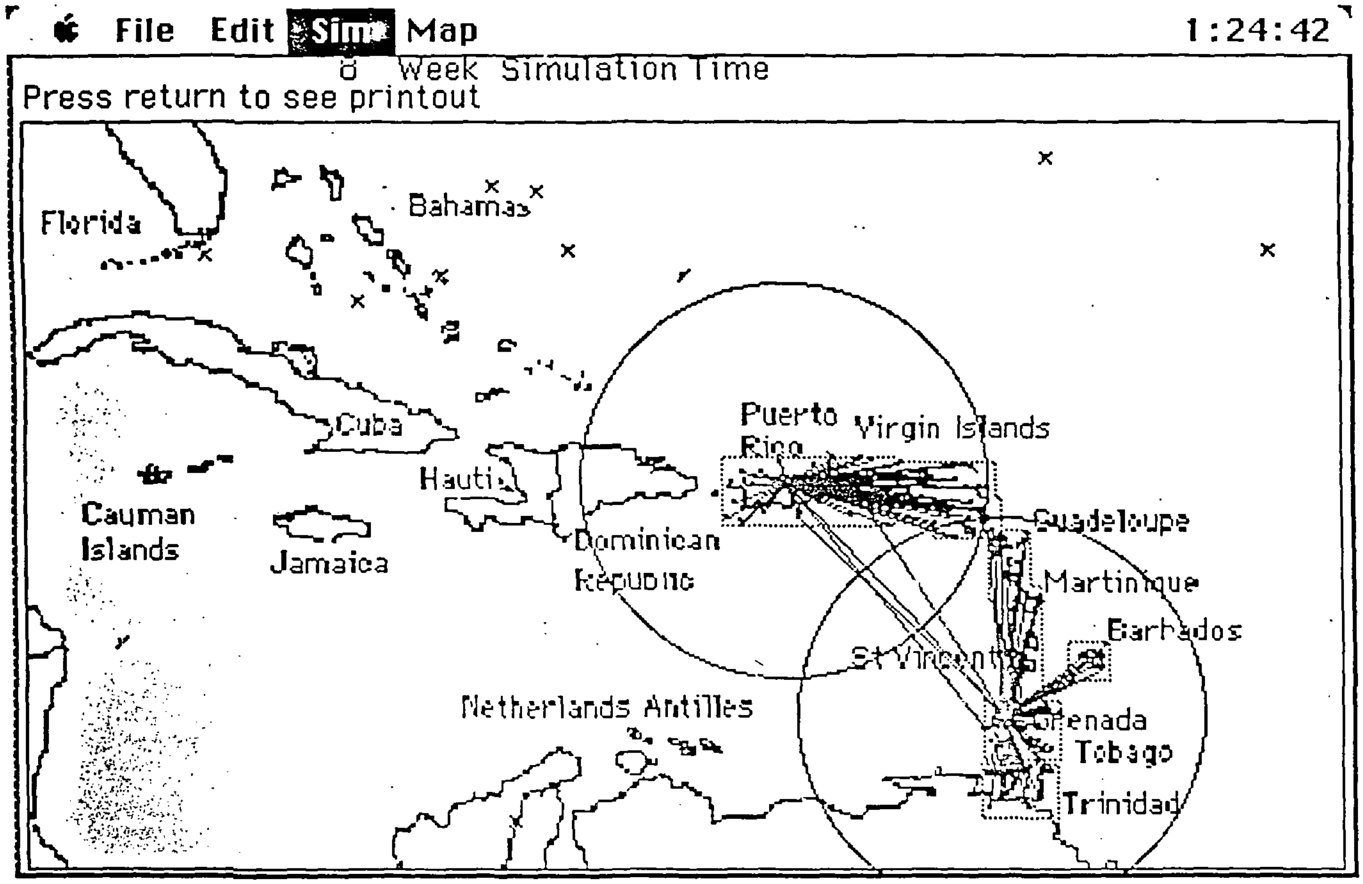
For each center the information displayed includes the center name, the status of the center and the number of flights received by the center.

For each aircraft located at a center the following information is displayed:

(a) The number of the aircraft.
(b) The range of the aircraft.
(c) The number of flights by this aircraft during the simulation.
(d) The average total rescue time for this aircraft. This includes response time, flight time and pickup at time.
(e) The average time spent waiting for this aircraft.
(f) The average number of flight hours per week.
(g) The average distance to the scene of the accident.
(h) A histogram of flight distances.

The user may select the 'PRINT' button with the mouse in order to print out a summary of this information. Selecting the 'Re-Simulate' button causes another simulation to be run using a different time frame if desired. By selecting the 'OK' button the user can return to the original initial level in which the pull down menus are activated.

FIGURE 1. GRAPHICAL DISPLAY OF SIMULATION




ORIGINAL PAGE IS
OF POOR QUALITY

The following pages contain a listing of the entire program.

```
Program Birds;

{$U Birds:Units.F: Sim.Lib }
{$O Birds:Birds}
{$R Birds:birds.Rsrc }

{$B+}
{$S+}
{$R+}



  uses
  {$S       } Memtypes, QuickDraw, OSIntf, ToolIntf, PackIntf,sane,
                MacExtras, BirdGlobals, Dialogunit,MacPrint,
  {$S Sim   } BirdSim, MapMer ;


 const
       WindowID    = 1000;
       WindowID2   = 1001;
       FileID     = 2;
       SimID     = 1000;
       MapID     = 1003;

       Newcmd    = 1;
       CloseCmd = 2;
       OpenCmd  = 3;
       SaveCmd  = 4;
       QuitCmd  = 5;

       copyCmd  = 3;
       pasteCmd = 4;

       RunCmd     = 1;

       NewMap     = 1;
       OpenMap    = 2;
       FromScrap = 3;
       SaveMap    = 4;
 var

    spt,pp1                  : ptr;
    RefN                     : Integer;
    pp                       : picptr;
    ph1                      : PicHandle;
    Hndl                     : Handle;
    Lnth,Offset              : Longint;

    SimMenu                  : MenuHandle;
    MapMenu                  : MenuHandle;
    done,ok              : Boolean;
    quitRequested        : Boolean;
    windowOpen           : Boolean;


 Procedure dokeypress( ch : char);
   { do something with incoming char }
   begin
```

```
   end;  { of dokeypress }

 procedure domouseclick( whichwindow : windowptr );
  {process mouse clicks inside windows }
 Begin;

 end;    { domouseclick }


Procedure SetupWindow;
begin;
  If not windowOpen then
  begin
  wPtr := GetNewWindow( WindowID, @Wrec, Pointer( -1 ) );
  windowOpen := wPtr <> NIL;
  If windowOpen Then
    begin
     SetPort( wPtr );
     SelectWindow(wPtr);
     { ClipRect( wPtr^.portRect ); }
     MapFrame := wPtr^.portRect ;
     InsetRect(MapFrame,3,13);
     offsetRect(MapFrame,0,12);
     SetRect(TxtFrame, 3,10,509,25);
     ClipRect( MapFrame );
    end
  end
end;


Procedure DoNew; { get Picture from the scrap }
begin
   SetupWindow;
     readdat := false;
    if (length(Data.MapName)=0) then KillPicture(ph) else ReleaseResource(Hndl);
    { Kill old Pic, if MapName='' then it came from scrap or NewMap else a resource file }
     Hndl := NewHandle(0);
     lnth := GetScrap( Hndl,'PICT',offset);  { Get Handle from scrap     }
     if (lnth = NOTypeErr) then
                begin
                ClipRect( wPtr^.portRect );
                gotoxy(1,1);cleareol;
                writeln('No Map on scrap, You must "copy" one from the ScrapBook (Press retu
                readln;gotoxy(1,1);cleareol;
                ClipRect(MapFrame);
                end
     else
      begin
       ph   := picHandle( Hndl);           { convert it to Pic Handle }
       DrawPicture( ph, MapFrame);
      { SetWindowPic( wPtr, ph );       ???  }
       EnableItem( fileMenu, CloseCmd );
       Disableitem(filemenu, Newcmd );
       EnableItem( SimMenu, RunCmd);
      end;
end;  { doNew }


 procedure closeprogramwindow;
  { close the global wptr window }
   begin
     if windowopen then
     begin
       closewindow(wptr );
       windowopen := false;
       Enableitem(filemenu, newcmd );
```

```
      Disableitem(filemenu, closecmd );
      Disableitem(Editmenu, copycmd );

    end  {if }
  end;  {closeprogramwindow }

procedure doclose;
 { respond to file menu close command }
begin
  readdat := false;
   if (length(Data.MapName) = 0) then KillPicture(ph)
                                   else ReleaseResource(Hndl);
  Data.MapName := '';
  if frontwindow = wptr
    then closeprogramwindow
    else closedawindow;
    windowopen := false;
end;  { doclose }


Procedure GetMap( ResName : string255);
var
   name       : string255;
begin
   SetupWindow;
   if (length(Data.MapName) = 0) then KillPicture(ph)
                                   else ReleaseResource(Hndl);
   name := 'thepic';
      Refn := OpenResFile(Resname);
      Hndl := NewHandle(0);
      Hndl := GetNamedResource('PICT', name);
      DetachResource( Hndl);
      ph := PicHandle(Hndl);

       { save it to the scrap also , this may not be a good idea }
       { Lnth := zeroscrap;                                      }
       { Lnth := ph^^.picsize;                                   }
       { Lnth := putscrap(lnth,'PICT',Hndl^);                    }

    CloseResFile(Refn);
    DrawPicture( ph ,MapFrame);
    { SetWindowPic( wPtr, ph );   }

end; { of GetMap }


Procedure MapOpen;
var
  okayflag        : boolean;
  ResName         : string255;
  id              : integer;
  reply           : SFReply;
  where           : Point;
  typelist        : SFTypeList;
  fileKind        : OSType;
begin
 readdat := false;
 fileKind := 'MAPA';  { ignored if  -1  in 4th parm }
 typelist[0] := fileKind;
 where.h := 60; where.v := 50;
 SFGetFile(where,'',NIL,1,typeList,NIL,reply);
 Resname := reply.fname;
 GetMap(ResName);
 data.MapName := ResName;
    EnableItem( fileMenu, CloseCmd );
    DisableItem( fileMenu,NewCmd );
```

```
    EnableItem( SimMenu, RunCmd);
    EnableItem( EditMenu, copycmd);

end;


Procedure mapSave;
var
   okayflag             : boolean;
   name, Resname        : String255;
   id, err1,err         : integer;
   where                : point;
   reply                : SFReply;
   vRefNum              : integer;
   fndrInfo             : FInfo;
   volName              : StringPtr;
   OSerror              : OSErr;
   Hndl1                : Handle;

begin
   SetupWindow;
   where.h := 60; where.v := 50;
   SFPutFile(where, 'Save File as :','',NIL,reply);
   Resname := reply.fname;

   name := 'thepic';
   CreateResFile(Resname);
     err1:= ResError;
     if ( err1 <> 0 ) then begin
              gotoxy(1,1);cleareol;
              write(' creat ERROR # ',err1, ' Hit Return to continue ');
              readln;gotoxy(1,1);cleareol;
              end;

   RefN:= OpenResFile(Resname);
   err:= ResError;
   if ( err <> 0 ) then begin
               gotoxy(1,1);cleareol;
               write(' creat ERROR # ',err, ' Hit Return to continue ');
               readln;gotoxy(1,1);cleareol;
               end;

     if ( err1 = -48 ) then  { file already exists, so replace contents }
       begin
       Hndl1 := NewHandle(0);
       Hndl1 := GetNamedResource('PICT', name);
       RmveResource(Hndl1);
       DisposHandle(Hndl1);
       end;

   id :=  UniqueID('PICT');
   Hndl := NewHandle(ph^^.picsize);
   Hndl := Handle(ph);

   DrawPicture(Pichandle(Hndl),MapFrame);      { lets see it again }
   HNoPurge(Hndl);
    AddResource( Hndl, 'PICT', id , name);
    err := ResError;
    if ( err <> 0 ) then begin
               gotoxy(1,1);cleareol; write(' ADD ERROR # ',err); readln;
               end;
   HPurge(Hndl);
   CloseResFile(RefN);

   OSerror := GetVol( volName, vRefNum);
   OSerror := GetFInfo(Resname, vRefNum,fndrInfo);
```

```
   fndrInfo.fdType := 'MAPA';                { make it a MAPA file Type }
   OSerror := SetFInfo(Resname, vRefNum,fndrInfo);
   data.MapName := ResName;{ Map Name stored in 'data' so can call it back }

end;



Function getdatfile( VAR done : Boolean ) : Boolean;
var
   okayflag            : boolean;
   folder, Filename    : String[64];
   reply               : SFReply;
begin
 okayflag := false;
 inName := '';
 if GetFileName(reply,'DATA') then inName := reply.fName;
 done := length( inName ) = 0;
 If not done Then
 begin
   {$i-} Reset( datafile, inName );
         read( datafile, Data );
         close(datafile);
   {$i+}
   If IoResult <> 0
   then begin
        gotoxy(1,1); cleareol;Write(' ERROR: cannot find/read ', inName )
        end
   else
     okayflag := ( IoResult = 0 );
  end;
 getdatfile := okayflag;
end;




procedure dofilemenucommands( cmdnumber :integer );
 { excute command in the file menu }
begin
  case cmdnumber of
    newcmd    : donew;
    OpenCmd   : begin
                  readdat := GetDatFile(ok); { readdat true tells sim we have a file }
                  GetMap(data.MapName);
                  EnableItem( SimMenu, RunCmd);
                  EnableItem( EditMenu, copycmd);

                end;
    SaveCmd   : begin    end;  { nothing hear now }
    closecmd  : doclose;
    quitcmd   : quitrequested := true
  end {of case }
end;  { dofilemenucommands }

procedure doeditmenucommands( cmdnumber : integer );
 { execute command in the edit menu }
begin
  if not systemedit( cmdnumber -1 ) then
  begin
   case cmdnumber of

     copyCmd  : begin  { save it to the scrap }
                 Hndl := NewHandle(Sizeof(ph));
                 Hndl := Handle(ph);
                 Lnth := zeroscrap;
```

```
                Lnth := ph^^.picsize;
                Lnth := putscrap(lnth,'PICT',Hndl^);
                end;
     pasteCmd : begin  { get it from the scrap }
                  doNew;
                  Data.MapName := '';{ empty Mapname => can't save datfile}
                end;
    end; { of case }


  end {if }
end;  {doeditmenucommands }


procedure doMapMenucommands( cmdnumber :integer );
 { excute command in the Map menu }
 begin
  EraseRect(MapFrame);
  Framerect(MapFrame);
  case cmdnumber of
    NewMap    : begin
                 SetupWindow;
                 readdat := false;
                 Data.MapName := '';   { empty Mapname => can't save datfile}
                 GetMap('WORLD') ;
                 MakeMap;  UnloadSeg(@MakeMap);
                 { SetWindowPic( wPtr, ph );     }

         { save it to the scrap also , this may not be a good idea }
         {       Hndl  := NewHandle(Sizeof(ph));          }
         {       Hndl  := Handle(ph);                     }
         {       Lnth  := zeroscrap;                      }
         {       Lnth  := ph^^.picsize;                   }
         {       Lnth  := putscrap(lnth,'PICT',Hndl^);  }

                 EnableItem( SimMenu, RunCmd);
                 EnableItem( FileMenu, CloseCmd);
                 EnableItem( EditMenu, CopyCmd);
                end;

    OpenMap   :  begin
                 MapOpen;
                 EnableItem( EditMenu, CopyCmd);
                 end;

    FromScrap :  Begin
                  doNew;
                  Data.MapName := '';{ empty Mapname => can't save datfile}
                 end;
    SaveMap   : mapSave;
  end {of case }
end;  { doMapmenucommands }


Procedure ActivateEvents;
 Begin
  with theEvent Do
  Begin
   whichWindow := WindowPtr( message );
   SetPort( whichWindow );
   If BitAnd( modifiers, activeFlag ) <> 0
    then FixEditMenu( False )
    else FixEditMenu( True )
  end
end;
```

```
  Procedure SetupmenuBar;
  Begin
    appleMenu      := GetMenu( Appleid );   { read menu resources }
    fileMenu       := GetMenu( FileID );
    editMenu       := GetMenu( EditID );
    SimMenu        := GetMenu( SimID );
    MapMenu        := GetMenu( MapID);

    InsertMenu( AppleMenu, 0 );
    InsertMenu( fileMenu, 0 );
    InsertMenu( editMenu, 0 );
    InsertMenu( SimMenu, 0 );
    InsertMenu( MapMenu, 0 );
    AddresMenu( appleMenu,'DRVR' ); { add desk accessory names }
    DrawMenuBar
  end;


  Function QuitConfirmed : Boolean;
   { shut the sucker down }
  begin
    if quitRequested then
       if windowopen then CloseProgramwindow;
    QuitConfirmed := quitrequested;
  end;  { quit }


Procedure DoSystemTasks;
Begin
 SystemTask;
 If FrontWindow = NIL Then
   begin
     FixEditMenu( False );
     EnableItem( MapMenu, NewMap );
     EnableItem( MapMenu, OpenMap );
     EnableItem( MapMenu, FromScrap);
     EnableItem( MapMenu, SaveMap );
     EnableItem( EditMenu, PasteCmd);
     DisableItem( fileMenu, CloseCmd );
   end Else
       If FrontWindow <> wPtr Then
       begin
        FixEditMenu( True );
        EnableItem( fileMenu, CloseCmd )
       end  { else / if }
end;  { dosystemtasks }



 Procedure Docommand( command :longint );
   { execute a menu command }
 var
    whichmenu      : integer;
    whichitem      : integer;
 begin
   whichmenu := hiword( command  );
   whichitem := loword( command );
   case whichmenu of
    appleid     : doapplemenucommands( whichitem );
    fileid      : dofilemenucommands( whichitem );
    editid      : doeditmenucommands( whichitem );
    simid       : begin
                    SetupWindow;
```

```
                    EraseRect(wptr^.portRect);
                    simulate;  UnloadSeg(@simulate)
                  end;

    Mapid      : doMapMenucommands( whichitem);



    { add other program menus here }
   end; { case }

   hilitemenu( 0 ) { unhighlit menu title }
 end;  { docommand }




procedure MouseDownEvents;
 { check location and respond to mouse button }

var
  partCode : Integer;    { what item was clicked }
begin
      with theEvent do
      begin
      partCode := FindWindow( where, whichWindow );
      case partCode of
        inMenuBar      : docommand( menuselect( where ) );
        insyswindow    : systemclick( theEvent, whichwindow );
        incontent      : if whichwindow <> frontwindow
                            then selectwindow( whichwindow )
                            else doMouseClick( whichwindow );
        inDrag         : Dragthewindow( whichWindow, where );
        ingrow         : if whichwindow <> frontwindow
                            then selectwindow( whichwindow )
                            else resizeWindow(whichwindow, theevent.where);
        ingoaway       : if trackgoaway(whichwindow, where )
                            then doclose;
        inzoomin, inzoomout
                       : if trackbox( whichwindow, where, partcode )
                            then zoominout( whichwindow, partcode )
       end  { case }
      end { with }
 end;  { mousedownevents }


 procedure keydownevents;
  { a key was pressed, do something with char }
 var
    ch : char;
 begin
   with theEvent do
   begin
     ch := chr(BitAnd( message, charCodeMask ) ); { get character }
     if BitAnd( modifiers, CmdKey ) <> 0 { in command key pressed }
        then doCommand( MenuKey( ch ) )  { then execute command }
        else doKeypress( ch )            { else use character }
   end { with }
 end;  { keydownevents }


{ MAIN Program }

Begin
    Data.MapName := '';
    ph    :=  PicHandle(NewHandle(0));
```

```
    Hndl :=  NewHandle( 0);
    pp1  :=  Newptr( 0);
    SetUpMenuBar;
    quitRequested := False;
    windowOpen := False;
    DisplayAboutBox;
  Repeat
    DosystemTasks;
    If GetNextEvent( everyEvent, theEvent ) then
      case theEvent.what of
        MouseDown    : mouseDownEvents;
        KeyDown      : keyDownEvents;
        AutoKey      : { ignored } ;
        ActivateEvt : ActivateEvents
      end  { of Case }
   until QuitConfirmed
end.
```

```
UNIT BirdSim( 130 );


{$O Bird:Units.F: Sim.Lib }
{$U Bird:Units.F: Sim.Lib }

INTERFACE


 USES

 Memtypes, QuickDraw, OSIntf, ToolIntf, PackIntf, sane, MacPrint,
 BirdGlobals,MacExtras, Dialogunit ;

 type
    tcall = record
    location    : point;
    destination : point;
    priority    : integer; {  0= can wait .. 2=>go to 2-cntr, 3=> to 3-cntr }
    time        : extended;
    jobtyp      : integer;  { 3 = transfer, 4 = accident }
   end;




  Procedure Paintcircle(pt:point; r:real);
  Procedure Printresult;
  Function  Findist( p1, p2 : point) : extended ;
  Procedure addq (x : tcall);
  Procedure Delq;
  Function Expo (x : extended) : extended;
  Function Norm (u, v :extended) : extended;
  Procedure Nextevent;
  Procedure Choosbird (p1, p2 : point;
                   var distance,wtime, rtime, ftime   : extended;
                   var                     avail      : integer  );
  Procedure Genaccident;
  Procedure Choosdest (var dest, loc : point;
                    priority : integer;  var  distance : extended);
  Procedure Gentransf;
  Procedure Fly;
  Procedure Simulate;
  procedure initialize;



  IMPLEMENTATION

 Const  one = 1;
        keyReturn = 36;
        keyEnter  = 76;

        QuestId          =1010;
        WindowID2        =1001;
        AskId            =1000;     { resource ID for ask Dialog }
        RplId            =1020;     { resource ID for report Dialog}
        HospQId          =1030;
        BirdQId          =1040;

        IstEditItemAsk      = 3;
        IstEditItemRpl      = 5;
        IstEditItemHpQ      = 2;
```

```
      IstEditItemBdQ      = 2;

      MaxFieldAsk         = 10;
      MaxFieldRpl         = 6;
      MaxFieldHpQ         = 3;
      MaxFieldBdQ         = 2;

      NMaccidItem      = 1;
      PerachiItem      = 2;
      PerctrmaItem     = 3;
      PertranhiItem    = 4;
      PercrprItem      = 5;
      DurrprItem       = 6;
      percbwthrItem    = 7;
      durwthrItem      = 8;
      PickuptItem      = 9;
      RespondtItem     = 10;


Var

 wRec2           : windowRecord;
 wPtr2           : WindowPtr;

 ItemHandles                : ARRAY[1..MaxFieldAsk] of Handle;
 Pnt                                          : Point;
 dPtr                                         : DialogPtr;
 itemNo                                       : integer;

 hirgn, tmprgn                                : rgnhandle;
 ansrct, tmpRect                              : rect;
 ii,jj,tcount,birdnum, eventyp                : integer;
 died, destcntr, toofar                       : integer;
 xliml,xlen,yliml,ylen                        : extended;
 simtime, clock, min, rtime, ftime            : extended;
 inref, outref, nextweather                   : extended;
 percinl                                      : extended;

 refuse                      : array[1..nbds, 1..4] of integer;
                               { 1-repair,2-weather,3-trans,4-accid }
 nextrepair                  : array[1..nbds] of extended;
 nextaccident, nexttransfer  : tcall;
 transferq                   : array[1..100] of tcall;

 DefStatus,DefNum,DefSpeed,DefRange    : longint; { default variables }
 DefName                               : String255;


Procedure MakeHist( M,N : Integer; Hst: Ivect; F : Rect );
{ Make a histogram of the values in Hst  }
{ each N values to be grouped together }
{ M is the Max Index of Hst to be considered. ( N should div M+1 ) }

Var
  P1,P2,P3, P4,P5       : Point;
  Bar                   : Rect;
  I,J,Num,tmp           : Integer;
  Ht,Wth,d1,Max         : Integer;
  Count                 : Ivect;
  Hmult                 : real;
  Pat                   : Array[1..5] of Pattern;

Begin
  PenPat( Black);
  Pat[1]:=White; Pat[2]:=LtGray;
  Pat[3]:=Gray;Pat[4]:=DkGray;Pat[5]:=Black;
```

```
  P1 := F.TopLeft;
  P2 := F.BotRight;
  P3.h := P1.h; P3.v:=P2.v;
  Ht  := P2.v - P1.v;
  Wth := P2.h - P1.h;
  Num := (M+1) div N;
  d1  :=  Wth div Num;
  Max := 0;
  For I := 0 to Num-1 do
    begin        { partition Hst into Num subgroups of size N, then sum }
     tmp := 0;
     for J := 0 to N-1 do tmp := tmp+Hst[(I*N)+J];
     Count[I] := tmp;
     if ( tmp > Max ) then Max := tmp;
    end;
  Hmult := Ht / Max;    {    ********    error if Max = 0   ***** }

  P5.v := P2.v;
  For I := 0 to (Num-1) do
      begin
       P4.h := P1.h + I*d1;
       P4.v := p2.v - round(Count[I]*Hmult);
       P5.h := p4.h + d1;
       moveto(p5.h,p5.v); Lineto(p5.h,p5.v-1);
       Pt2Rect(p4,p5,Bar);
       FillRect(Bar,Pat[ ((I mod 5) +1) ]);
       FrameRect(Bar);
      end;
  moveto(p1.h,p2.v); Lineto(p2.h,p2.v);
   FOR I:= 0 TO MAX DO
    begin
     d1 := round(i*Hmult);
     moveto(p1.h,p2.v-d1); lineto(p1.h+1,p2.v-d1);
    end;

 end; { of MakeHist }



Procedure VartoDialog( field : Str255; itemNum : Integer );
Begin
 SetIText( itemHandles[ itemNum ], Field );
end;

Procedure DialogToVar( Var field : Str255; len, itemNum : Integer );
Begin
 GetIText( itemHandles[ itemNum ], field );
 If Length( field) > len
    Then field := copy( field,1,len)
 end;

Procedure asktoDialog;
var
  s  : str255;
  inmaccid,iperachi,iperctrma,ipertranhi,ipercrpr,idurrpr  : integer;
  ipercbwthr,idurwthr,ipickupt, irespondt                  : integer;

Begin
With Data do
begin
 inmaccid   := trunc(NMaccid);
 iPerachi   := trunc(perachi * 100);
 iperctrma  := trunc(perctrma * 100);
 ipertranhi := trunc(pertranhi * 100);
 ipercrpr   := round(percrpr*168.0);
```

```
 idurrpr    := trunc(durrpr);
 ipercbwthr := round(percbwthr*720.0);
 idurwthr   := trunc(durwthr);
 ipickupt   := trunc(pickupt);
 irespondt  := trunc(respondt);

 SeliText(dPtr, IstEditItemAsk, 0, 0);
 NumToString(iNMaccid,s);   vartoDialog(s,NMaccidItem  );
 NumToString(iPerachi,s);   vartoDialog(s,PerachiItem  );
 NumToString(iperctrma,s);  vartoDialog(s,PerctrmaItem );
 NumToString(ipertranhi,s); vartoDialog(s,pertranhiItem);
 NumToString(iPercrpr,s);   vartoDialog(s,PercrprItem  );
 NumToString(idurrpr,s);    vartoDialog(s,durrprItem   );
 NumToString(ipercbwthr,s); vartoDialog(s,percbwthrItem  );
 NumToString(idurwthr,s);   vartoDialog(s,durwthrItem   );
 NumToString(ipickupt,s);   vartoDialog(s,pickuptItem  );
 NumToString(irespondt,s);  vartoDialog(s,respondtItem );
 SeliText(dPtr,IstEditItemAsk,0,MaxInt);

end;
end;

Procedure DialogToAsk;
var
 s : Str255;
  inmaccid,iperachi,iperctrma,ipertranhi,ipercrpr,idurrpr   : longint;
  ipercbwthr,idurwthr,ipickupt, irespondt                   : longint;

begin
with Data do
begin

 DialogtoVar(s,6,NMaccidItem);    StringToNum(s,iNMaccid  );
 DialogtoVar(s,6,PerachiItem);    StringToNum(s,iPerachi  );
 DialogtoVar(s,6,perctrmaItem);   StringToNum(s,iperctrma );
 DialogtoVar(s,6,pertranhiItem);  StringToNum(s,ipertranhi);
 DialogtoVar(s,6,PercrprItem);    StringToNum(s,iPercrpr  );
 DialogtoVar(s,6,durrprItem);     StringToNum(s,idurrpr   );
 DialogtoVar(s,6,percbwthrItem);  StringToNum(s,ipercbwthr  );
 DialogtoVar(s,6,durwthrItem);    StringToNum(s,idurwthr   );
 DialogtoVar(s,6,pickuptItem);    StringToNum(s,ipickupt  );
 DialogtoVar(s,6,respondtItem);   StringToNum(s,irespondt );

 nmaccid   := iNMaccid;
 Perachi   := iperachi * 0.01;
 perctrma  := iperctrma * 0.01;
 pertranhi := ipertranhi * 0.01;
 percrpr   := ipercrpr/168.0;
 durrpr    := idurrpr;
 percbwthr := ipercbwthr/720.0;
 durwthr   := idurwthr ;
 pickupt   := ipickupt;
 respondt  := irespondt;
 end;
end;


Procedure Question( query : str255; var answer : str255 );
{ sets up dialog to ask 'query' and get answer 'answer' }
Var
  itemRect                      : Rect;
  itemType,itemNo               : integer;
  item                          : Handle;
begin
  Flushevents(Everyevent,0);
  dPtr := GetNewDialog( QuestID, NIL, Pointer(-1) );
```

```
  ParamText(query,'','','');
  itemNo := 4;
  GetDItem(dPtr, itemNo, itemType, item, itemRect);
   SeliText(dPtr, itemNo, 0, 0);
    Repeat
    ModalDialog(NIL, itemNo);
    Until (itemNo = ok ) ;
  GetIText( item, answer );
  DisposDialog(dPtr);
end;



Procedure SetUpDialog( ID, MaxField, IstEditItem: integer);
  var
    itemType ,itemNo                : integer;
    item                            : Handle;
    editArea, ButtonArea,itemRect   : Rect;

   begin
   Flushevents(Everyevent,0);
   dPtr := GetNewDialog( ID, NIL, Pointer(-1) );
   if dPtr = NIL
     then ExitToShell;
   for itemNo := 0 to MaxField - 1 do
     begin
       GetDItem(dPtr, itemNo + IstEditItem, itemType, item, itemRect);
       if item = NIL then ExitToShell;
       itemHandles[ itemNo + 1 ] := item;
     end;  { for }
   editArea := dPtr^.portRect;  {   *******  modify this   ******  }
   buttonArea := editArea;
   GetDItem(dPtr, 1, itemType, item, itemRect );    { 1'st button }
   with itemRect do
      begin
        top := top-4;
        editArea.bottom := top;
        buttonArea.top := top;
      end;
  end;   { setupdialog  }



Procedure DoAsk;
  Var
  itemType                        : integer;
  item                            : Handle;
  editArea, ButtonArea,itemRect   : Rect;


begin
  SetupDialog(AskID,MaxFieldAsk,IstEditItemAsk)   ;
  OutlineOK( dPtr);
  AsktoDialog;
  Repeat
    ModalDialog(NIL, itemNo)
  Until (itemNo = ok ) or ( itemNo = Cancel);
  If itemNo = ok then Dialogtoask;
  DisposDialog(dPtr);
end; { DoAsk }



Procedure RpltoDialog;
var
```

```
  s                      : string255;
  i                      : integer;
  Dat                    : array[1..6] of longint;
  tmp2,tmp3,tmp4         : real;
  itmp                   : longint;

Begin
for i:=1 to MaxFieldRp1 do dat[i]:=0;
 { dat contains the edit items, index i corresponds to item i }
tmp2 := 0; tmp3 := 0; tmp4 := 0;
With Data do
begin
     itmp:= round(Simtime*52/Minyear);
     NumToString(itmp,s);
     s := Concat(s,' Week Simulation Time ');
     ParamText(s,'','','');
 for i:=1 to Nbds do
 with Bird[i] do
  begin
   dat[1] := dat[1] + nflts;          { total flights }
   tmp2 := tmp2 + CallDist/convert;   { total distance }
   tmp3 := tmp3 + waitt;              { wait time }
   tmp4 := tmp4 + Waitt + resct;      { total time }
  end;
 Dat[2] := round( tmp2/dat[1]);      { find averages per flight }
 dat[3] := round( tmp3/dat[1]);
 dat[4] := round( tmp4/dat[1]);
 dat[5] := Num2LongInt(outref);            { out of range refused }
 Dat[6] := Num2LongInt(inref);             { in range refused }

 SeliText(dPtr, IstEditItemRp1, 0, 0);
 for i:=1 to MaxFieldRp1 do
      begin
       NumToString(dat[i],s);
       vartoDialog(s,i);
      end;
  SeliText(dPtr,1,0,MaxInt);
 end;
end;

Procedure HospReport(Num: integer); {what happend at Num hospital }
 var
  i, j, k, d2    : integer;
  await, aresc   : integer;
  Dat            : array[1..7] of longint;
  Ds             : array[1..7] of string255;
  s,s2,s3        : string255;
  x,y,del        : integer;
  k1,k2,k3       : longint;


  Procedure MyNumtostring(k:longint; n:integer;var s:string255);
   begin
    NumToString(k,s);
    while(length(s) < n) do Insert( ' ',s,1);
   end;

begin
 x:=1;y:=40;del:=24;
 EraseRect(wptr2^.portRect);
 with Data do
  with hosp[Num] do
  begin
  k := 0; await := 0; aresc := 0;
  gotoxy(1,2);
  writeln(name:10, '   Status =',status:3, '  Flights Received =',rcvd:6 );
```

```
  if (Numinest > 0) then
   begin
    Moveto(x,y);
    Drawstring('   craft   Rnge  N-Flts Av Rsc  Av wait   Flt Hrs  Av Dst');
    y:=y+11;
    Moveto(x,y);
    Drawstring('                                         (min.)      (min.)    /week        ');
     for i := 1 to numinest do
      begin
       y:=y+del;
       Moveto(x,y);
       with bird[fleet[i]] do
         if (nflts > 0) then
         begin
          dat[1]:= i;
          dat[2]:= round(range/convert);
          dat[3]:= nflts;
          dat[4]:= round((resct + Waitt)/nflts);
          dat[5]:= round(waitt/nflts);
          dat[6]:= round(168.0*fltime/simtime);
          dat[7]:= round( (calldist/convert)/nflts );
             k   := k + nflts;
          await := await + round(waitt);
          aresc := aresc + round(resct + waitt);
          for j:=1 to 7 do MyNumToString(dat[j],9,ds[j]);
          s := concat(ds[1],ds[2],ds[3],ds[4],ds[5],ds[6],ds[7]);
          DrawString(s);
            d2 := trunc(del/2)-3;
            { draw an arrow pointing at the average in the Histogram }
             pnt.h:=round(350+(float(dat[7])/500.0)*150.0);
   {   500 mile range of the histogram, 150 = length of histogram}
   { things will need to be changed if craft have range >500 mi}
             pnt.v:= y-d2;
             Moveto(300,pnt.v-4);
             Lineto(pnt.h,pnt.v-4); Lineto(pnt.h,pnt.v);
             moveto(pnt.h-2,pnt.v-3);
             Lineto(pnt.h,pnt.v); Lineto(pnt.h+2,pnt.v-3);

            SetRect(tmpRect,350,y-d2,500,y+d2);
            PenPat(Black);
            MakeHist( 49,2 ,dHist , tmpRect );{ the 49 => 50 int => 500 mi}
         end;
     end
    end;
   y:=y+12;
   Moveto(x,y);
 DrawString('_______________________________________________________________');
 y:=y+12;
 Moveto(x,y);
 if (k > 0) then
   begin
   k1:=k; k2:=round(aresc/k); k3:= round(await/k);
   MyNumtoString(k1,9,s); MyNumtoString(k2,9,s2); MyNumtoString(k3,9,s3);
   s := Concat('                     ',s,s2,s3);
   DrawString(s);
   end;

end
end;



Procedure DoReport;
  Var
  itemType,cnum                  : integer;
```

```
  item                           : Handle;
  editArea, ButtonArea,itemRect  : Rect;
  hPrint                         : THPrint;
  thePPort                       : TPPrPort;
  prStatus                       : TPrStatus;

begin  { of DoReport }
     wPtr2 := GetNewWindow( WindowID2, @Wrec2, Pointer( -1 ) );
     SetPort( wPtr2 );                 { new window to put hosp reports in }
     ClipRect( wPtr2^.portRect );
     cnum := 1;

  SetupDialog(Rp1ID,MaxFieldRp1,IstEditItemRp1)  ;
  OutlineOK( dPtr);
  Rp1toDialog;
  HospReport(cnum);
  Repeat
    SetPort( wPtr2 );
    ModalDialog(NIL, itemNo);
    if  (itemNo = 1) then    { display next hosp report }
      begin
       cnum := cnum + 1; if ( cnum > Data.numhsp) then cnum := 1;
       HospReport(cnum);
       SysBeep(5);
      end;
    if (itemNo = 3) then      { Send Hosp reports to printer }
      begin
       PrOpen;                                    { lets try printing it }
       hPrint := THPrint(NewHandle(sizeof(TPrint)));
       PrintDefault(hPrint);
       if PrJobDialog(hPrint) then
        Begin
         thePPort := PrOpenDoc(hPrint,NIL,NIL);
           PrOpenPage(thePPort,NIL);
           printresult;
           PrClosePage(thePPort);
         PrCloseDoc(thePPort);
         PrPicFile(hPrint,NIL,NIL,NIL,prStatus)
        end;
       DisposHandle(Handle(hPrint));
       PrClose;
      end;

  Until (( itemNo = 2) or (itemNo = 4) ) ;{ 2=ok button, 4=Another sim }
  DisposDialog(dPtr);
  closewindow(wptr2 );
  SetPort( wPtr );

 end; { DoReport }



 Procedure DoHospDiag( var  Name: string255; var Status, Number : integer);
   Var
   s     : String255;
   tmp   : Longint;
  begin
   SetupDialog(HospQId,MaxFieldHpQ, IstEditItemHpQ) ;
   OutlineOK( dPtr);
         tmp := DefStatus;
         NumToString(tmp,s);
         vartoDialog(s,2);
         tmp := DefNum;
         NumToString(tmp,s);
         vartoDialog(s,3);
```

```
      SeliText(dPtr,IstEditItemHpQ,0,0);
      Repeat
      ModalDialog(NIL, itemNo);
      Until ( itemNo = ok) ;
    DialogtoVar(Name,8,1);
    DialogtoVar(s,6,2);    StringToNum(s,tmp ); Status := tmp;
    DialogtoVar(s,6,3);    StringToNum(s,tmp ); Number := tmp;
  DisposDialog(dPtr);

  DefStatus := Status;
  DefNum    := Number;
 end; { DoHospDiag }


Procedure DoBirdDiag(Name :string255; Num:integer; var Speed,Range:extended);
  Var
  s     : String255;
  tmp   : Longint;
 begin
  SetupDialog(BirdQId,MaxFieldBdQ, IstEditItemBdQ) ;
  OutlineOK( dPtr);
     tmp:= Num;
     NumToString(tmp,s);
     s := Concat('Rotorcraft: #',s); { identify which bird }
     ParamText(s,'','','');

       tmp := DefSpeed; NumToString(tmp,s);  vartoDialog(s,1);{ put def val}
       tmp := DefRange; NumToString(tmp,s);  vartoDialog(s,2);
      SelIText( dPtr,IstEditItemBdQ,0,MaxInt);
      Repeat
      ModalDialog(NIL, itemNo);
      Until ( itemNo = ok) ;
    DialogtoVar(s,6,1);    StringToNum(s,tmp ); Speed := tmp;
    DialogtoVar(s,6,2);    StringToNum(s,tmp ); Range := tmp;
  DisposDialog(dPtr);

  DefSpeed := trunc(Speed);
  Defrange := trunc(Range);
 end; { DoBirdDiag }



Procedure Paintcircle;
 var  box : rect;
      i   : integer;
 begin
    i := round(r);
    Setrect( box, pt.h - i, pt.v - i, pt.h + i, pt.v + i);
    FrameOval( box );
 end;


Function Findist ;

 var dist1,tp1,tp2 : extended;
 begin
   { gotoxy(2,1);cleareol;write(' FINDIST ');    }

   tp1:=p1.v - p2.v;
   tp2:=p1.h - p2.h;
```

```
    distl := tp1*tp1 + tp2*tp2;
    if ( distl < 0.001 ) then distl := 0.0;
    findist:=sqrt(distl);
  end;



procedure clickpoint (var pt : point);
  var
   h, v : integer;
begin
  repeat
  until button;
  repeat
   getmouse(pt);
  until not (button);
  end;


procedure loccenters; {proc to locate and initialize hospitals}
  var
   done               : boolean;
   pt, p1, p2, p3     : point;
   i, j, h, v         : integer;
   itemp              : longint;
   s                  : string255;

begin
  gotoxy(1,1); cleareol;
  write('  Use mouse to indicate a distance of 100 miles   ');
  repeat
  until button;
  getmouse(p1);
  p2 := p1;
  penpat(Gray);
  penmode(patXor);
    repeat
     moveto(p1.h,p1.v);
     lineto( p2.h, p2.v);
      repeat
        Getmouse(p3);
      until not Equalpt(p2, p3);
     moveto( p1.h, p1.v);
     lineto( p2.h, p2.v);
     p2 := p3;
    until not Button;
  Penpat(black);
  Penmode(Patcopy);
  moveto( p1.h, p1.v);
  lineto( p2.h, p2.v);

  with Data do
  begin
  numbird:= 0;
  convert := findist(p1,p2)/100.0;
  done := false;
  i := 1;

   gotoxy(1,1); cleareol;
   writeln('locate hospitals by clicking with mouse   ');
   gotoxy(1,2); cleareol;
   writeln(' click in "OK" box after all hospitals have been located   ');

  while (done = false) do
   begin
    clickpoint(pt);
```

```
    if ptinrect(pt, ansrct) then
    done := true
   else
    begin
     paintcircle(pt, 2);
     with hosp[i] do
      begin
       location.h := pt.h;
       location.v := pt.v;

       DoHospDiag(Name,Status,numinest);

       if (numinest > 0) then
        begin
         paintcircle(pt, 4);

         for j := 1 to numinest do
          begin
           numbird := numbird + 1;
           with bird[numbird] do
            begin

             DoBirdDiag(Name,j,speed,range);

             speed := speed * convert / 60.0;
             range := (range * convert);
             base := location;
             jobtyp := 0;
             timer := -1.0;
             busy := false;
             nflts:=0; fltime:=0.0; rptime:=0.0;
             waitt:=0.0; resct:=0.0; cumtot:=0;
            end;
           fleet[j] := numbird;
          end  { of for }
        end;
       if (status < 3) then
        begin
         Question('No. of ground transport emergencies this center treats per year?',s);
         stringToNum(s,itemp);
         totemerg := totemerg + itemp;
         cumtot := totemerg;
        end;
         rcvd:=0; emrgperday:=0;
      end;
    i := i + 1;

       moveto(ansrct.topleft.h+3,ansrct.botright.v-1);
       Drawstring('OK');
       Framerect(ansrct);
      { DrawPicture( ph ,MapFrame);   }
       Framerect(MapFrame);

    end
  end;

 numhsp := i - 1
end
end;


procedure Rinput;

var i,j, itmp        : Integer;
    tmp              : extended;
    str1             : string255;
```

```
 procedure iwrtrd(strl : string255;   var itmp : integer);
  begin
     gotoxy(1,2); cleareol;
     write(str1,itmp:5,' ');
     readln(str1);
     if (length(str1) > 0)  then itmp := round(Str2Num(str1));
  end;

  procedure wrtrd(strl : string255;   var tmp : extended);
  begin
     gotoxy(1,2); cleareol;
     write(str1,tmp:8:2,' ');
     readln(str1);
     if (length(str1) > 0)  then tmp := Str2Num(str1);
  end;

begin

 with Data do
 begin
  totemerg := 0;
  for i:=1 to numhsp do
  begin
    with hosp[i] do
    begin
    gotoxy(1,1); cleareol; write(' Hospital Number   : ',i);
    iwrtrd(' status = ',status);
    if (numinest > 0)  then
      for j:=1 to numinest do
       with bird[fleet[j]] do
            begin
            speed := speed*60.0/convert;
            range := range/convert;
            gotoxy(1,1);cleareol; write(' Rotorcraft Number   : ',j);
            itmp:=round(speed);
            iwrtrd(' speed =      ',itmp);
            speed:=itmp;
            itmp:=round(range);
            iwrtrd(' range =      ',itmp);
            range:=itmp;
             speed := speed * convert / 60.0;  { convert back for screen }
             range := (range * convert);
            end;
    if (status < 3) then
        begin
          if (i=1) then itmp := cumtot
             else itmp := (cumtot - hosp[i-1].cumtot);
          gotoxy(1,1); cleareol; write(' Hospital Number   : ',i);
          iwrtrd(' No. of ground transport emergencies per year  :    ',itmp);
          totemerg := totemerg + itmp;
          cumtot := totemerg;
        end;
        gotoxy(1,1); cleareol;
        gotoxy(1,2); cleareol;
        rcvd:=0; emrgperday:=0;
    end
  end
 end
end;

Procedure Drawranges;
var i, irange   : integer;
    rl          : rect;
begin
with Data do
```

```
 begin
 for i := 1 to numbird do
  with bird[i] do
   begin
    irange := trunc( range);
    setrect(r1, base.h - irange, base.v - irange, base.h + irange, base.v + irange);
    FrameOval(r1);
   end
  end
 end;


procedure locrect;
 var
  i, h1, h2, v1, v2           : integer;
  r1, r2                      : rect;
  done                        : boolean;
  p1, p2, p3                  : point;
  s                           : str255;
  itemp                       : longint;
  x1,x2,y1,y2                 : integer;
begin
 with Data do
 begin
 gotoxy(1,2); cleareol; gotoxy(1,1); cleareol;
 writeln(' Use mouse to enclose high accident regions   ');

 done := false;
 i:=1;
 rgncnt := 0;    { count of # of points used to def region }
 repeat          { until done = true }
  repeat
  until button;

  getmouse(p1);
  p2 := p1;
  rgnpts[i]:=p1;  { this is to keep a record of the region defining points }
  PenPat(Gray);
  Penmode(PatXor);
   repeat
    Pt2Rect(p1, p2, r1);
    FrameRect(r1);
    repeat
     GetMouse(p3);
    until not EqualPt(p2, p3);
    FrameRect(r1);
    p2 := p3;
   until not Button;
   rgnpts[i+1] := p3;  { 2nd coord of region defining rect }
  i:=i+2;
  Penpat(Black);
  PenMode(PatCopy);
  if ptinrect(p1, ansrct) then
   done := true
  else
   begin
    FrameRect(r1);
    tmprgn := Newrgn;
    rectrgn(tmprgn, r1);
    Unionrgn(hirgn, tmprgn, hirgn);
    DisposeRgn(tmprgn);
    rgncnt := rgncnt + 2;   { count of # of points used so far }
   end;
 until (done = true);       { finished defining region }

 p1:=  rgnpts[1];
```

```
 xl:=pl.h; yl:=pl.v; x2:=xl; y2:= yl; { looking for smallest and largest }
 for i:=2 to rgncnt do
   begin
    p3:=rgnpts[i];
    If ( p3.h < xl) then xl := p3.h;
    if ( p3.h > x2) then x2 := p3.h;
    if ( p3.v < yl) then yl := p3.v;
    if ( p3.v > y2) then y2 := p3.v;
   end;
   xlim2 := xl;               { smallest x-val of hirgn }
   xlen2 := (x2-xl)/32767.0;  { scaled dist to highest x val }
   ylim2 := yl;               { smallest y val }
   ylen2 := (y2-yl)/32767.0;  { scaled dist to highest y val }
 end;
Drawranges;
end;


procedure Redraw;

var r            : rect;
    i,irange     : integer;
    ans          : char;
    pt           : point;

begin
 EraseRect(theport^.portRect);
 moveto(ansrct.topleft.h+3,ansrct.botright.v-1);
 Drawstring('OK');
 Framerect(ansrct);
 DrawPicture( ph ,MapFrame);
 Framerect(MapFrame);

 with Data do
  begin
  for i:=1 to numhsp do
  with hosp[i] do
       paintcircle(location, status);

  PenPat(Gray);
  Penmode(PatXor);
  frameRgn( hirgn );

  Penpat(Black);
  PenMode(PatCopy);
end;
Drawranges;
end;

Procedure randomize;  { start new random sequence }
var time : longint;
 begin
  GetDateTime(time);
  RandSeed := time;
 end;



Function getdatfile( VAR done : Boolean ) : Boolean; { Now in Bird4.pas }
var
   okayflag            : boolean;
   folder, Filename    : String[64];
   reply               : SFReply;
begin
 okayflag := false;
```

```
  inName := '';
  if GetFileName(reply,'DATA') then inName := reply.fName;
  done := length( inName ) = 0;
  If not done Then
  begin
    {$i-} Reset( datafile, inName );
          read( datafile, Data );
          close(datafile);
    {$i+}
    If IoResult <> 0
    then begin
         gotoxy(1,1); cleareol;Write(' ERROR: cannot find/read ', inName )
         end
    else
      okayflag := ( IoResult = 0 );
   end;
  getdatfile := okayflag;
 end;

 Function putdatfile( VAR done : Boolean ) : Boolean;
 var
    okayflag            : boolean;
    folder, Filename    : String[64];
    reply               : SFReply;

 begin
  okayflag := false;
  outName := '';
  if makeFileName(reply, 'Save file as', outName) then outName:= reply.fName;
  done := length( outName ) = 0;
  If not done Then
  begin
    {$i-} FileType := 'DATA';
          Rewrite( datafile, outName );
          write(datafile, Data );
          close(datafile);
          FileType := 'BINA';
    {$i+}
    If IoResult <> 0
    then
     begin
     gotoxy(1,1); cleareol;
     Write(' ERROR: cannot create ', outName, '  press "return" '); readln;
     end
    else
      okayflag := ( IoResult = 0 );
   end;
  putdatfile := okayflag;
 end;



procedure Initvars;
var i,j  : integer;

begin;
with Data do
  begin

  DefStatus :=3;
  DefNum    :=1;
  DefSpeed  :=150;
  DefRange  :=150;

  birdnum:=0; tcount:=0;totemerg := 0;
  clock := 0.0; avwait := 0.0; avresc := 0.0;
```

```
  inref := 0.0; outref := 0.0;
  died:=0; toofar:=0;
  xlim1 := MapFrame.topleft.h; xlen := (MapFrame.botRight.h - xlim1)/32767;
  ylim1 := MapFrame.topleft.v; ylen := (MapFrame.botRight.v - ylim1)/32767;
  for i := 1 to nbds do
    with Bird[i] do
    begin
     for j:=0 to 50 do dHist[j]:=0;
     busy := false; timer := 0; nflts := 0; fltime := 0; rptime := 0;
     waitt := 0; resct := 0; calldist := 0;
    end;

     for i:=1 to numhsp do
      with hosp[i] do
            begin rcvd:=0;
                  emrgperday:=0
            end;

  end;
end;



 procedure initialize;

  var
   i          : integer;
   itemp      : longint;
   r,r1       : rect;
   s          : str255;
   rt,ok      : Boolean;
   pt,pt1     : point;
 begin
  randomize;
  Initvars;
  ClipRect(MapFrame);
  DrawPicture( ph ,MapFrame);
  Framerect(MapFrame);
  ClipRect(theport^.portRect);
  pt:=MapFrame.botRight;
  pt1:=MapFrame.topleft;
  SetRect(ansrct,pt.h-25,pt1.v-15, pt.h,pt1.v);
  moveto(ansrct.topleft.h+3,ansrct.botright.v-1);
  Drawstring('OK');
  Framerect(ansrct);
  with Data do
   begin
       if ( readdat = true) then
          begin
           hirgn := Newrgn;
           i:=1;
           repeat
            pt2rect(rgnpts[i],rgnpts[i+1],r1);
            tmprgn := Newrgn;
            rectrgn(tmprgn, r1);
            Unionrgn(hirgn, tmprgn, hirgn);
            DisposeRgn(tmprgn);
            i:=i+2;
           until (i > rgncnt);
          end
       else
        begin
         hirgn := Newrgn;
         moveto(ansrct.topleft.h+3,ansrct.botright.v-1);
         Drawstring('OK');Framerect(ansrct);
         loccenters; { procedure to locate centers }
```

```
        locrect;    { procedure to locate high accid regions }
       end;  { of data definition section }

  Redraw;
  if (readdat = true ) then
           begin
            Question('Do you want to enter new rotor craft data ? (Y/N) ',s);
            if (s = 'Y') or (s = 'y') then Rinput;
           end;

 Question('What % of the accidents occur in the high accident region? ',s);
 stringToNum(s,itemp);
 percin  := 0.01 *itemp;
 percinl := percin * 65535.0 - 32768.0;
 DoAsk;

 rt:=false;
  If ( length(MapName) > 0 ) then
     begin
     Question('Do you want to Save the Data in a file ? (Y/N) ',s);
     if ( s = 'y' ) or (s = 'Y') then rt := putdatFile(ok);
     end;
 Question('Enter simulation time ( in weeks ): ',s);
 stringToNum(s,itemp);

 ClipRect(MapFrame);
 Redraw;
 ClipRect(theport^.portRect);

 gotoxy(1,1); cleareol;
 write('                  ',itemp:5,'  Week  Simulation Time');

 launchdist := (10 * convert);
 simtime := itemp * minyear / 52;
 nmwthr  := percbwthr * minyear/ durwthr;
 nmrpr   := percrpr * minyear / durrpr;

 end
end;




Procedure Printresult; {what happend }
 var
  i, j, k : integer;
  tmp     : real;
begin
 with Data do
 begin
 Gotoxy(1,1); cleareol;
 k := 0;

 writeln('             ',round(simtime*52/minyear):4,'    Week  Simulation Time: ');

 writeln('_____________________________________________________________');

 writeln('Cntr Stat Rcvd  Bird  Rnge total   Av. resc.  Av. wait  Flt Hrs.   Av. Dist');
 writeln('-------------- num. ----- flts    (min.)      (min.)      /week');
 for j := 1 to numhsp do
  begin
   writeln(j:4, hosp[j].status :3, hosp[j].rcvd : 6);
   if hosp[j].numinest > 0 then
    begin
     for i := 1 to hosp[j].numinest do
      begin
```

```
        with bird[hosp[j].fleet[i]] do
         if (nflts > 0) then
          begin
           tmp := (calldist/convert)/nflts;
           write('                    ');
           write(i :4,round(range / convert) :8, nflts :6);
           writeln((resct+waitt)/nflts :10:1, waitt/nflts :9:1, (168.0 * fltime / simtime) :


           k := k + nflts;
           avwait := avwait + waitt;
           avresc := avresc + (resct+waitt);
          end;
       end
     end;
   end;
  writeln('______________________________________________________________');
  write(k : 17,'                         ');
  if (k > 0) then
   writeln(avresc / k :7 : 1, avwait / k : 10: 1);
  writeln('______________________________________________________________');
  write('There were   ', Num2Integer(outref) : 5, '  " out of range " calls and  ');
  writeln( Num2Integer(inref) : 3, '  non-responses due to');
  writeln('"in-service" or "in-repair"   ');

end
end;


Procedure addq ;
 var
  i, j : integer;
begin                { gotoxy(2,1);cleareol;write(' ADDQ '); }
 with Data do
 begin
 x.time := clock;
 paintcircle(x.location,1);
 if (tcount = 0) then
  transferq[1] := x
 else
  begin
           { gotoxy(1,1);cleareol; write('tcount =  ',tcount); readln; }
   j := tcount + 1;
   for i := tcount + 1 downto 2 do
    if (transferq[i - 1].priority < x.priority) then
     begin
      transferq[i] := transferq[i - 1];
      j := i - 1;
     end;
   transferq[j] := x;
  end;
 tcount := tcount + 1;
           { gotoxy(1,3);cleareol; write('tcount =  ',tcount); readln; }
end
end;

procedure Delq;
 var
  i : integer;
begin
                     {  gotoxy(2,1);cleareol;write(' DELQ ');   }
with Data do
begin
 tcount := tcount - 1;
 for i := 1 to tcount do
  transferq[i] := transferq[i + 1]
```

```
 end
 end;

Function Expo;
     { x is the average number of events per year }
     { the result is the exponential waiting time until next event }
   var
   r     : longint;
   y     : extended;
 begin
                              {  gotoxy(2,1);cleareol;write(' EXPO ');  }
  r := ABS(random) + one;
  y := -ln(r / 32768);
  expo := (y * minyear)/ x;
 end;

 Function Norm;
                    {produces normal variate with mean u and st dev=v }
  var
   u1, u2 : extended;
 begin
  u1 := abs(random / (32767));
  u2 := abs(random / (32767));
  norm := u + v * sqrt(-2 * ln(u1)) * cos(2 * pie * u2);
 end;

 Procedure Nextevent;
 {proc to choose next event, update clock,timers }
 {eventype 0->free,1->repair,2->weather,3->transfer,4->accid }
 var
   i     : integer;
   min   : extended;
 begin
                              {  gotoxy(2,1);cleareol;write(' NEXTEVENT ');   }
 with Data do
 begin
  min := 1E+20;
  if min > nextaccident.time then
   begin
    min := nextaccident.time;
    eventyp := 4;
   end;
  if min > nexttransfer.time then
   begin
    min := nexttransfer.time;
    eventyp := 3;
   end;
  for i := 1 to numbird do
   begin
    if ((bird[i].busy = true) and (min > bird[i].timer)) then
     begin
      min := bird[i].timer;
      eventyp := 0;
      birdnum := i;
     end;
    if (min > nextrepair[i]) then
     begin
      min := nextrepair[i];
      eventyp := 1;
      birdnum := i;
     end
   end;
  if min > nextweather then
   begin
    min := nextweather;
    eventyp := 2;
```

```
   end;
  nextaccident.time := nextaccident.time - min;
  nexttransfer.time := nexttransfer.time - min;
  for i := 1 to numbird do
   begin
    bird[i].timer := bird[i].timer - min;
    nextrepair[i] := nextrepair[i] - min;
   end;
  nextweather := nextweather - min;
  clock := clock + min;
 end
 end;


 Procedure choosbird;
 {wtime=time to arrive at scene, rtime=total rescue time( after start) }
 {ftime=flight time, avail 0-> found bird,1->in range but busy,2-> none in range }
 { p1= loc of call, p2= loc of destination of call }
 { distance = dist from call to his destination }

  var
   i                        : integer;
   min, dist1,  dist3       : extended;
   dst : array[1..nbds] of extended;
 begin
{  gotoxy(2,1);cleareol;write(' CHOOSBIRD ');     }
 with Data do
 begin
  avail := 2;
  min   := 1E+20; { min dist to avail bird so far }
  for i := 1 to numbird do
   with bird[i] do
   begin
    dist1 := findist(p1, base);
    dst[i] := dist1;
    if ((dist1 < range) and (avail = 2)) then  avail := 1;
    if ((min > dist1) and (busy = false)
                      and ((dist1 + distance) < ( range + range) ) )
     then
       begin
        avail   := 0;
        birdnum := i;
        min     := dist1;
       end
   end;  { with bird[i] }

  if (avail = 2) then
   outref := outref + 1;

  if (avail = 1) then
   begin
    inref := inref + 1;
    for i := 1 to numbird do
     begin
      if dst[i] < min then
       begin
        if ((bird[i].busy = true) and (bird[i].range > dst[i])) then
         case bird[i].jobtyp of
          1 :  refuse[i, 1] := refuse[i, 1] + 1;
          2 :  refuse[i, 2] := refuse[i, 2] + 1;
          3 :  refuse[i, 3] := refuse[i, 3] + 1;
          end   { of case }
       end      { of dst<min cond }
     end        { of i loop }
    end;        { of avail=1 cond }
```

```
 if (avail = 0) then
  begin
   dist3 := findist(p2, bird[birdnum].base);
   ftime := (min + distance + dist3) / bird[birdnum].speed;
   wtime := respondt + min / bird[birdnum].speed;
   rtime := wtime + pickupt + distance / bird[birdnum].speed;
     bird[birdnum].timer := ftime + respondt + pickupt;
     bird[birdnum].loc := p1;
     bird[birdnum].des := p2;
     i := trunc((min/convert)/10.0); if (i > 50) then i:=50;
{ ** if range is > than 500 this must change ! ** }
     bird[birdnum].dHist[i] := bird[birdnum].dHist[i]+1;
{ ** dHist[1] = # flts wth dst in [10,20)   ** }
     bird[birdnum].CallDist := bird[birdnum].CallDist + min;
  end  { of avail= 0 loop }
 end; { of with data }
end; { of choosbird }

Procedure Genaccident;
 var
  temp : extended;
begin
{     gotoxy(2,1);cleareol;write(' GENACCIDENT ');    }

with Data do
begin
 with nextaccident do
  begin
   jobtyp := 4;
   time := expo(nmaccid);
   temp := random;
   if temp < percin1 then
    repeat
     location.h := trunc(abs(random) * xlen2 + xlim2 );
     location.v := trunc(abs(random) * ylen2 + ylim2 );
    until (ptinrgn(location, hirgn))
   else
    begin
     location.h := trunc(abs(random) * xlen + xlim1 );
     location.v := trunc(abs(random) * ylen + ylim1 );
    end;

{ clean this up. }

   temp := abs(random) / 32768;
   if (temp < perachi) then
    priority := 3
   else
    priority := 2;
   end
end
end;

Procedure Choosdest;
 var
  i            : integer;
  dist, dist1 : extended;
begin
   {  gotoxy(2,1);cleareol;write(' CHOOSDEST ');    }

with Data do
begin
 dist := 1.0E30;

 for i := 1 to numhsp do
  begin
```

```
    dist1 := findist(loc, hosp[i].location) ;
    if ( dist1 < dist ) then
     if ( hosp[i].status >= priority)  then
      begin
       dist :=dist1;
       dest := hosp[i].location;
       destcntr := i;
      end
   end;
   distance := dist;
  end
  end;


 Procedure Gentransf;
  var
   i, temp2       : integer;
   temp1          : extended;
   found          : boolean;
 begin
 {  gotoxy(2,1);cleareol;write(' GENTRANSF ');    }
 { perctrma is the % of emerg which need transfer }
 { totemerg will be the total # of emerg at centers with priority < 3 }

 with Data do
 begin
  if (totemerg = 0) then
   nexttransfer.time := simtime
  else
   begin

    temp1 := perctrma * totemerg;
    temp2 := abs(random) mod (totemerg) + 1;
    nexttransfer.time := expo(temp1);
    nexttransfer.jobtyp := 3;
    i := 0;
    found := false;
    repeat
     i := i + 1;
     if ((hosp[i].status < 3) and (hosp[i].cumtot >= temp2)) then
      found := true;
    until (found = true);
    nexttransfer.location := hosp[i].location;
    temp1 := abs(random) / 32768.0;    { assume 25% are high priority }

    if (temp1 < pertranhi) then
     nexttransfer.priority := 3
    else
     nexttransfer.priority := 2;
   end
 end
 end;

 Procedure Fly;
  var
   distance, wtime, ftime        : extended;
   avail, priority               : integer;
   locc, destt                   : point;
   worthit                       : boolean;

 begin
{  gotoxy(2,1);cleareol;write(' FLY ');    }
 with Data do
 begin
  worthit := true;
  locc := transferq[1].location;
```

```
  priority := transferq[1].priority;
   if (transferq[1].jobtyp=3) then priority := 3; { kludge to insure that dest for a transfe
  choosdest(destt, locc, priority, distance);
  if       (distance < launchdist)
           then begin
                 worthit := false;  penpat(white);
                                    paintcircle(locc,1);    { *********** }
                                    penpat(black);
                 delq;
                end;
  if (worthit = true) then
   begin
     choosbird(locc, destt, distance, wtime, rtime, ftime, avail);
     if (avail = 0) then    { birdnum is global var identifying which bird }
      begin
        with bird[birdnum] do
        begin
         hosp[destcntr].rcvd := hosp[destcntr].rcvd + 1;
         nflts := nflts + 1;
         fltime := fltime + ftime;
         waitt := waitt + wtime + (clock - transferq[1].time);
         resct := resct + rtime + (clock - transferq[1].time);
         jobtyp := transferq[1].jobtyp;

         moveto(base.h,base.v);
         lineto( loc.h, loc.v);
         lineto( des.h, des.v);
         lineto( base.h, base.v);
        end; { of with }

      delq;
      end { of if ( avail = 0 )  }

     ELSE
      begin
       if (transferq[1].priority = 3) or (avail = 2) then
        begin
         if (priority = 3) then
                          begin
                          died   := died + 1;
                          moveto(locc.h-2,locc.v-2);
                          lineto(locc.h+2,locc.v+2);
                          end;

         if (avail = 2)     then
                          begin
                          toofar := toofar + 1;
                          moveto(locc.h-2,locc.v+2);
                           lineto(locc.h+2,locc.v-2);
                          end;
         delq;
         end;  { priority=3 or avai=2  }

     { ****** }
     { gotoxy(1,2);cleareol; write( ' tfar=  ', toofar:4,'  inque=   ',tcount,'  dist =  ',di
     { ****** }

      end  { of ELSE }
    end { of worthit = true }
 end
 end;


Procedure Simulate;
 begin
 with Data do
```

```
begin
 NMaccid   := 1000;
 Perachi   := 0.90;
 Perctrma  := 0.50;
 Pertranhi := 0.90;
 Percrpr   := 6.0/168.0;
 Durrpr    := 60;
 percbwthr := 4.0/720.0;
 durwthr   := 30;
 Pickupt   := 10;
 Respondt  := 10;

repeat
initialize;
genaccident;
gentransf;
nextweather := expo(nmwthr);
for ii := 1 to numbird do
 nextrepair[ii] := expo(nmrpr);
while clock < simtime do
 begin
  nextevent;
  case eventyp of
   0 :                            { bird is free }
    begin
     bird[birdnum].busy := false;
     case bird[birdnum].jobtyp of
      1 : begin  end;                   {end of repair}
      2 :                     {end of bad weather}
        begin
         for jj := 1 to numbird do
         bird[jj].busy := false; { this isn't quite right }
        end;
      3 :       ;  { end of accident }
      4 :       ;   { end of transfer }
     end;
    if (tcount > 0) then
    fly;
   end;

   1 :                            { repair time }
    begin
     nextrepair[birdnum] := expo(nmrpr) + durrpr;
     with bird[birdnum] do
      begin
       busy := true;
       jobtyp := 1;
       rptime := rptime + durrpr;
       timer := durrpr;
      end;
    end;
   2 :                        { bad weather }
    begin
     nextweather := expo(nmwthr) + durwthr;
     for jj := 1 to numbird do
      with bird[jj] do
       begin
        jobtyp := 2;
        timer := durwthr;
        busy := true;
       end;
    end;
   3 :                      { transfer }
    begin
     addq(nexttransfer);
     gentransf;
```

```
      fly;
     end;
    4 :                       { accident }
     begin
      addq(nextaccident);
      genaccident;
      fly;
     end;
   end;
  end;

  DisposeRgn(hirgn);
  gotoxy(1,2); cleareol; write(' Press return to see printout   '); readln;
  DoReport;
 until ( itemNo <> 4 ) ;
 { Kludge, from DoReport, itemNo = 4 means he wants another simulation }

 end;
end;

begin


END.  { OF UNIT }
```

```
Unit MacExtras( 128 );

{$O Birds:Units.F: Sim.Lib }
{$U-}

Interface
  Uses
      memtypes, QuickDraw, OSIntf, ToolIntf, PackIntf;

  CONST
     AppleID     = 1;
       AboutCmd  = 1;
     EditID      = 3;
       UndoCmd   = 1;
       {--------}
       CutCmd    =3;
       CopyCmd   =4;
       PasteCmd  =5;
       ClearCmd  =6;
     ScBarWidth  =15;
     MenuBarWidth = 18;
     MaxMenuCmds  = 31;

   TYPE
     MenuCmdSet = SET Of 1 .. MaxMenuCmds;
        string64    =  string[64];
        string255   = string[255];

   Var
     appleMenu     : MenuHandle;
     fileMenu      : MenuHandle;
     EditMenu      : MenuHandle;

     theEvent      : EventRecord;
     whichWindow   : WindowPtr;

Function InRange(n, min, max  : Integer ): Boolean;

Procedure Pause;

Procedure EnableMenu (mh : MenuHandle; Commands : MenuCmdSet );

Procedure DisableMenu( mh : MenuHandle; Commands : MenuCmdSet );

Procedure FixEditMenu( enableCommands : Boolean );

Procedure DragTheWindow( whichWindow : WindowPtr; startPoint : Point );

Procedure ResizeWindow( whichWindow : WindowPtr; startPoint : Point);

Procedure zoomInOut( whichWindow : WindowPtr; partCode : Integer );

Procedure CloseDAWindow;

Procedure GetPortSize( VAR width, height : integer );

Procedure CalcControlRects( whichWindow : WindowPtr;
     VAR hbarRect, vbarRect, gbRect : Rect );

Function TextHeight( wptr : WindowPtr ) : Integer;

Procedure CenterString( h,v,w : Integer; s : Str255 );

Procedure DisplayAboutBox;
```

```
Procedure DoAppleMenuCommands( cmdNumber : Integer );


IMPLEMENTATION



Function InRange;
{ true if min <= n <= max }
 begin
    InRange := ( min <= n ) and ( n <= max )
 end;

Procedure Pause;
 begin
 while button do systemtask;
 while not button do systemtask;
 flushEvents(KeyDownMask + autoKeyMask, 0 )
 end;

Procedure EnableMenu;
 var
   theCommand : 1 .. maxMenuCmds;
 Begin
  for thecommand := 1 to MaxMenuCmds Do
      If theCommand In commands
          then EnableItem( mh, theCommand )
 end;

Procedure DisableMenu;
 Var
  theCommand : 1 .. MaxMenuCmds;
 begin
  for theCommand := 1 to MaxMenuCmds Do
     If theCommand In commands
       then DisableItem( mh, theCommand )
 end;

Procedure FixEditMenu;
 Var
  editSet : MenuCmdSet;
 Begin
   editSet := [ UndoCmd, CutCmd, CopyCmd, PasteCmd, ClearCmd ];
   If enableCommands
      Then EnableMenu( editMenu, editSet )
      Else DisableMenu( editMenu, editSet )
 end;

Procedure DragTheWindow;
Var
  LimitRect : Rect;
Begin
  With screenBits.Bounds Do
     SetRect( limitRect, left + 4, top + 24, right - 4, Bottom - 4 );
  DragWindow( whichWindow, startPoint, LimitRect )
end;

Procedure ResizeWindow;
var
   size            : longint;
   width, height   : longint;
   limitRect       : Rect;
begin
  with screenBits.Bounds Do
```

```
   setRect( limitRect, 100,75,right,bottom-24 );
  size := GrowWindow( whichWindow, Startpoint, limitrect );
  if size <> 0 then
     with whichWindow^  DO
     begin
        EraseRect( PortRect );
        width  := LoWord( size );
        height := HiWord( size );
        SizeWindow( whichWindow, width, height, True );
        InvalRect( portRect )
     end  { if }
end;   { resizeWindow }


Procedure ZoomInOut;
var
 oldPort : GrafPtr;
begin
  Getport( oldPort );
  SetPort( whichWindow );
  EraseRect( whichWindow^.PortRect );
  zoomWindow( whichWindow, partCode, True );
  SetPort( oldPort )
end;


Procedure CloseDAWindow;
Var
  DANumber : Integer;
  DAWindow : WindowPeek;
Begin
  DAWindow  := WindowPeek(FrontWindow );
  DANumber := DAWindow^.windowKind;
  CloseDeskAcc( DANumber )
End;


Procedure GetPortSize;
Begin
 with theport^.portRect DO
 begin
  width := right - left;
  height := bottom - top
 end
end;

Procedure CalcControlRects;
Begin
  with whichWindow^.portRect DO
  begin
   gbRect.top := bottom - scBarWidth;
   gbRect.left := right - ScBarWidth;
   gbRect.bottom := Bottom;
   gbRect.right := right;
   hBarRect := gbRect;
   hBarRect.left := left;
   hBarRect.right := gbRect.left;
   vBarRect := gbRect;
   vBarRect.top := top;
   vBarRect.bottom := gbRect.top
  end
 end;


Function TextHeight;
 var
```

```
  fInfo : FontInfo;
  oldPort : GrafPtr;
 Begin
   GetPort( OldPort );
   SetPort( wPtr );
   GetFontInfo( fInfo );
   with fInfo DO
     TextHeight := ascent + descent + leading;
   SetPort( oldPort )
 end;

Procedure CenterString;
 begin
  w := w - StringWidth( s );
  if w < 0
    then w := 0;
  moveTo( h + ( w DIV 2 ), v );
  DrawString ( s )
 end;

 Procedure DisplayAboutBox;
 { requires 6-string STR# resource with IF = 1 containing strings
   to display in window as follows:
     STR#1    = Program name
     STR#2    = Author
     STR#3    = Version
     STR#4    = Copyright
     STR#5    = Address
     STR#6    = Phone Number      }

 Const
   strListID   = 1;    { rewource ID of Str# resource   }
 var
  oldPort     : GrafPtr;
  wp          : WindowPtr;
  wRec        : WindowRecord;
  wr          : Rect;
  i           : integer;

  messages : ARRAY[1 .. 6 ] of Str255;
 begin
   for i := 1 to 6 DO
     GetIndString( messages[i], strListID, i );
     wr := screenBits.bounds;
     InsetRect(wr,100,75);
     wp := NewWindow(@wRec, wr, '', TRUE, altDBoxProc, Pointer(-1), FALSE,0);
     if wp <> NIL then with wp^.portRect DO
     BEGIN
      GetPort( oldPort );
      SetPort ( wp );
      TextFont( systemFont );
      TextSize(12);
      CenterString( 0,30,right,messages[1] );
      TextFont ( geneva );
      TextSize(9);
      Centerstring(0,        60, right, messages[2] );
      Centerstring(0,        90, right, messages[3] );
      CenterString(0, bottom-60, right, messages[4] );
      Centerstring(0, bottom-40, right, messages[5] );
      CenterString(0, bottom-20, right, messages[6] );
      Pause;
      CloseWindow( wp );
      SetPort( oldPort )
     end;
   end;
```

```
Procedure DoAppleMenuCommands;
Var
 daName     : Str255;
 result     : integer;
begin
   if cmdNumber = AboutCmd
     then
       DisplayAboutBox
     else
       begin
       if FrontWindow = NIL
         then FixEditMenu( TRUE );
       GetItem( appleMenu, cmdNumber, daName );
       result := OpenDeskAcc( daName )
     end
end;


BEGIN
 { init toolbox managers }
 InitGraf( @thePort );
 InitFonts;
 InitWindows;
 InitMenus;
 TEInit;
 InitDialogs( NIL );
 InitCursor;
 FlushEvents( everyEvent, 0 )
END.
```

```
UNIT BirdGlobals( 135 );

{$O Birds:Units.F: Sim.Lib }

INTERFACE

USES
  Memtypes, QuickDraw, OSIntf,ToolIntf;

 CONST
        ok      = 1;
        cancel  = 2;
        pie     = 3.141593;
        nbds    = 20;
        minyear = 525600;
        hsp     = 20;



 type
      string64     =  string[64];
      string255    =  string[255];
      Ivect        =  array[0..50] of integer;


  helic = record
    base, loc, des : point;
    busy    : boolean;
    jobtyp  : integer; {  1=repair, 2=weather, 3=transfer,4=accid }
    timer   : extended;
    speed   : extended;
    range   : extended;
    nflts   : integer; { total number of flights }
    fltime  : extended; { total flight time }
    rptime  : extended; { total repair time }
    waitt   : extended; {total time spent waiting for bird }
    resct    : extended; {total time with patient on board }
    dHist    : Ivect;    { histogram of distances to calls }
    Calldist : extended;
   end;

  center = record
    status       : integer;
    emrgperday   : integer;
    cumtot       : integer;  { for assigning prob of transfer  }
    numinest     : integer;  { number of helicopters at this center }
    location     : point;
    fleet        : array[1..nbds] of integer;
    rcvd         : integer;
    name         : string255;

   end;

 Info = record
  MapName                                                            : String255;
  numbird,  rgncnt                                                   : integer;
  totemerg, numhsp                                                   : integer;
  percbwthr, durwthr, durrpr, perachi, pertranhi                     : extended;
  scale, avwait, avresc, convert, perctrma, percin, percrpr          : extended;
  nmwthr, nmrpr                                                      : extended;
  launchdist, nmaccid, respondt, pickupt                             : extended;
  xlim2,xlen2,ylim2,ylen2                                            : extended;
  bird                         : array[1..nbds] of helic;
```

```
 hosp                          : array[1..hsp] of center;
 Rgnpts                        : array[1..40] of point;
end;


var

 Data               : Info;  { this is record with all global variables  }
 datafile           : file of info;

 wRec                                      : windowRecord;
 wPtr                                      : WindowPtr;
 ph                                        : PicHandle;
 Hndl                                      : Handle;
 MapFrame, TxtFrame                        : rect;
 ans                                       : char;
 inName, outName                           : String255;
 readdat                                   : Boolean;

 IMPLEMENTATION
  begin

  END.  { OF UNIT }
```

```
Birds:Birds.RSRC

*---------------------------------------------*


TYPE STR#                      ;;
   ,1 (32)                     ;;
6                              ;;
Rotor Craft Rescue             ;;
    Vers 1.0                   ;;
(c) 1988 R. W. Smith           ;;
University of Puerto Rico      ;;
Mayaguez, Puerto Rico 00708    ;;
---------------------------    ;;



Type MENU
   ,1
\14
  About Picture
   (-


Type MENU
   ,2
File
   New /N
   (Close
   Open /O
   Save /S
   Quit /Q

*---------------------------------------------*

Type MENU
   ,3
Edit
   (Undo /z
   (-
   (Cut /X
   (Copy /C
   (Paste /V
   (Clear

*---------------------------------------------*

Type MENU
   ,1000
Sim
   (Run
   (-


*---------------------------------------------*

Type MENU
   ,1003
Map
   (NewMAP
   (OpenMap
   (FromScrap
```

```
   (SaveMap


*-------------------------------------------*




Type WIND
   ,1000 (32)
Map Window
20 2 340 510
Visible GoAway
4
0


*-------------------------------------------*


Type WIND
   ,1001 (32)
Rpt2
133 2 340 510
Visible GoAway
2
0


*-------------------------------------------*

Type DLOG
   ,1000
ask
20 10 330 500
Visible NoGoAway
1
0
1000


Type DITL
   ,1000 (32)              ;; Item list
22

BtnItem Enabled
280 350 305 450
OK

BtnItem Enabled
280 150 305 250
Cancel


EditText Enabled                 ;; 3-12 Field entry areas
30 445 48 485

EditText Enabled
50 445 68 485

EditText Enabled
```

```
70 445 88 485

EditText Enabled
90 445 108 485

EditText Enabled
110 445 128 485

EditText Enabled
130 445 148 485

EditText Enabled
150 445 168 485

EditText Enabled
170 445 188 485


EditText Enabled
190 445 208 485

EditText Enabled
210 445 228 485


StatText Disabled                    ;; 13-22 Field Name areas
30 10 50 440
Number of Accidents/Year requiring Rotorcraft

StatText Disabled
50 10 70 440
% of Accidents to Level 3 Trauma Center

StatText Disabled
70 10 90 440
% of Ambulance Accidents later needing air Transfer

StatText Disabled
90 10 110 440
% of Transfers Which must go to Level 3 Trauma Center

StatText Disabled
110 10 130 440
Hours/Week out of service due to maintenance

StatText Disabled
130 10 150 440
Average Duration of Maintenance [min]


StatText Disabled
150 10 170 440
Hours/Month out of service due to Weather

StatText Disabled
170 10 190 440
Average Duration of Weather Downtime [min]


StatText Disabled
190 10 210 440
Average time on ground to pick up patient [min]

StatText Disabled
210 10 230 440
Average time to respond to call [min]
```

```
*--------------------------------------------*

Type DLOG
   ,1010
Question?
22 10 45 490
Visible NoGoAway
1
0
1010


Type DITL
   ,1010 (32)                ;; Item list
4

BtnItem Enabled
280 350 310 450
OK

BtnItem Enabled
280 150 310 250
Cancel


StatText Enabled
4 10 20 435
^0


EditText Enabled
4 445 20 475


*--------------------------------------------*


Type DLOG
   ,1020
Rp1
26 10 130 500
Visible NoGoAway
1
0
1020


Type DITL
   ,1020 (32)                ;; Item list
17


BtnItem Enabled                 ;; # 1   Next center
85 272 100 360
Next Center


BtnItem Enabled                 ;; #2 Ok button
85 440 100 480
OK
```

```
BtnItem Enabled                ;; # 3   Print it
85 370 100 425
PRINT


BtnItem Enabled                ;; # 4   another simulation
65 370 80 460
Re-Simulate



EditText Enabled               ;; #5  # of flights   5- 10 Field entry areas
25 222 40 250


EditText Enabled               ;; #6 av dist
45 222 60 250


EditText Enabled               ;; #7 av wait
65 222 80 250


EditText Enabled               ;; #8 av time
85 222 100 250

EditText Enabled               ;; #9 out of range
25 450 40 478

EditText Enabled               ;; #10 unavailable
45 450 60 478



StatText Enabled               ;; #11  {x week simulation time }
1 2 20 218
^0

StatText Disabled
25 2 40 218
Number of Rotorcraft Flights

StatText Disabled
45 2 60 218
Avg Distance to Scene

StatText Disabled
65 2 80 218
Avg Wait for Rotorcraft arrival

StatText Disabled
85 2 100 218
Avg Total Transfer Time

StatText Disabled
25 275 40 440
Missed (out of Range)

StatText Disabled                             ;; # 17
45 275 60 440
Missed (Unavailable)
```

```
Type DLOG
   ,1030
HospQ
22 10 45 500
Visible NoGoAway
1
0
1030


Type DITL
   ,1030 (32)                ;; Item list
7

BtnItem Enabled              ;;  #1 ok
280 350 310 450
OK


EditText Enabled                  ;; #2  NAME    2- 4 Field entry areas
3 55 18 125


EditText Enabled                             ;; #3 status
3 295 18 325


EditText Enabled                             ;; #4 Num of Birds
3 460 18 485

StatText Disabled                            ;; #5
3 2 18 50
Name:

StatText Disabled                             ;; #6
3 140 18 290
Status (1,2,3)

StatText Disabled                             ;; #7
3 345 18 455
# of Rtrcraft



Type DLOG
   ,1040
BirdQ
22 10 45 500
Visible NoGoAway
1
0
1040


Type DITL
   ,1040 (32)                     ;; Item list
6

BtnItem Enabled                   ;;  #1 ok
280 350 310 450
OK
```

```
EditText Enabled                    ;; #2  speed    2- 3 Field entry areas
3 80 18 110


EditText Enabled                    ;; #3 range
3 203 18 233


StatText Enabled                     ;; #4  rotorcraft Number
3 300 18 450
^0

StatText Disabled                    ;; #5
3 2 18 77
Speed:

StatText Disabled                    ;; #6
3 125 18 200
Range:


* End
```

Ames Grant
IN-01-CR
149522
P-99

FINAL REPORT
September 1987 - July 1988

# THE ROLE OF AVIATION TECHNOLOGY IN THE CARIBBEAN BASIN:

NASA-Ames Cooperative Agreement No. NCC 2-496


Robert W. Smith
University of Puerto Rico
Mayaguez, Puerto Rico 00709


C-2