ITARS ROBUST DEMONSTRATION SYSTEM INTEGRATION

Mr. Ray Patrick
Mr. G. Mel Barney

Merit Technology Incorporated
Plano, Texas

Abstract

With the availability of Digital Terrain Elevation Data (DTED), mass storage systems such as the ITARS, have been developed for use in avionics applications requiring real-time access to large amounts of DTED data. Merit Technology has developed an avionics simulation that interfaces directly to DTED mass storage systems.

Under a subcontract with Hughes Aircraft Corporation, Merit Technology has developed this simulation system with a direct interface to the ITARS digital map. The system, known as the Robust Demonstration System (RDS) effectively demonstrates how ITARS digital terrain data could be used by aircraft of the future involving Terrain Following, Terrain Avoidance and SITAN avionics algorithms. This paper describes the ITARS/RDS system architecture, integration results, and areas of possible improvement.

INTRODUCTION

With the advent of Terrain Following radar, a new breed of navigation subsystems has emerged that requires accurate and detailed knowledge about surrounding terrain. Early TF systems relied entirely on elevation data measured by return signals form an onboard radar. Such systems allowed aircraft to fly safely at very low altitudes with little or no pilot intervention. The drawback of this method, however, was that a sizable radar signature was produced from the radiation emitted from the TF radar. Within the last ten years, large scale digital terrain elevation data (DTED) storage systems have become a reality and have the potential to greatly improve TF systems, as well as make feasible other avionic algorithms such as terrain avoidance, terrain navigation, and threat avoidance.

The ITARS is an airborne digital map system that stores, manages and displays large quantitates of cartographic and mission data including (but not limited to) terrain elevation data and cultural feature data. This data is maintained by the system and made available to various avionics subsystems.

The ITARS Robuts Demonstration System (RDS) provides a realistic set of avionics subsystems that exercise the ITARS's ability to manage and distribute terrain elevation, and cultural feature data. The RDS provides Navigation, Terrain Following, and Terrain Avoidance/Threat Avoidance subsystems along with an aircraft flight model, threat assessment model, RDS to ITARS communication, joystick flight controls, and a user-friendly scenario generation interface. The system is configured with commercially available components including two MicroVAXs and an IBM PC/AT.

ITARS/RDS System Overview

Functional description of ITARS

The ITARS is designed to interact with three primary avionic subsystems which require timely access to digital terrain and feature data. These subsystems are known as "users" and consist of a Navigation user, Terrain Following user, and Terrain Avoidance/Threat Avoidance user. Communication between the ITARS and these user subsystems is accomplished with messages sent over two communication paths. ITARS control, status and fault isolation messages travel along a MIL-STD 1553B data bus. Larger messages (typically output data received from the ITARS) travel along a High Speed Data Bus (HSDB). As currently implemented, the HSDB is an Ethernet data link.

As shown in Figure 1, terrain data is sent to these users in the form of Terrain Output Messages which contain elevation data points measured in meters above sea-level and are organized as North Up blocks of 16 by 16 points. For the Navigation and TF user, each block is 48 arcseconds on a side (resulting in a point resolution of 3 arcseconds). For the TA/ThA user, each block is 240 arcseconds on a side (resulting in a point resolution of 15 arcseconds). Each user receives an

CH2596-5/88/0000-0083 $1.00 © 1988 IEEE

Area Load of data which consists of a series of blocks organized in a two dimensional manner. This area load contains elevation data for an area surrounding the current aircraft position. The size of the area load, as well as the rate at which it is sent to each user is specified in the ITARS User Request Message. Feature data is sent to each user in the form of Feature Output Messages. The area for which features are supplied, is the area covered by the user's terrain area load.

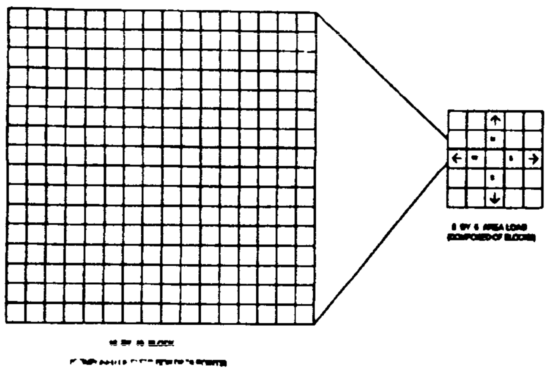

Figure 1. ITARS Terrain Output Data

Positional information is sent to the ITARS in the form of a Navigation/Aircraft State Vector Message. This message contains the position, as well as the current heading, pitch, and ground speed of the aircraft.

Threat features and waypoint information may be introduced into the ITARS with the ITARS Point Feature/Mission Data Message. This message, in conjunction with the Aircraft State Vector message, allows the pilot to designate a point on the ITARS display. Table 1 summarizes the major ITARS messages used by RDS.

Message	Type	Description
Aircraft State Vector	1553B	Provides positional information to the ITARS
User Request Message	1553B	Specifies which users are active and the size and rate of area loads
Point/Feature Message	1553B	Provide threat and waypoint information to ITARS (at any time)
Terrain Output Message	HSDB	Provides terrain data to the RDS
Feature Output Message	HSDB	Provides feature data to the RDS

Table 1. RDS/ITARS Messages

In addition to managing data, the ITARS can simultaneously generate two situation displays. These displays are full color raster views of data stored within the system, with a great deal of control possible over each image rendered. Typical formats for these displays include a perspective (out the window) view and a plan view with contour and shaded relief map options.

Functional description of the RDS

The RDS software simulates the user subsystems communicating with the ITARS and provides a scenario generation capability to define waypoints, threats, and means of flight control (autopilot or joystick). In addition, the user may specify the size and rate of area loads to be received from the ITARS. The system also includes a Route Planning algorithm which is executed if Terrain Avoidance/Threat Avoidance is enabled.

Once the scenario is constructed, waypoint and threat information is sent to the ITARS with an ITARS Point/Feature Mission Data Message. Selected user algorithms, as well as the size and desired update rates of designated area loads is sent to the ITARS with the ITARS User Request Message. An initial aircraft position (corresponding to the first defined waypoint) is sent to the ITARS with an ITARS Aircraft State Vector. After receiving these messages, the ITARS starts data transmission to the enabled user subsystems. This data is continually sent by the ITARS and is received by each user subsystem which maintains a double-buffer of received area loads. The TA/ThA area is handled differently in that it is sent on demand (i.e. when the User Request Message is sent to the ITARS) and contains data for the entire ITARS gaming area. The RDS uses this data to perform threat masking calculations as well as Terrain Avoidance/Threat Avoidance computations.
When the flight simulation is started, the RDS flight model simulates the movement of an A-7 aircraft through the set of designated waypoints. As each flight position is computed, an aircraft state vector message is constructed and sent to the ITARS. To maintain an acceptable update rate, the RDS interpolates between

flight model positions resulting in several aircraft state vectors for each flight model position derived.

The flight simulation is monitored with the RDS console. An aircraft symbol is displayed on the gaming region map and moves from waypoint to waypoint The display also maintains a separate window for monitoring ITARS to RDS message traffic.

Internal RDS organization

As shown in Figure 2, the user algorithms, the flight model, and the ITARS communication software are organized into separate processes coexist on two microVAXs. Each process is a self-contained program that is controlled by the main process (which also contains the flight model and route planner). Each process is highly independent of the others, with communication between each process being accomplished using Digital Equipment's DECNET software. The software is partitioned across both microVAXs according to memory, throughput and hardware interface requirements.

Figure 3 illustrates the hardware configuration of the RDS system and its relationship with the ITARS. The user interacts with the system using the IBM PC/AT keyboard and mouse. The PC monitor is a high resolution color display (1024 by 800 pixels) driven by a high resolution graphics controller (VMI-1024). During the simulation flight, the user may maneuver the aircraft using the joystick connected to the MicroVAX. In addition to the microVAX, a separate IBM PC/AT called

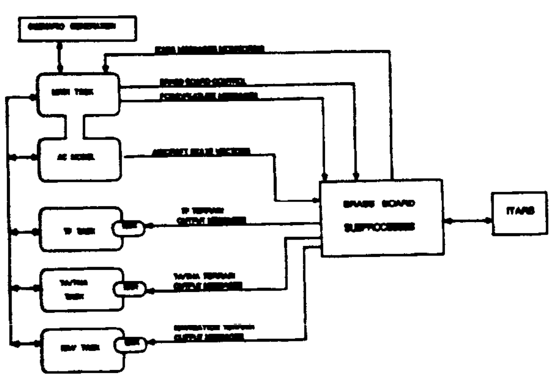

Figure 2. RDS Internal Organization

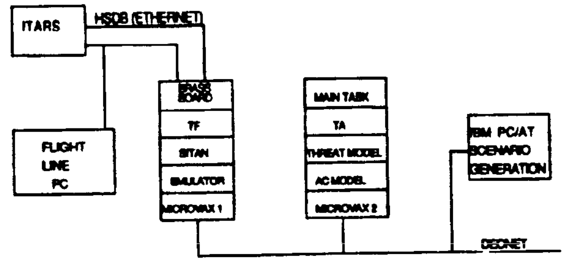

Figure 3. RDS Hardware Configuration

the flightline PC is also connected to the 1553. This PC sends ITARS Mode Control Messages and Point Feature Messages to the ITARS. Mode control messages control the ITARS mode of operation, allowing the user to alter the view on either of the ITARS displays.

All ITARS communication is handled by one program called the Brass Board process. This program provides both a 1553B and Ethernet communication link with the ITARS. Messages sent to the ITARS from the RDS are first routed to the Brass Board process via DECNET and mailbox communication links (a form of interprocess communication). The Brass Board process examines the message header to determine which 1553 subaddress the message should be sent to. The message is then forwarded to the ITARS. The Brass Board subprocess acts as the Bus Master for the entire 1553 data bus, coordinating all message traffic including mode message and point/feature message traffic originating from the Hughes flightline PC.

The Brass Board process asychronously forwards 1553B messages to the ITARS while managing incoming data from the Ethernet link. When a message is received from the ethernet link, the system performs a validity check on the message and then parses the message header to determine which user subsystem should receive the message. The message is then forwarded to the correct user via DECNET or a mailbox communication link. Each time an area load of data is received by the Brass Board process, a message is sent to the RDS main task detailing the size and type of area load received. The main task forwards this message to the PC/AT software for display on the screen. This mechanism allows the operator to monitor incoming data from the ITARS.

Each use subsystem interfaces with the Brass Board process with a set of routines known as the ITARS Data Interface Routines (IDIR) (see Figure 2). These routines provide a high-level, device-independent communication layer between the user subsystems and the ITARS hardware. These routines properly construct ITARS messages from the values specified in their input arguments and send the messages to the Brass Board process. In addition, these routines asynchronously receive data from the Brass Board process, and double-buffer it for later use by the user subsystem.

To accelerate the integration effort, Merit Technology created a program know as the ITARS Emulator. This program coexisted on the microVAX with the other user subsystems and provided terrain elevation data (at a reduced resolution) in a message format defined in the ITARS ICD. The Emulator also received aircraft state vectors, point Feature messages and user request messages sent by the aircraft model and main RDS task. Like the Brass Board process, each user subsystem communicated with the ITARS Emulator through the IDIR software. The Emulator proved to be an invaluable tool for system debugging and led to a significant change in the integration plan.

The original integration plan defined the IDIR as the final layer between the user subsystems and the RDS/ITARS communication hardware. The plan called for rewriting the IDIR software to communicate directly with the ITARS. This was determined to be a significant effort because the original IDIR software used DECNET for all of its interhost communication. The RDS/ITARS IDIR software would also need functionality similar to DECNET. The idea arose to add Ethernet and 1553 interfaces to the ITARS Emulator and forward messages "through" the Emulator to the ITARS hardware, hence the Brass Board process was born. An early version of the Brass Board process was an ITARS Emulator with a 1553 interface forwarding messages to the ITARS. This configuration provided a needed incremental integration step and allowed the RDS to operate without an Ethernet link from the ITARS (terrain data was generated by the Emulator). Later versions eliminated the ITARS Emulation logic and included an Ethernet interface. The advantage of this layered approach preserved the DECNET IDIR interface, eliminating any code changes to the user subsystems. Indeed, the user subsystems are "unaware" of the source of their elevation data.

Results of system integration

During integration several key lessons were learned about the RDS. As the system neared completion, it became apparent that limitations existed in the area of software partitioning. Although it is possible to host each subsystem in any machine, the configuration shown in Figure 2 proved to be optimal due to the through put requirements of each algorithm. The TA/ThA algorithm combined with the flight model, route planner, and main RDS task consume all of the processing capability of the first microVAX (the route planner executes before the flight simulation and therefore does not contend with TA/ThA). Analysis has shown that the bulk of processing on this microVAX is credited to the TA/ThA algorithm. The TF, SITAN and communication software load the second microVAX to 88.0 percent of its capability. If the ITARS Emulator is used instead of the Brass Board process, the microVAX becomes fully loaded. Based upon these estimates, the RDS fully loads both microVAX systems. Full real-time execution speed is achievable in a TF only configuration, however, with the addition of a third microVAX, full real-time execution speed could be achieved in a full-up configuration.

A larger than expected reduction in system throughput was noticed when the ITARS message monitoring mechanism was integrated into the system. This can be explained as follows: the main RDS task maintains a queue structure which collects arriving monitor messages from the Brass Board process. Each cycle through the simulation, this queue is emptied as each message is uploaded to the PC/AT. A time penalty is paid for this process resulting in a reduced overall hertz rate. A better idea would be to send a monitor message specifying number of area loads per unit period of time (i.e. 6 area loads/second). Data from this message would be used to render a gauge-type instrument on the PC/AT screen (bar chart).

Areas of Improvement

The system seems to perform as expected, however, minor modifications to the overall system architecture would improve system performance. One of these modifications would be to create a separate process for the aircraft state vector interpolation software. This software currently reside within the flight model and is interrupt driven (using a timer interrupt). Incorporation of the flightline PC software into the RDS would allow greater ITARS control from the RDS PC/AT. The ITARS system software could be improved by eliminating duplicate consecutive area loads. Currently, the ITARS sends area loads at a rate specified in the ITARS User Request Message. At the fastest rate (1 hz) several consecutive area loads will have identical data in them because the aircraft has not crossed a 16 by 16 block boundary. This can also occur if the aircraft is in a tight turn. This data unnecessarily crowds the bandwidth of the ethernet and requires some additional processing time on the receiving microVAX. Currently, the Brass Board process detects duplicate consecutive area loads and discards them.

Conclusion

The ITARS Robust Demonstration System has proven to be more than a testbed for the ITARS. With its innovative user algorithms, elegant software partitioning, and clever communication software, the system provides a near real-time avionics environment for virtually any digital map storage system. In addition, the RDS can easily interface with other avionic subsystems, providing an incremental integration path from the lab environment to the aircraft environment. A high degree of independence exists between each of the user subsystems and brassboard interface, thus minimizing the difficulty of porting the software to various hardware configurations. Incorporating the user subsystems within the ITARS hardware would provide a truly self-contained, airborne TF/TA/ThA digital map navigation system.

Acknowledgments

The author wishes to thank the many people involved in the development and integration of the RDS software. To Richard McKenzie, who developed the flight model and also responsible for the overall structure of the system. To Dan Adams who wrote and improved the Route Planner and many thanks to Mark Newburn, whose thoroughness and assistance made the project a success. Thanks to Chuck Lau and Barry LeBoeuf for their administrative support. Thanks to Mark Walden and Ron Young at Hughes Aircraft, for their help and patience during system integration. To Mr. Don Small, program manager of the ITARS Program and Lt. Gordon Hengst of AFWAL/AAAS-3 who provided excellent guidance for the project.