(2)

NAVAL POSTGRADUATE SCHOOL
Monterey, California

UNITED STATES NAVY · NAVAL POSTGRADUATE SCHOOL ·

THESIS

DESIGN OF AN INTERACTIVE SATELLITE
COMMUNICATIONS SYSTEM
ANALYSIS PROGRAM

by

Charles Craig Howard

June 1990

Thesis Advisor:	Donald v. Z. Wadsworth
Co-Advisor	George Knott

Unclassified

Security Classification of this page

REPORT DOCUMENTATION PAGE

1a Report Security Classification Unclassified

1b Restrictive Markings

2a Security Classification Authority

3 Distribution Availability of Report
Approved for public release; distribution is unlimited.

2b Declassification/Downgrading Schedule

4 Performing Organization Report Number(s)

5 Monitoring Organization Report Number(s)

6a Name of Performing Organization
Naval Postgraduate School

6b Office Symbol *(If Applicable)* 39

7a Name of Monitoring Organization
Naval Postgraduate School

6c Address *(city, state, and ZIP code)*
Monterey, CA 93943-5000

7b Address *(city, state, and ZIP code)*
Monterey, CA 93943-5000

8a Name of Funding/Sponsoring Organization

8b Office Symbol *(If Applicable)*

9 Procurement Instrument Identification Number

8c Address *(city, state, and ZIP code)*

10 Source of Funding Numbers

Program Element Number	Project No	Task No	Work Unit Accession No

11 Title *(Include Security Classification)* DESIGN OF AN INTERACTIVE SATELLITE COMMUNICATIONS SYSTEM ANALYSIS PROGRAM

12 Personal Author(s) Charles C. Howard

13a Type of Report
Master's Thesis

13b Time Covered
From To

14 Date of Report *(year, month,day)*
June 1990

15 Page Count
77

16 Supplementary Notation The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17 Cosati Codes

Field	Group	Subgroup

18 Subject Terms *(continue on reverse if necessary and identify by block number)*
Communications Link Analysis; Geosynchronous Satellite; Orbital Parameters.

19 Abstract *(continue on reverse if necessary and identify by block number*

This thesis addresses the design of an interactive satellite communications system analysis program. The program provides the capability to analyze/design a system comprised of two earth terminals and one or two geosynchronous satellites. The principal goal is to simplify the analysis/design process via a graphically-oriented, menu-driven computer program. The program leads the user methodically through the process and provides feedback that enables the user to visualize the elements of the system and their role relative to the other system components. Hypertext concepts are employed in an object-oriented programming environment to achieve the graphics orientation. The success of the program validates the use of innovative software tools to design programs that can enhance user understanding and increase productivity.

20 Distribution/Availability of Abstract
[X] unclassified/unlimited [ ] same as report [ ] DTIC users

21 Abstract Security Classification
Unclassified

22a Name of Responsible Individual
Donald v. Z. Wadsworth

22b Telephone *(Include Area code)*
(408) 646-2115

22c Office Symbol
EC/Wd

DD FORM 1473, 84 MAR

83 APR edition may be used until exhausted

All other editions are obsolete

security classification of this page
Unclassified

Approved for public release; distribution is unlimited.

Design of an Interactive Satellite Communications System Analysis Program

by

Charles C. Howard
Major, United States Army
B.S., United States Military Academy, 1977

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL

June 1990

Author: Charles C Howard
Charles C. Howard

Approved By: D. v. Z. Wadsworth
D. v. Z. Wadsworth, Thesis Advisor

G. Knott
G. Knott, Co-Advisor

John P. Powers
John P. Powers, Chairman, Department of Electrical and Computer Engineering

ABSTRACT

This thesis addresses the design of an interactive satellite communications system analysis program. The program provides the capability to analyze/design a system comprised of two earth terminals and one or two geosynchronous satellites. The principal goal is to simplify the analysis/design process via a graphically-oriented, menu-driven computer program. The program leads the user methodically through the process and provides feedback that enables the user to visualize the elements of the system and their role relative to the other system components. Hypertext concepts are employed in an object-oriented programming environment to achieve the graphics orientation. The success of the program validates the use of innovative software tools to design programs that can enhance user understanding and increase productivity.

TABLE OF CONTENTS

LIST OF FIGURES

I. INTRODUCTION

The analysis and design of a modern satellite communications system is a complex process. The design engineer's job would be greatly simplified if he could model a system, vary its complexity and parameters, and observe the effects of each (or all) of these actions. For the designer the ability to *visualize* each element of a complex system and its role relative to the other system components can be valuable feedback. While other computer programs which perform satellite communications link analysis have been produced, most are essentially "number-crunchers," which provided little intuitive or visual explanation of the system or its inner workings. This program attempts to remedy that shortfall.

The goal of this thesis is to simplify the analysis and design process via a graphics-oriented, menu-driven computer program. The program provides the capability to analyze/design a system comprised of two earth terminals and a geosynchronous satellite. If required, a second geosynchronous satellite can be added to provide an intersatellite link.

The program requires a standard Macintosh with a minimum of one megabyte of random access memory (RAM) and HyperCard version 1.2 or later. The system should be running under the Finder; the program will not run properly under MultiFinder unless the machine has at least two megabytes of RAM.

II. INTERACTIVE PROGRAM DESIGN

A. GENERAL

The program is designed with several underlying philosophies, each of which enhances the user-friendliness of the program. First, maximum use of graphics and visual effects is made to direct the user's attention to the action to be taken. Second, visual feedback is provided to the user to indicate his progress through the program. Third, explanations of potentially unfamiliar terms and typical ranges of values for the input parameters are provided. Fourth, the opportunity to obtain hard copy output from each of the main parts of the program is included.

The analysis of a satellite communications system is divided into two main parts. The first part addresses the determination of the geography-dependent parameters and requires the entry of the locations of both earth terminals and the satellite(s). Outputs are the terminal and orbital parameters and the geography-dependent rain attenuation data. The second part addresses the communications link analysis and design. Its outputs are a picture of the system with key system data displayed and a set of uplink and downlink, and, if necessary, intersatellite link budget tables.

Upon completing an analysis or design the user has the opportunity to change selected key parameters and re-run the program to observe how the final results are affected. The user also has the ability to examine the equations used by the program to perform its calculations.

The principal constraint on the program is the requirement to keep the program size under 200 kilobytes. This constraint manifested itself in limiting the number of maps and other graphics-intensive screens that could be included. A detailed explanation of this limitation is provided at Appendix B, III.

B. MAIN MENU

The first screen the user sees is the "Main Menu" and consists of three buttons as shown in Figure 1.

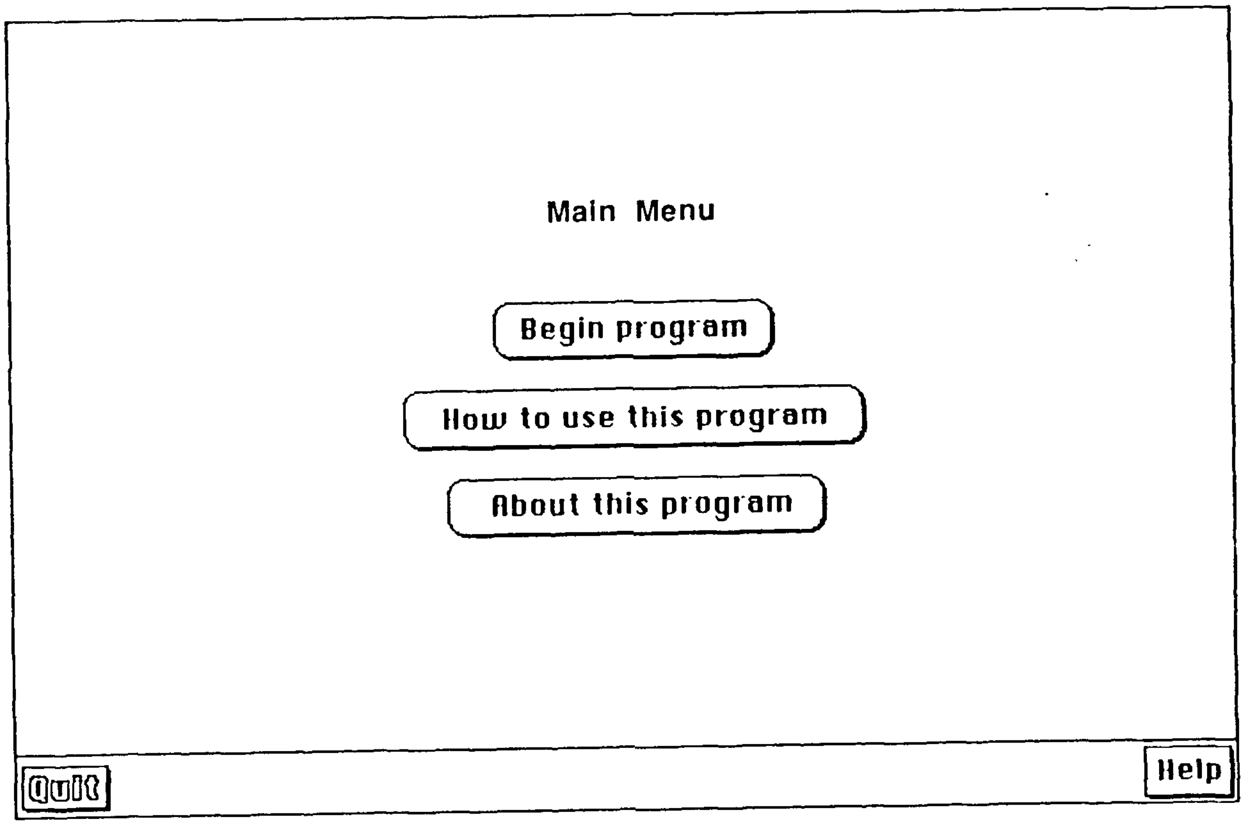

Figure 1. Main Menu

The first button launches the first part of the program. The second button, "How to use this program," provides a short tutorial on using the Macintosh interface, i.e., the mouse, buttons, and scrolling fields. The third button provides background information on the program design.

This screen is designed to immediately acquaint the user with the concept of pointing and clicking the mouse to initiate actions. If the user does not click on one of the buttons within approximately 30 seconds, a "hand with pointing finger" icon appears, moves to the second button, and clicks on it, revealing a screen that explains how to use the program. This animation sequence is designed to provide an example of how pointing and clicking initiates actions. The user is then guided through pointing and clicking to return to the main menu.

C. PART 1: THE GEOGRAPHY-DEPENDENT PARAMETERS

1. Program Procedure

The first part of the program is designed to graphically lead the user through the establishment of a satellite communications system. The user first builds a viable uplink, then, if necessary, an intersatellite link, and finally, the downlink.

Clicking on the first button in the main menu, "Begin program," launches the user into the first part of the program and displays the screen shown in Figure 2.

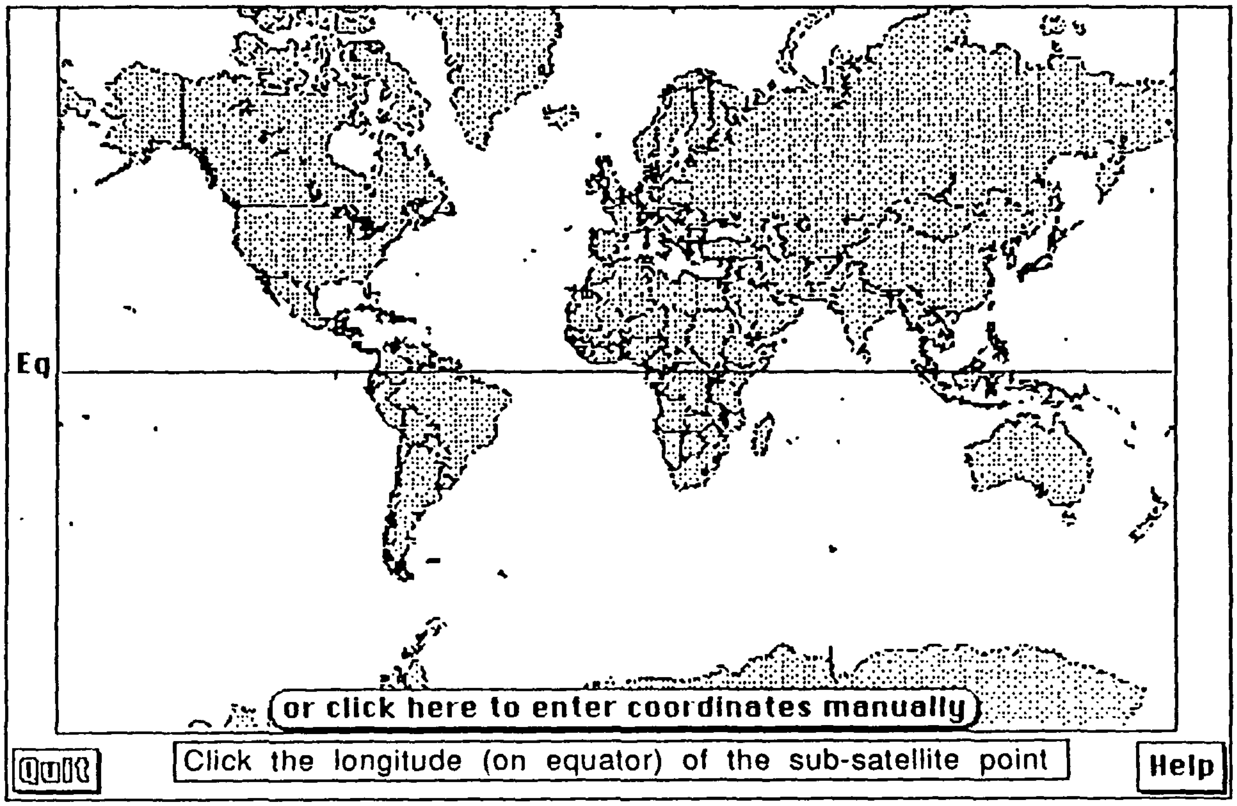

Figure 2. World Map [Ref. 1]

A key factor in establishing a satellite communications system is the selection of locations for the uplink and downlink earth terminals and the satellite(s). To bring an intuitive feel to this process, a graphics-oriented selection procedure is used. The user is presented with a conformal mercator world map centered at 0° longitude and displaying an equator line. Overlays of grid lines of varying resolution, i.e., every 10° or every 30°, were tried and rejected because of the degradation in quality of the map's graphics on the nine inch Macintosh screen.

The user is prompted first to select the longitude of the geosynchronous satellite; the latitude is nominally 0° by definition for any geosynchronous satellite. The selection of location can be made either by clicking the mouse on the map or, to obtain greater accuracy, by clicking on the button titled "or click here to enter the coordinates manually", which results in the displaying of the screen shown in Figure 3.

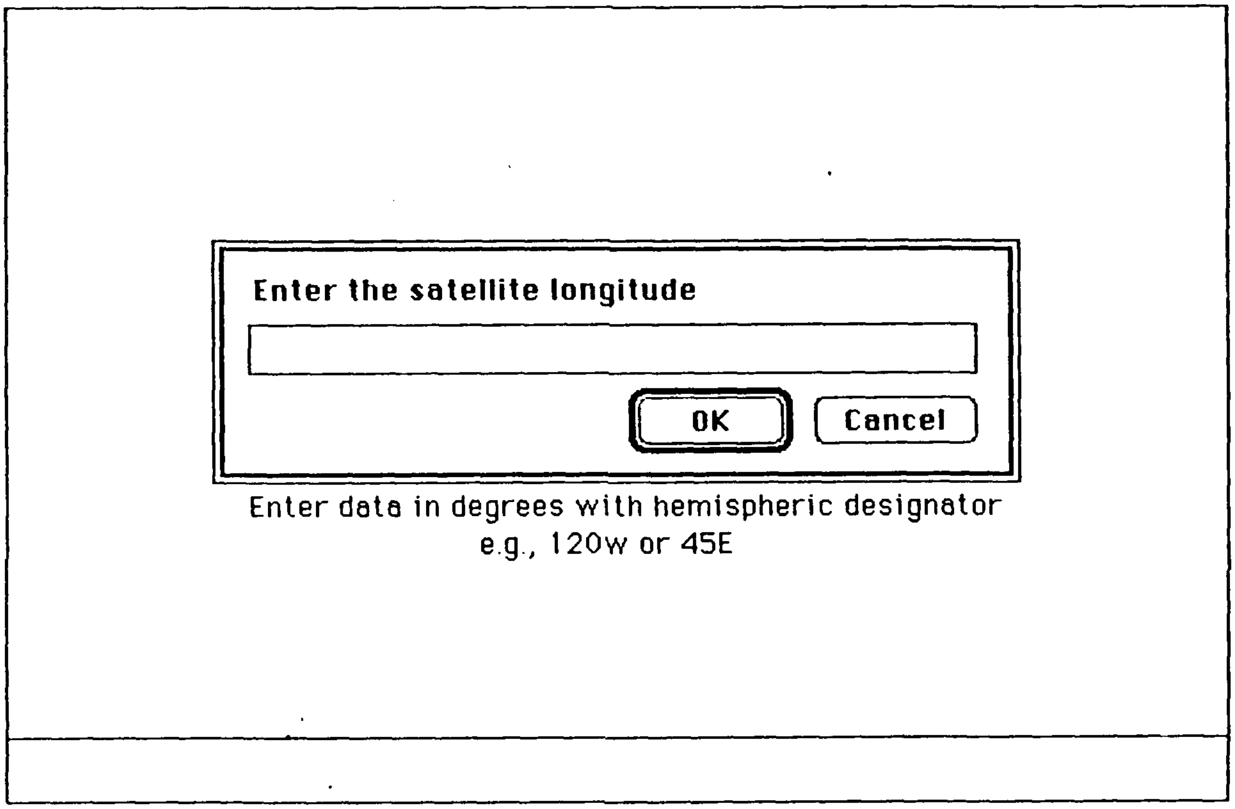

Figure 3. Manual Entry of Coordinates

Once the user has entered the satellite location, he is queried for the uplink and downlink earth terminal antenna elevation angles as shown in Figure 4.

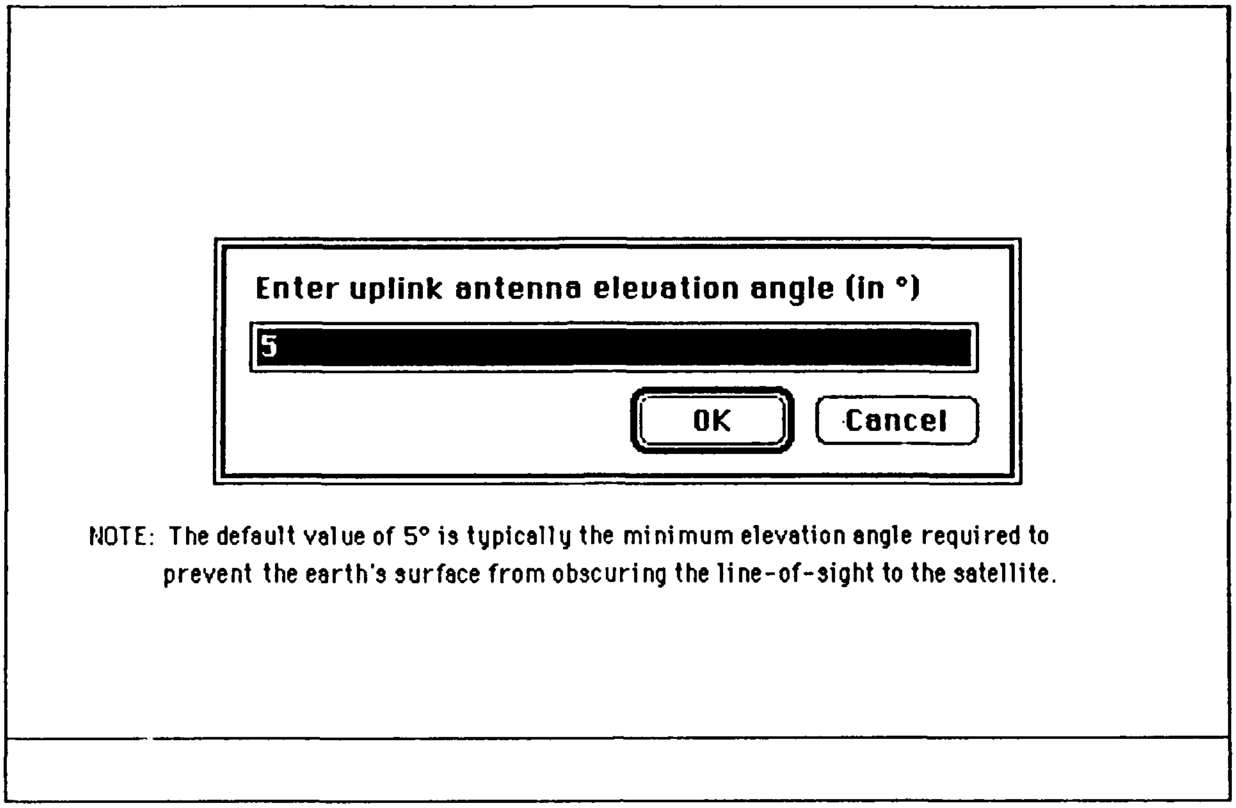

Figure 4. Query for Antenna Elevation Angle

The minimum elevation angle to preclude the earth's horizon from obscuring the path from an earth terminal antenna to a satellite is typically assumed to be five degrees. The query includes a five degree default value and an explanatory note to the user. The program requires the minimum elevation angles at this point, since the next step is to draw a satellite footprint on the world map. The area enclosed by the satellite's oval footprint is the area within which a terminal with the minimum specified elevation angle will be visible by the satellite. An explanation of this fact is displayed at the top of the screen as shown in Figure 5.

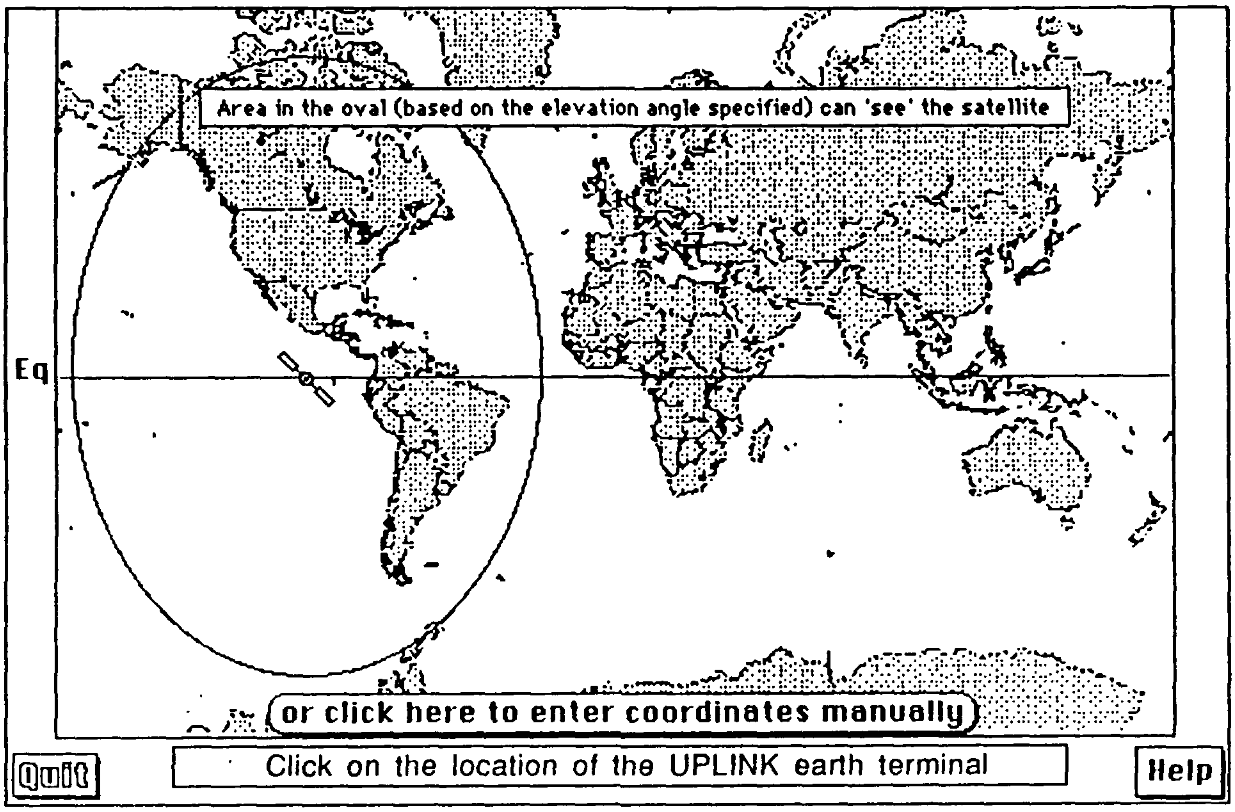

Figure 5. World Map with satellite footprint

Once the satellite footprint has been drawn, with the satellite icon appearing at the center of the footprint on the equator, the user is queried for the location of the uplink earth terminal. The user can determine immediately whether or not the choice of location for the earth terminal is visible from the satellite. The user can select the location by clicking on the map or entering the data manually. If the user clicks outside the footprint or manually enters a set of coordinates which places the terminal outside of the footprint, the program displays the screen shown in Figure 6.

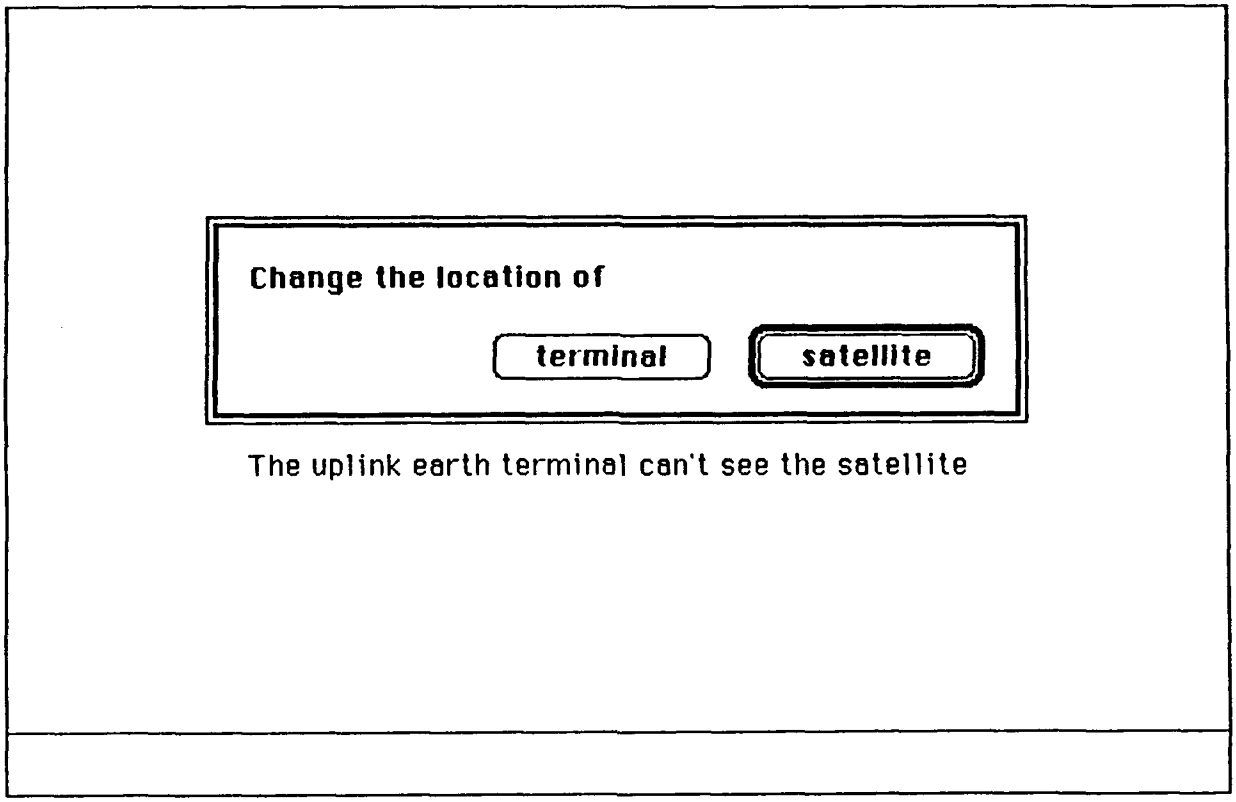

Figure 6. Satellite or Terminal location change options

The user now has the options of moving the location of the uplink earth terminal or the satellite. If the user chooses to move the satellite, the original screen of the world map, as in Figure 2, is displayed. Next, the user is queried to enter the satellite location as the process is begun anew. If he chooses to move the uplink terminal, the map with the satellite footprint, as shown in Figure 5, is displayed. Next, he is queried for the new uplink terminal location.

If the user clicks on an area outside the borders of the world map or manually enters a longitude greater than 180° or a latitude greater than 81.3°, the program displays a screen indicating an invalid entry has been made, as shown in

Figure 7. The screen specifies which item is invalid and returns the user to the appropriate screen with the instruction to enter valid data.

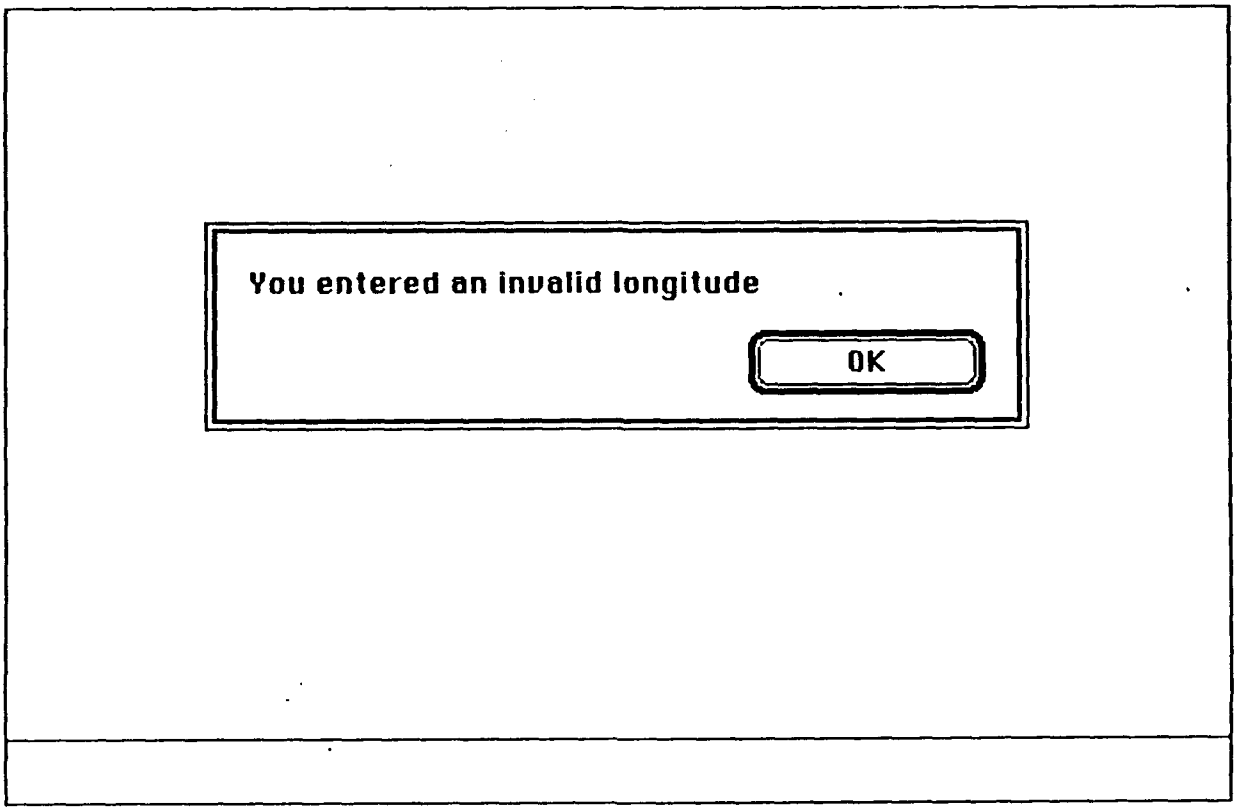

Figure 7. Invalid entry of data

Once mutually-visible locations have been selected for the uplink terminal and the satellite, these are indicated on the display screen. The display screen includes the satellite, its footprint, and the uplink terminal icon at its specified location, as shown in Figure 8.

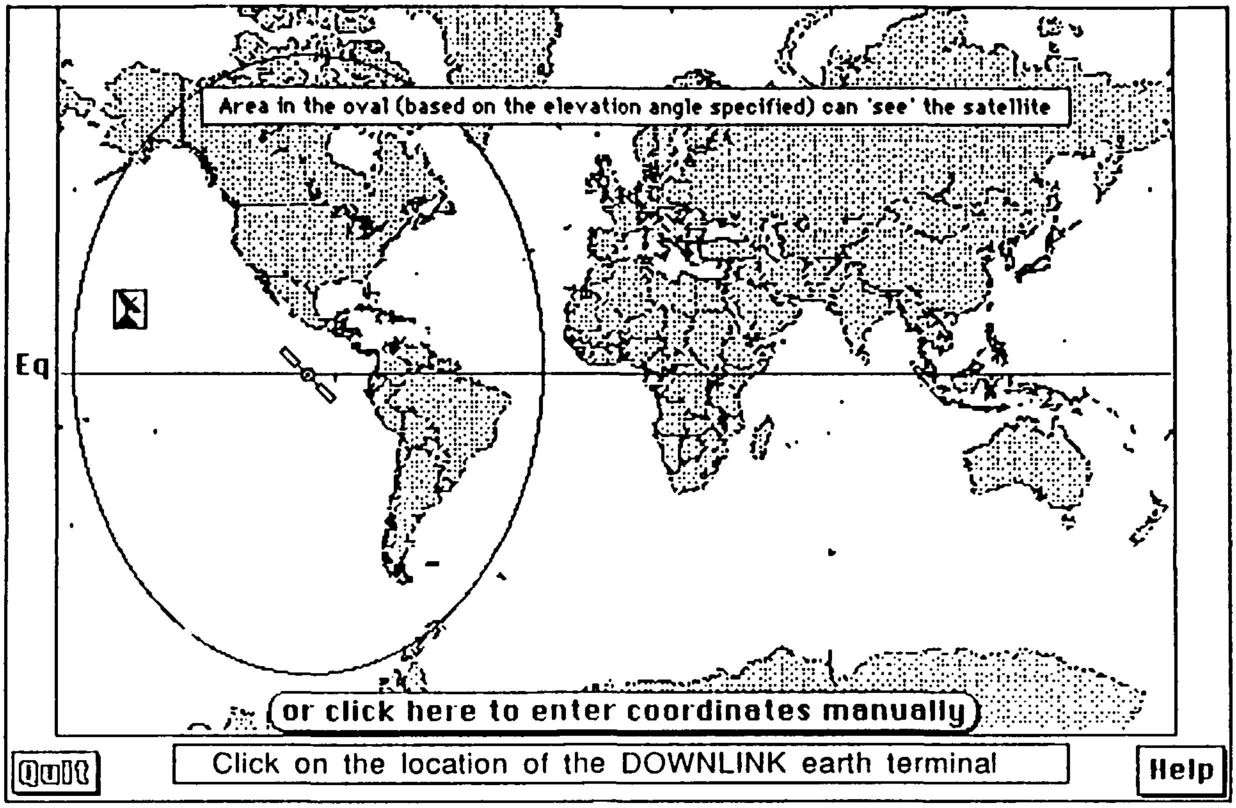

Figure 8. World Map with satellite, footprint, and uplink terminal

The user is now queried for the location of the downlink earth terminal. The selection procedure is the same as for the uplink terminal, except that if the downlink terminal location chosen is outside of the footprint, the screen shown in Figure 9 is displayed.

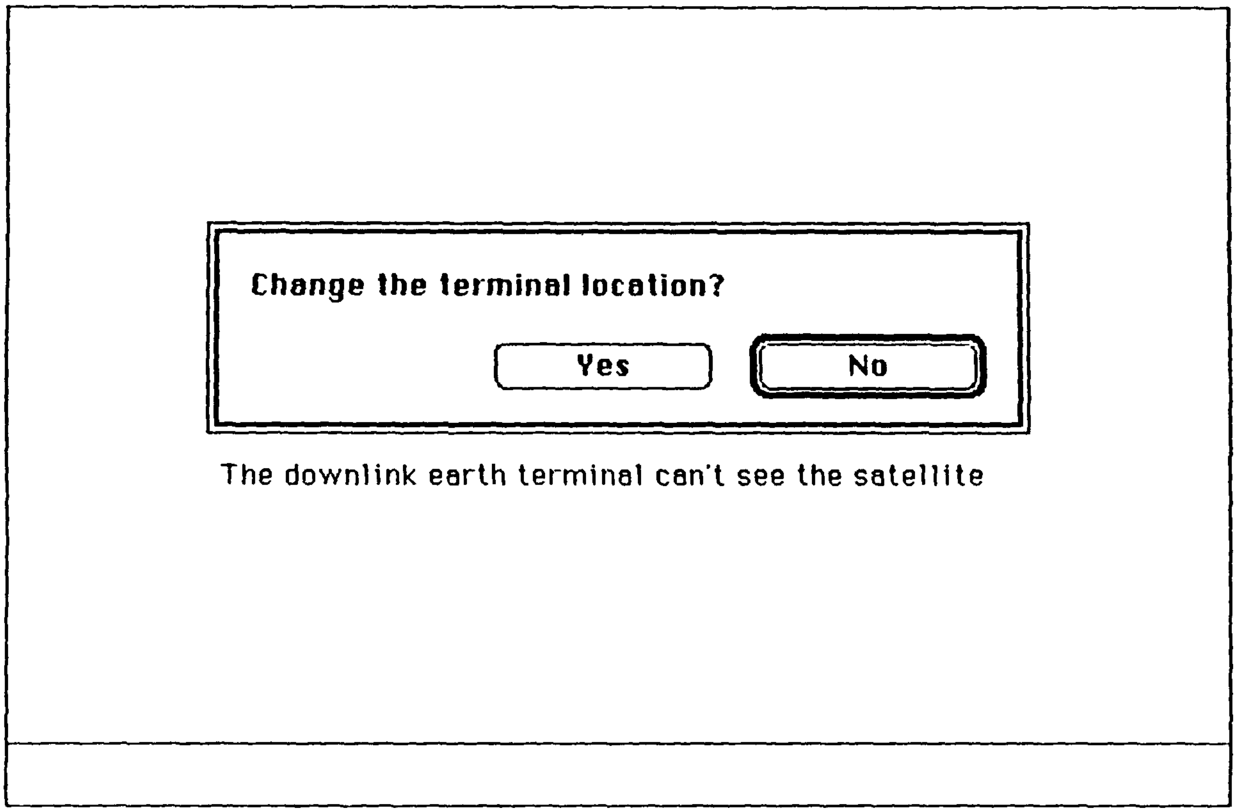

Figure 9. Downlink terminal location change

Since a valid uplink has been established, the user is not permitted to change either the location of the uplink terminal or the satellite. Instead, he has the options of moving the downlink terminal or adding a second satellite. If he chooses not to move the downlink terminal, the screen shown in Figure 10 is displayed, instructing that he must add a second satellite.

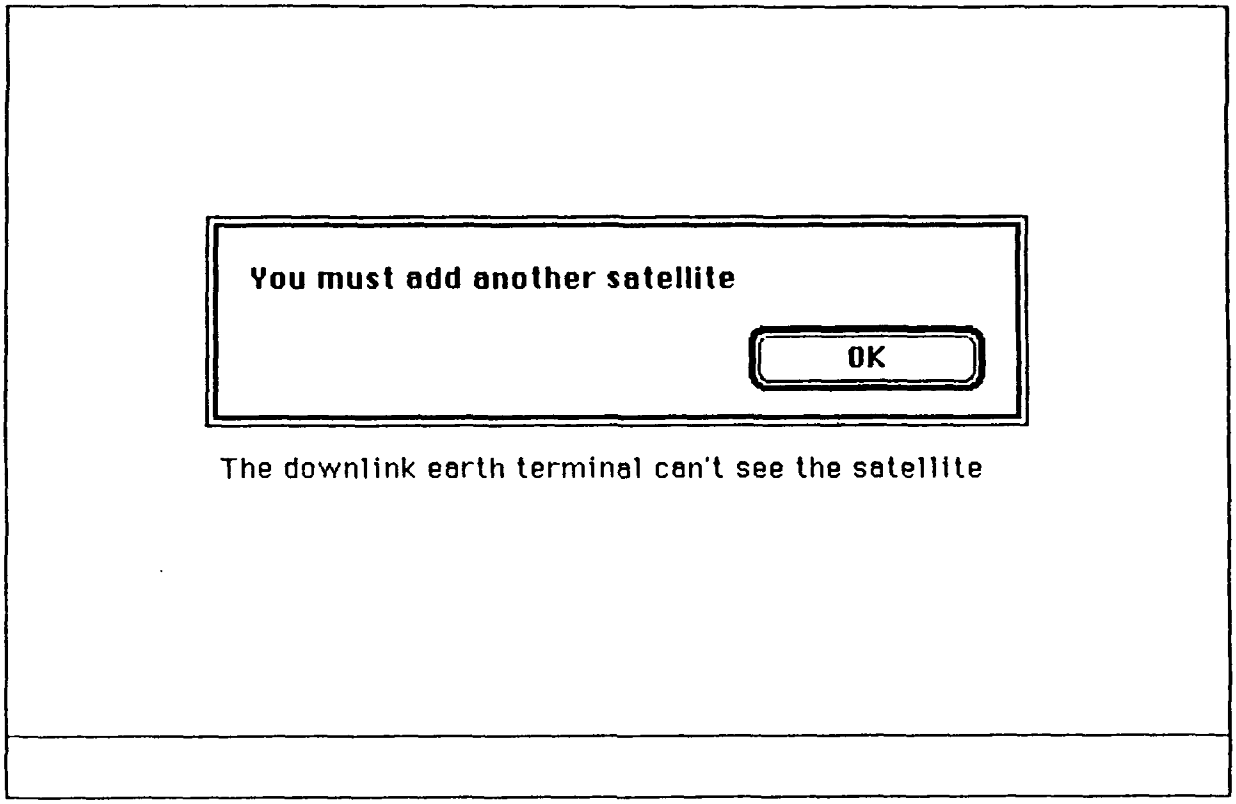

Figure 10. Requirement to add another satellite

When the user clicks on the "OK" button, he is returned to the world map and queried for the location of the second satellite, as shown in Figure 11.

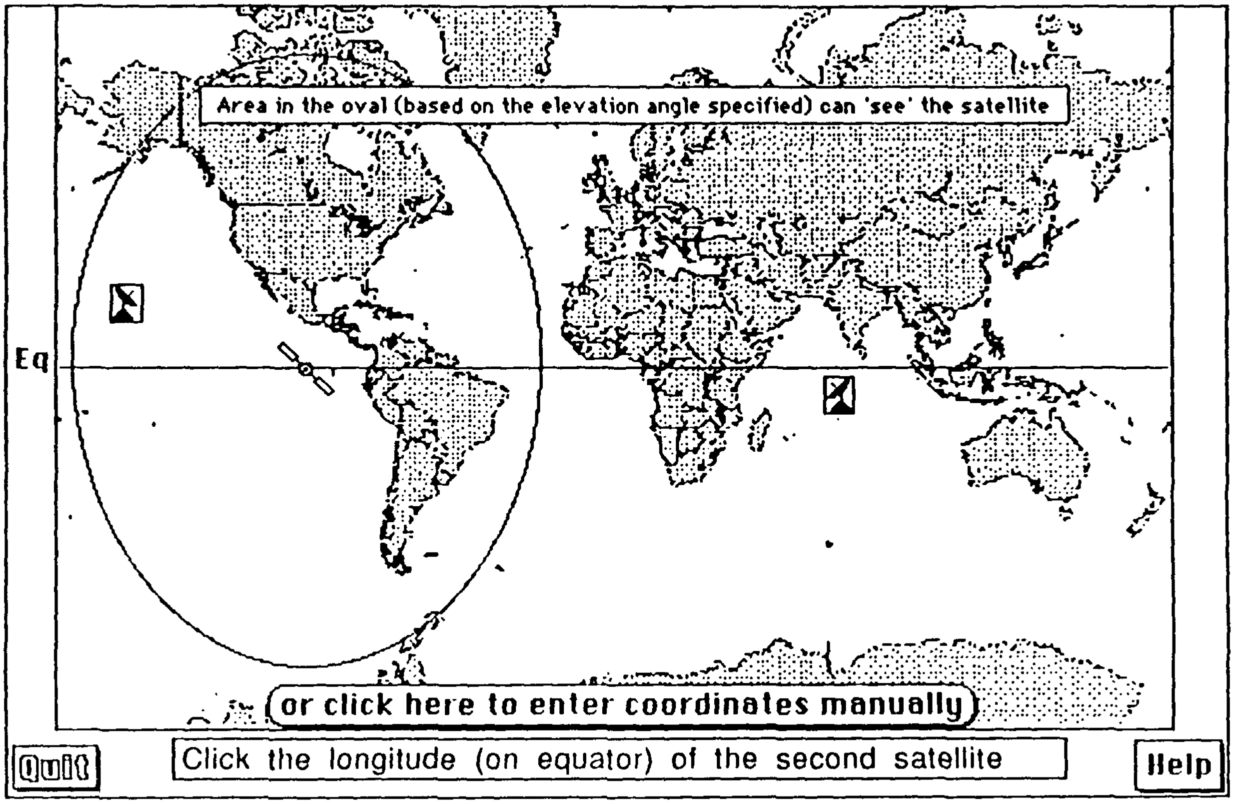

Figure 11. Query for location of second satellite

If the location chosen for the second satellite is greater than 162.6° in longitudinal separation from the first satellite, the two satellites will not be mutually visible. In this case the screen shown at Figure 12 is displayed and the user is instructed to select another location for the second satellite.

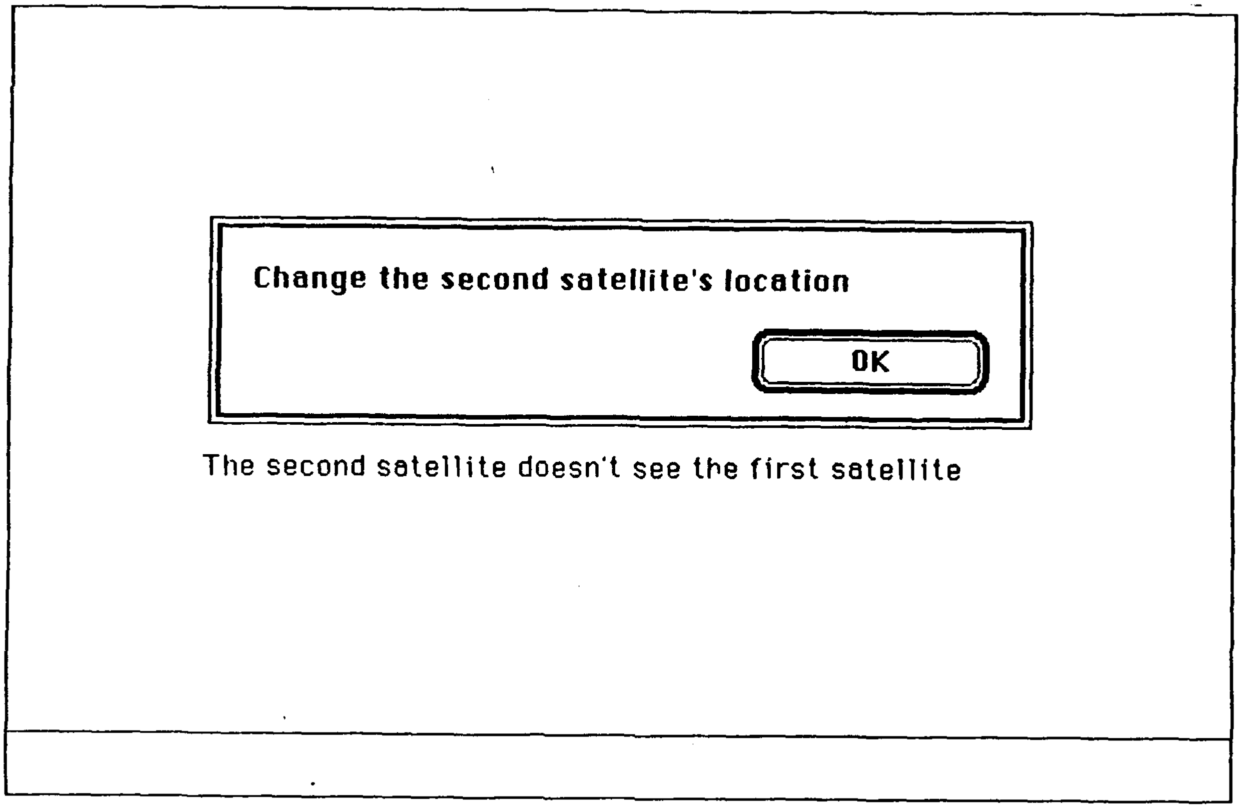

Figure 12. Instruction to change location of second satellite

Once the second satellite is placed in a location so that it is visible to the first satellite, a second satellite footprint is drawn with the satellite icon in the center, and the user is queried for the location of the downlink terminal.

Once the downlink terminal has been located within the footprint of the second satellite, the system is complete and the screen in Figure 13 is displayed.

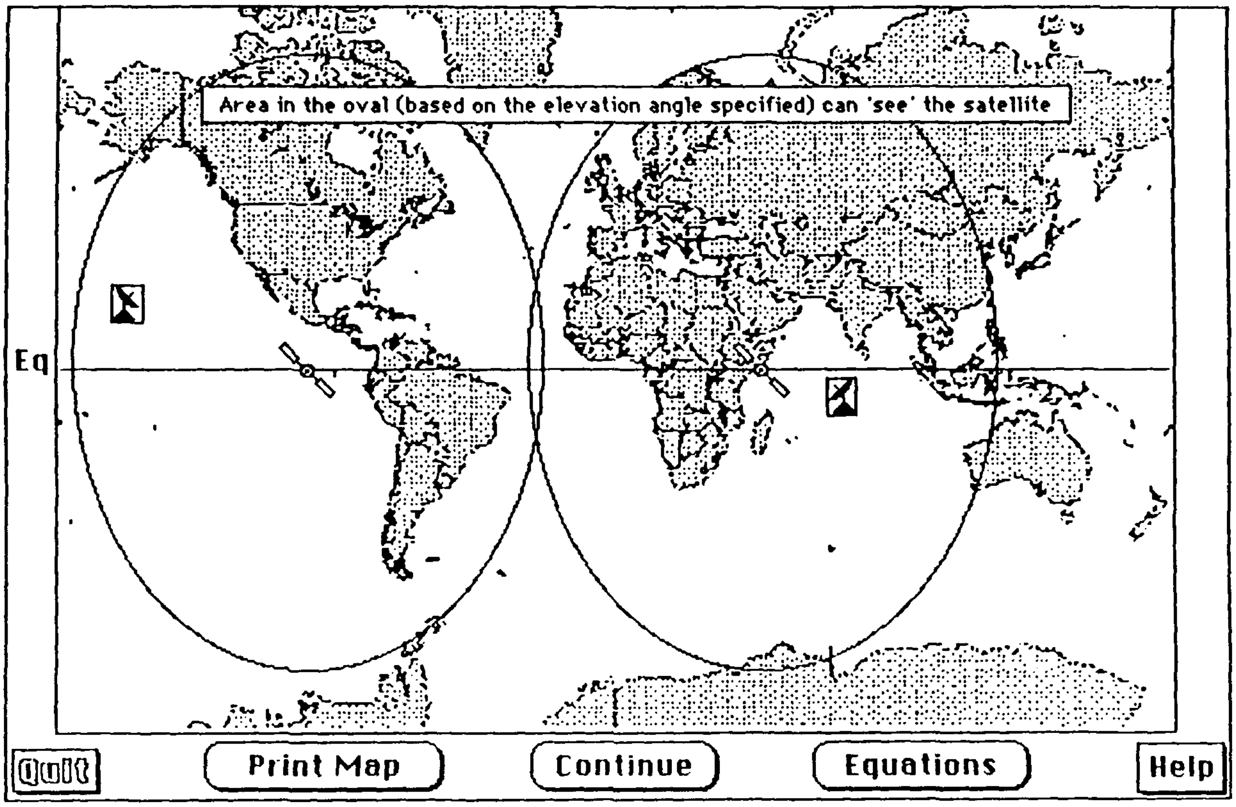

Figure 13. Completed System on World Map

The user now has three options. He can get a hard copy of the map with all data displayed by clicking on the "Print Screen" button. He can examine the equations used to perform the calculations by clicking on the "Equations" button. Lastly, he can begin the second part of the program, the communications link analysis, by clicking the "Continue" button.

The sequence of selection of the geographic parameters is deliberate; another sequence would require more screens or more buttons and/or fields, either of which would require more memory.

2. Program Equations

a. Basic orbital parameters

When the user clicks on a location on the world map, the program converts the cursor's location on the screen at the time of the click into a longitude, latitude, and a rain climate region from the Crane global rain model [Ref. 3:p. 157]. The program then calculates the azimuth angle, coverage angle, elevation angle, nadir angle, and slant range between the earth terminal and the satellite using Equations 2.1 through 2.6 from Reference 2:

$$\text{azimuth angle magnitude} = \tan^{-1}\left[\frac{\tan(\text{longitude}_{\text{diff}})}{\sin(\text{latitude}_{\text{es}})}\right] \tag{2.1}$$

$$\beta = \cos(\text{latitude}_{\text{es}})\cos(\text{longitude}_{\text{diff}}) \tag{2.2}$$

$$\text{coverage angle} = \beta_0 = \cos^{-1}(\beta) \tag{2.3}$$

$$\text{elevation angle} = [illegible] = \tan^{-1}\left[\frac{\cos(\beta_0)-0.15126}{\sin(\beta_0)}\right] \tag{2.4}$$

$$\text{nadir angle} = \alpha_0 = \sin^{-1}[0.15126\cos(q_e)] \tag{2.5}$$

$$\text{slant range} = 23192\sqrt{3.3811-\cos(\beta_0)} \tag{2.6}$$

where latitude_{es} is the earth station latitude and longitude_{diff} is the longitudinal difference between the earth station and the satellite.

Depending upon the location of the earth station with respect to the satellite, the azimuth angle is determined by the following relationships from Reference 3.

In the *northern* hemisphere, if the earth station is west of the satellite, the azimuth angle = 180° - (azimuth angle magnitude). If the earth station is east of the satellite, the azimuth angle = 180° + (azimuth angle magnitude).

In the *southern* hemisphere, if the earth station is west of the satellite, the azimuth angle = (azimuth angle magnitude). If the earth station is east of the satellite, the azimuth angle = 360° - (azimuth angle magnitude).

If $\text{latitude}_{es} > 81.3°$ or $\text{longitude}_{diff} > 81.3°$ or $\beta < 0.151$, then the satellite is obscured by the earth's surface and is not visible [Ref. 2:p. 222].

b. Intersatellite crosslink parameters

If the user has designed the system with two satellites, the determination of their separation distance and whether or not they are mutually visible must be made.

The crosslink range is calculated using equation 2.7:

$$\text{range (km)} = 84328.4 \sin\left(\frac{\theta}{2}\right) \tag{2.7}$$

where θ is the longitudinal separation in radians.

At geosynchronous altitude, the maximum longitudinal separation, q, which permits mutual visibility for two satellites is calculated in equation 2.8 as

$$q = 180 - 2\sin^{-1}\left(\frac{6378}{42164}\right) = 162.6° \tag{2.8}$$

If the longitudinal separation is larger, the earth's surface will block any intersatellite signal. In practice this longitudinal separation will be slightly less, so that the grazing ray lies above the sensible atmosphere. The determination of whether the two satellites are mutually visible proved to be a non-trivial problem. It is accomplished via the following algorithm:

1. If both satellites are in the same hemisphere, then if their longitudinal difference is less than 162.6°, they will be mutually visible.

2. If the satellites are in different hemispheres, then

   a. if the longitudes of both satellites are less than 90°, then if the sum of their longitudes is less than 162.6°, they will be mutually visible.

   b. if the longitudes of both are greater than 90°, then if 360° minus the sum of their longitudes is less than 162.6°, they will be mutually visible.

   c. if the longitude of one is greater than 90° and the longitude of the other is less than 90°, then subtract the larger longitude from 360°. Subtract the smaller longitude from the newly calculated longitude. If the difference is between 162.6° and 197.4°, the satellites will *not* be mutually visible.[Ref. 3]

D. PART 2: COMMUNICATIONS LINK ANALYSIS/DESIGN

1. Program Procedure

This part of the program is designed to lead the user methodically through the analysis/design of a satellite communications system link. As each major component of the system is addressed, the user is queried first for the derived parameter, e.g., the uplink effective isotropic radiated power (EIRP), or a receiver's gain for system noise temperature ratio (G/T). If the derived parameter is unknown, the program queries the user for the basic parameters or specifications needed to calculate the derived parameter.

The communications link analysis or design begins with the displaying of the screen in Figure 14.

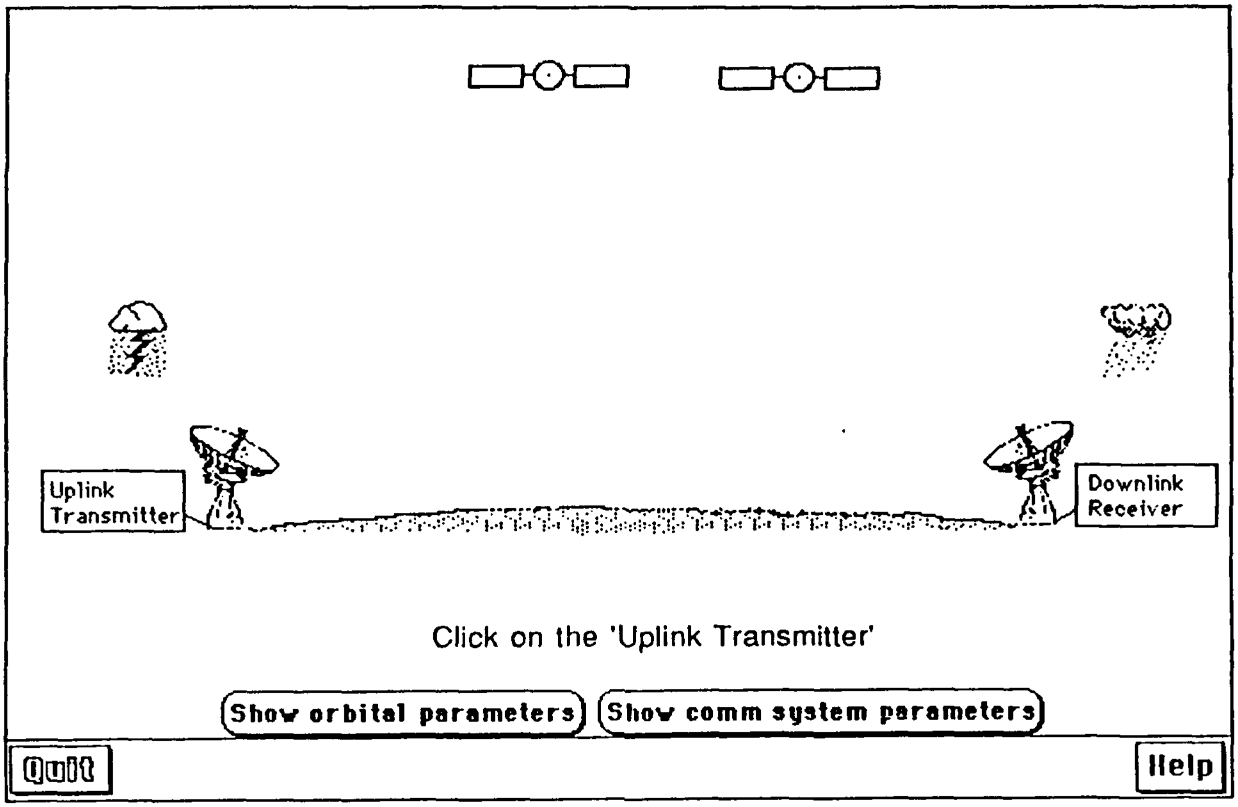

Figure 14. Picture of Satellite System

Prior to beginning the link analysis or at any time during the process, the user can display key data calculated from the geographic parameters entered in the first part of the program. When the user clicks on the "Show terminal parameters" button, the screen shown in Figure 15 is displayed.

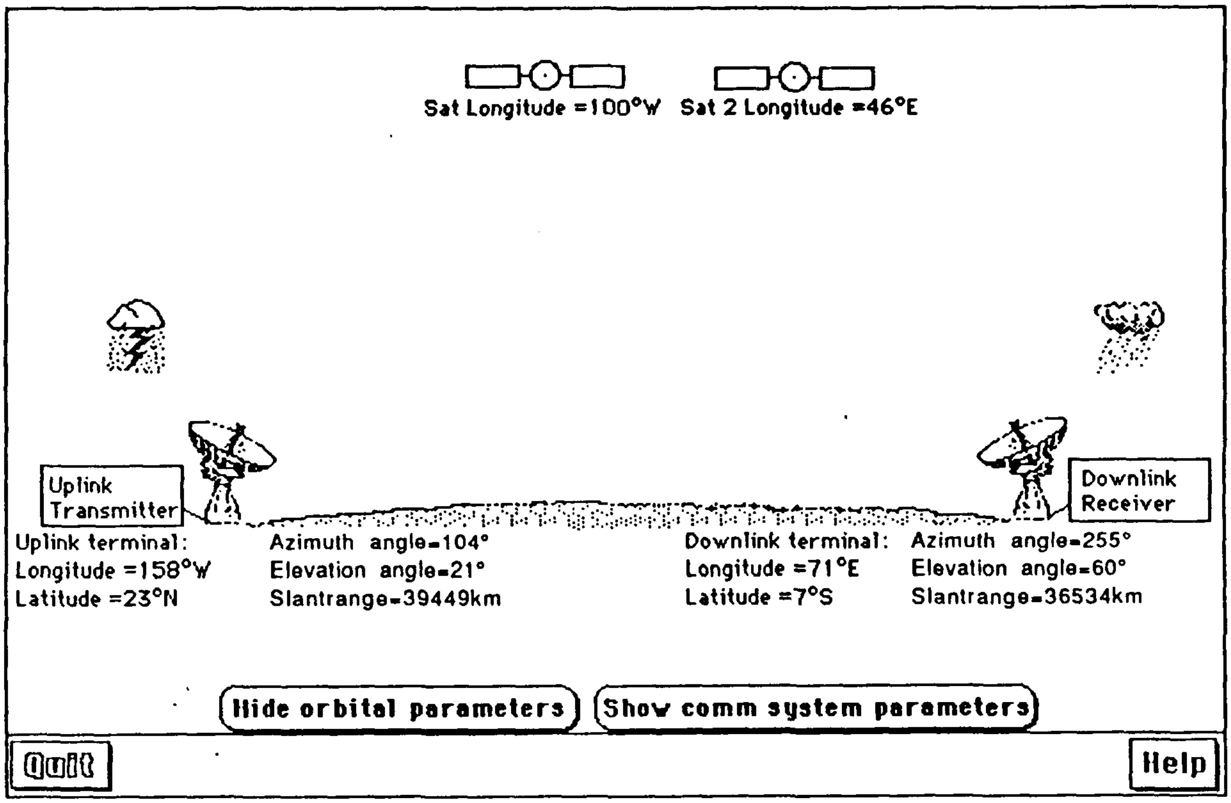

Figure 15. Picture of system with terminal parameters displayed

To return to the screen shown in Figure 14, the user need only click on the "Hide terminal parameters" button.

To initiate the link analysis, the instruction "Click on the Uplink Transmitter" is typed across the screen and reverse-highlighted, followed by the uplink transmitter's icon flashing once to attract the user's attention. After the user clicks on the uplink transmitter icon, the screen in Figure 16 is displayed.

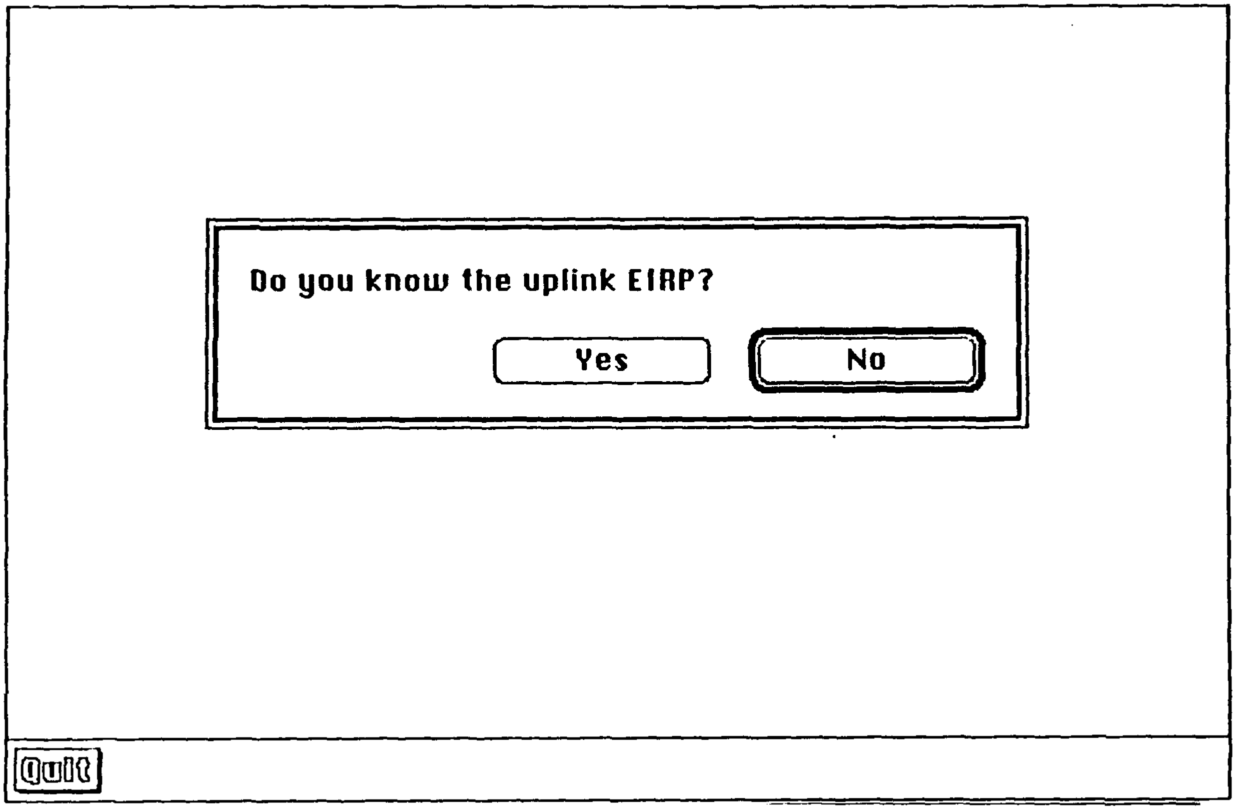

Figure 16. Query for a derived parameter

If the user knows the uplink EIRP, he enters this data and is queried for the system noise bandwidth using the format of the screen in Figure 16. The program then returns him to the screen in Figure 18, where the uplink transmitter icon is reverse-highlighted and an elongated z-shaped signal icon appears between the uplink antenna and the satellite. This display indicates the completion of the uplink transmitter and antenna portions of the system.

If the user does not know the uplink EIRP, he is queried first for the following parameters:

1) transmitter power in dBW or watts
2) reserve for end-of-life loss in dB

3) number of carriers

4) TWTA output back-off in dB

5) uplink frequency in GHz or wavelength in meters

6) feeder losses in dB

7) transmission line losses in dB

8) maintenance margin in dB

9) antenna pointing error loss in dB

10) pointing loss due to satellite motion in dB

These queries for basic parameters are presented sequentially in the format of Figure 16. The input data are used to calculate the transmitter power output at the input to the transmitting antenna. After all data have been entered, the user is returned to the screen shown in Figure 17.

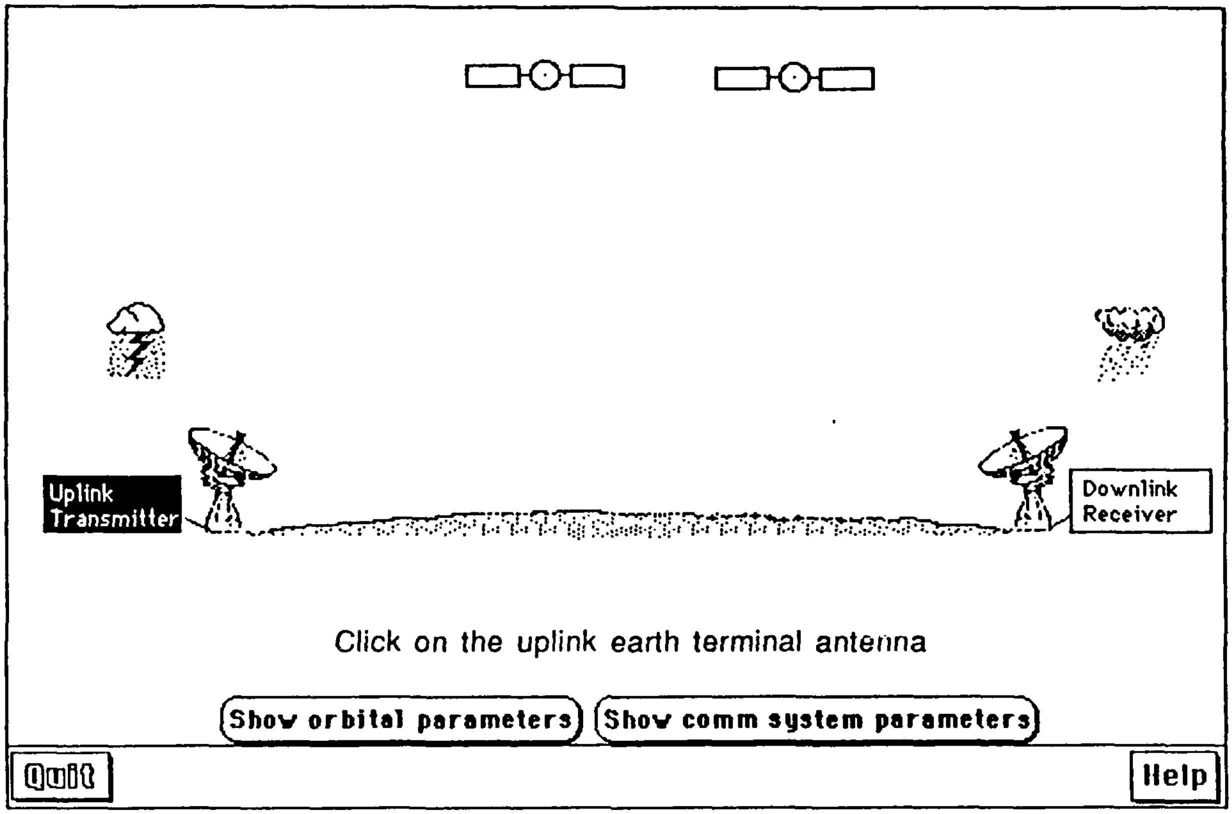

Figure 17. Completed Uplink Transmitter

The uplink transmitter icon is reverse-highlighted to indicate the completion of that component of the system. The instruction "Click on the Uplink Earth Terminal Antenna" is then typed across the screen, followed immediately by the flashing of the antenna icon. When the user clicks on the antenna icon, a screen with the format shown in Figure 16 queries him for the antenna gain in dBW or watts. If the user knows the antenna gain, then, after entering it, he is returned to the screen displayed in Figure 18.

If the user does not know the antenna gain, he is queried for the following parameters via the screen format of Figure 16:

1) antenna type, parabolic or other

2) area in meters-squared or diameter in meters

3) aperture efficiency

The program uses these data to calculate the antenna gain, then returns the user to the screen shown in Figure 18.

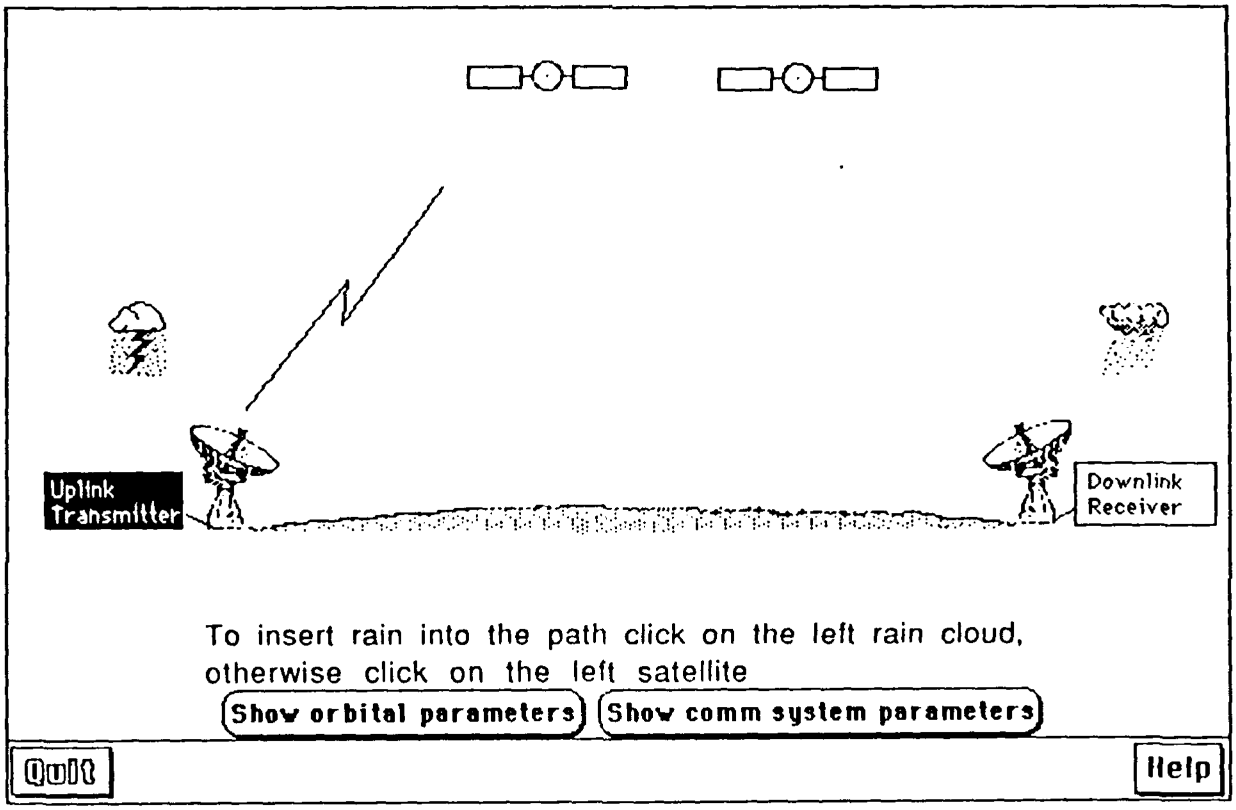

Figure 18. Completed Uplink EIRP

The instruction "To insert rain into the path click on the left rain cloud, otherwise click on the left satellite" is now typed across the screen, followed by a flashing of the left rain cloud icon. By clicking on the rain cloud located just to the left of the path between the uplink earth terminal and the satellite, the user can insert a rain-induced attenuation factor into the system. When the user clicks on the rain cloud, the cloud moves into the direct path between the earth terminal and the satellite. The program then queries the user for the earth terminal's altitude and, if not already entered, the uplink transmission frequency and rain climate region via a screen with the format shown in Figure 16.

After the earth terminal data is entered, the program displays the screen shown in Figure 19, from which the user selects one of the 12 surface point rain rates for his system. The "P%" represents the per cent of the year that the corresponding rate is exceeded.

The uplink earth terminal is in rain climate region: C

Select the surface point rain rate

P%	Rate (mm/h)	P%	Rate (mm/h)
○ 0.001	78	○ 0.1	7.2
○ 0.002	62	○ 0.2	4.8
○ 0.005	41	○ 0.5	2.7
○ 0.01	28	○ 1.0	1.8
○ 0.02	18	○ 2.0	1.1
○ 0.05	11	○ 5.0	0.5

P% = Per cent of the year that rate is exceeded

Help

Figure 19. Surface Point Rain Rates

After the user clicks on one of the 12 surface point rain rates, the program displays the screen shown in Figure 20. The rain cloud is in the path of and partially obscuring the elongated z-shaped signal icon. The instruction

"Click on the left satellite" is now typed onto the screen, followed immediately by the flashing of the satellite icon. These actions indicate to the user that he has successfully completed the uplink path from the earth to the satellite.

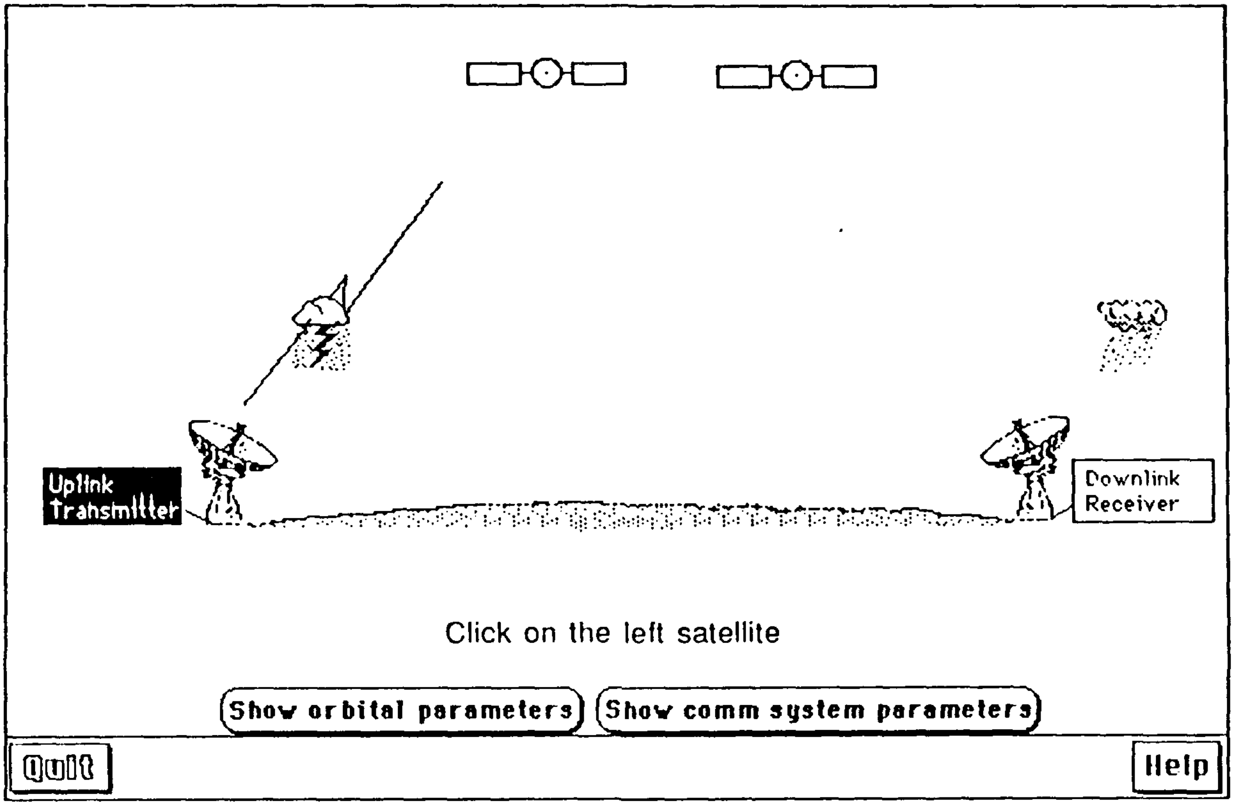

Figure 20. Completed uplink path with rain attenuation

When the user clicks on the satellite icon, the program displays the screen shown at Figure 21.

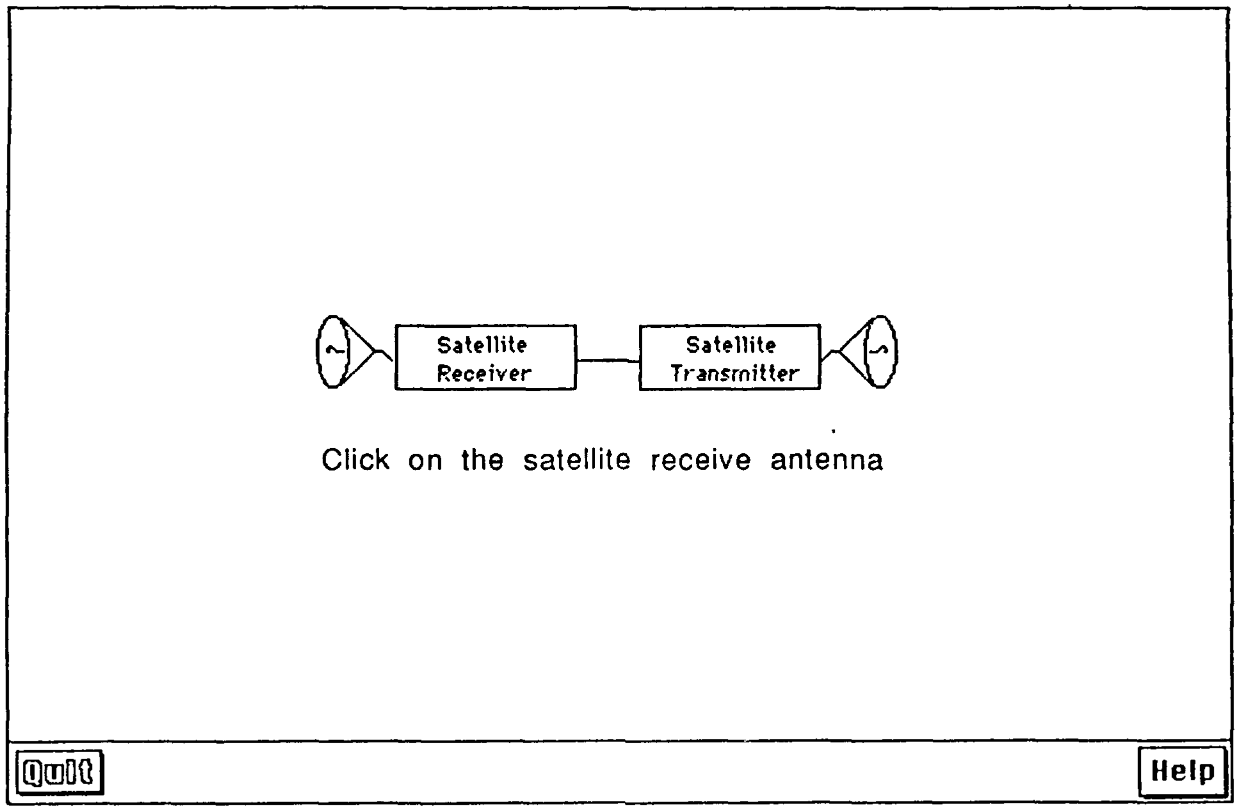

Figure 21. Satellite

The instruction "Click on the satellite receive antenna" is typed across the screen, followed by the flashing of the receive-side antenna icon. Upon clicking on the receive antenna icon, the user is queried for the satellite antenna G/T, the figure of merit, using the format of the screen in Figure 16. If this parameter is known, then, after its entry, the user is returned to the screen shown in Figure 21. If the G/T is not known, the program queries the user, via the screen format of Figure 16, for the following basic parameters needed to determine the antenna gain :

1) antenna type
2) area in meters-squared or diameter in meters

3) aperture efficiency

Once the antenna gain is calculated, the program displays the screen in Figure 21 and the instruction "Click on the Satellite Receiver," followed by the flashing of the receiver icon. When the user clicks on the receiver icon, the program requests the system noise temperature. If this parameter is known, it is subtracted from the antenna gain to obtain the G/T and the user is returned to the screen in Figure 21. If it is not known, the user is queried for the following basic parameters using the format in Figure 16:

1) antenna noise temperature in °K
2) waveguide loss in dBW or watts
3) low noise amplifier gain in dBW or watts
4) low noise amplifier noise temperature in °K
5) downconverter noise temperature in °K
6) ambient temperature in °K

After the system noise temperature is computed in dB, it is subtracted from the receive antenna gain to arrive at the G/T. The program then returns the user to the screen in Figure 21, where the instruction "Click on the Satellite Transmitter" is typed, followed by the flashing of the transmitter icon. When the user clicks on this icon, the program queries him for the satellite transmitter EIRP via the format of the screen in Figure 16.

Since both an intersatellite link and the downlink part of the system are similar to the uplink part, i.e., the system is comprised of a transmitter and transmitting antenna, includes a path between earth and space, and terminates at a receiving terminal antenna and receiver, the program follows the same steps as have been previously outlined for the uplink. The program continues to provide feedback of the user's progress by reverse-highlighting those components of the system that have been addressed, displaying an elongated z-shaped crosslink and downlink signal, and moving the right rain cloud into the downlink signal path if the user clicks on it.

One enhancement available on the downlink side of the system is based on the fact that sometimes the satellite uses a single antenna for both reception and transmission. In many cases the downlink earth terminal antenna is the same size, shape, and efficiency as the uplink terminal antenna, too. If the satellite EIRP is unknown, the program queries the user to determine if the transmit antenna is the same as the receive antenna, again using the format of the screen in Figure 16. Similarly, if the downlink earth terminal G/T is unknown, the program queries the user to determine if the downlink earth terminal antenna is the same as the uplink. If the antennas are the same in either case, the program automatically determines the unknown antenna gain based on a ratio of the squares of the uplink and downlink frequencies. This procedure expedites the program's execution and precludes querying the user for the basic antenna parameters a second time.

Upon entering the communications system data, the downlink earth terminal receiver will be reverse-highlighted as shown in Figure 22. The "Show comm system parameters" button will flash, indicating that the user may now display the results of his link analysis/design by clicking this button.

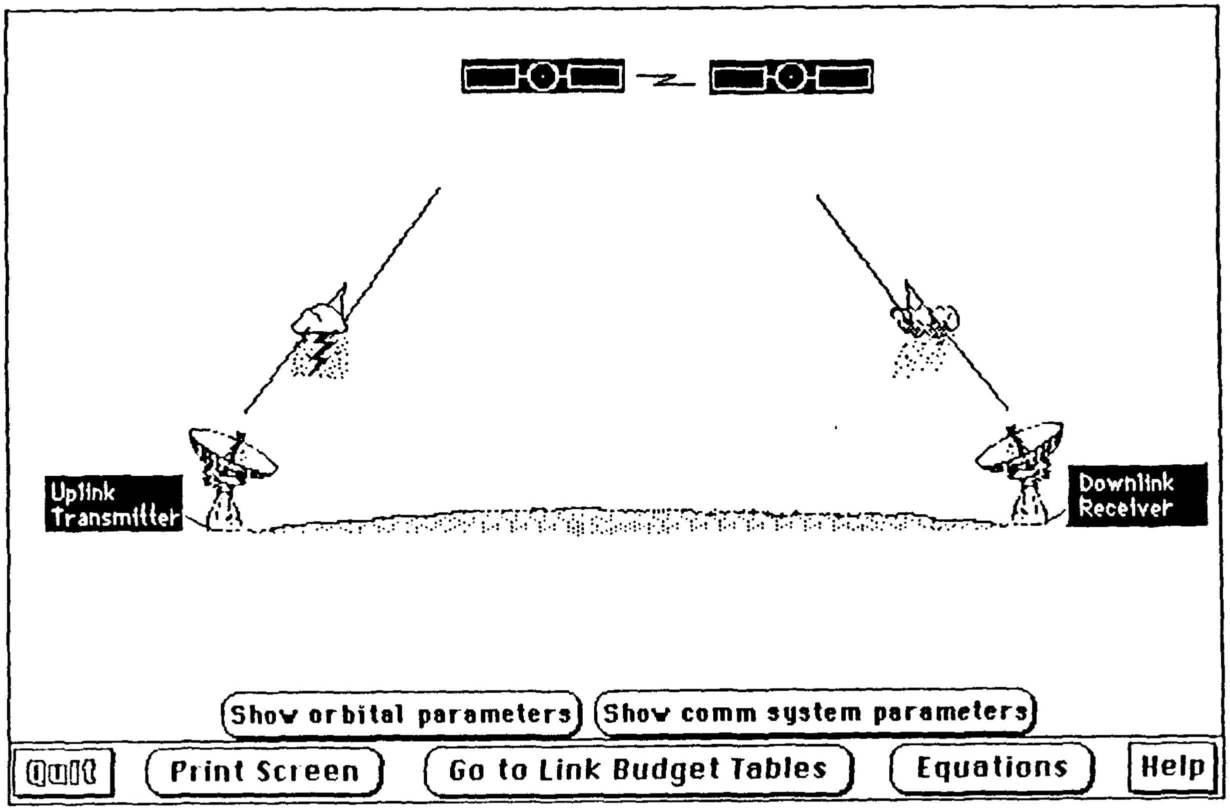

Figure 22. Picture of completed satellite system

At this point the user may get a hard copy of the communications system parameters, as shown in Figure 23, by clicking on the "Show comm system parameters" button.

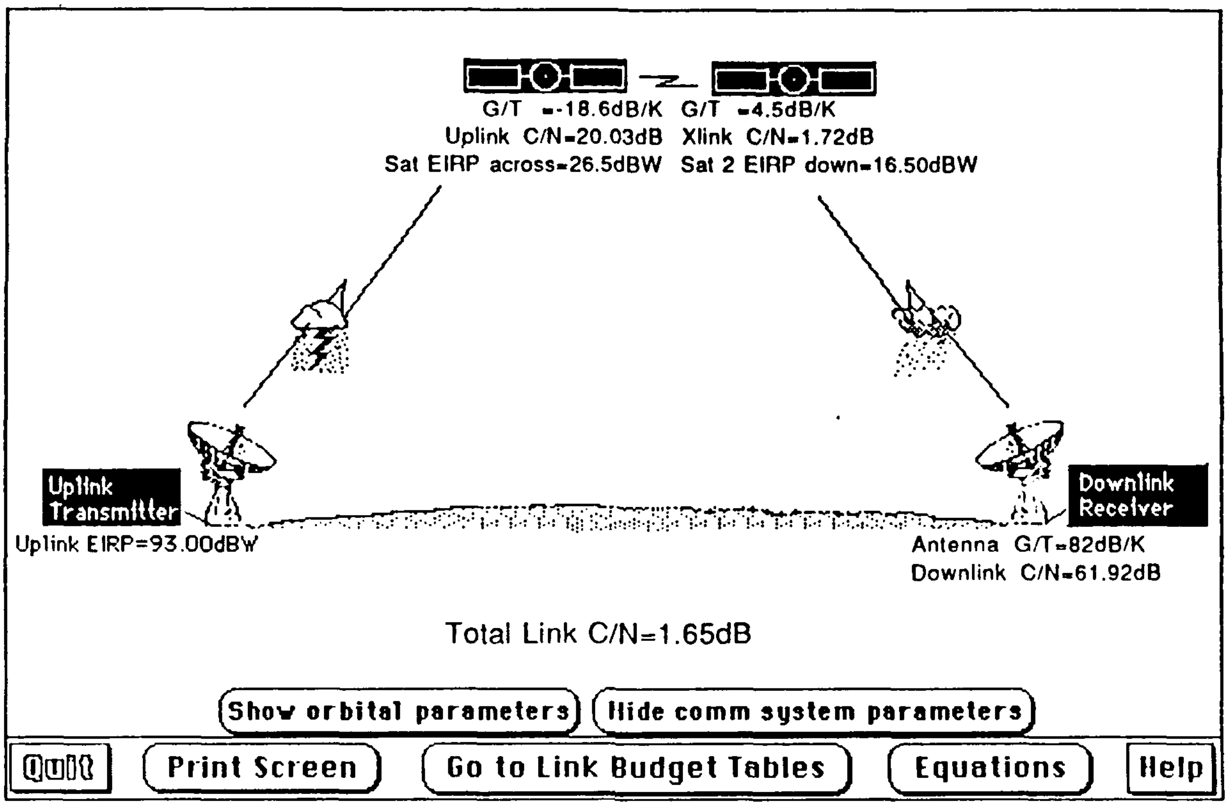

Figure 23. Picture of system with communications system parameters displayed

He can view the equations used by the program to perform its calculations by clicking the "Equations" button. Finally, the user can access the more detailed tabular output in the link budget tables by clicking the "Go to Link Budget Tables" button.

2. Program Equations

a. Uplink EIRP

The transmitter power at the input to the transmitting antenna is (dB equation):

$$P_T - Res_{EOL} - BO - L_{feeder} - L_{line} = P_{Net}\ (dB) \qquad (2.10)$$

where P_T is the transmitter saturated power rating in dB, Res_{EOL} is the reserve for end-of-life loss in dB, BO is the earth station TWTA output back-off in dB, L_{feeder} are the feeder losses due to couplers, filters, antenna feeds, etc., in dB, L_{line} is the transmission line loss in dB, and P_{Net} is the net power into the antenna.

The gain of an antenna with respect to an isotopic radiator is:

$$G = \text{Gain} = \frac{4\pi A \eta_a}{\lambda^2} \qquad (2.11)$$

where A is the antenna area in meters-squared, η_a is the aperture efficiency, and λ is the wavelength in meters.

If the antenna has a circular aperture, the physical aperture area is:

$$A = \frac{\pi D^2}{4}, \qquad (2.12)$$

where D is the antenna diameter in meters.

Alternatively, the antenna gain in dBi is

$$10 \log(G) = G\ (dB) \qquad (2.13)$$

Then the uplink EIRP is given by equation 2.14:

$$\text{EIRP} = P_{Net}(dB) + G(dB) - M_{Maint} - L_{point} - L_{Sat} \qquad (2.14)$$

where M_{Maint} is the maintenance margin in dB, L_{point} is the antenna pointing loss due to wind, snow, etc., in dB, and L_{Sat} is the antenna pointing loss, in dB, due to satellite motion .

b. Rain attenuation

In determining the rain-induced attenuation using the Crane global model, the program performs a series of calculations outlined in Reference 3. The first are for the frequency-dependent coefficients and are calculated as follows:

$$a = (4.21 \times 10^{-5})f^{2.42}, \quad 2.9 \le f \le 54 \text{ GHz} \qquad (2.15)$$

$$a = (4.09 \times 10^{-2})f^{0.699}, \quad 54 \le f \le 180 \text{ GHz} \qquad (2.16)$$

$$b = 1.41f^{-0.0779}, \quad 8.5 \le f \le 25 \text{ GHz} \qquad (2.17)$$

$$b = 2.63f^{-0.272}, \quad 25 \le f \le 164 \text{ GHz} \qquad (2.18)$$

where f is the frequency in GHz.

Next, the following parameters are calculated:

$$d = 3.8 - 0.6 \ln(R_p) \qquad (2.19)$$

$$x = 2.3R_p^{-0.17} \qquad (2.20)$$

$$v = 0.026 - 0.03 \ln(R_p) \qquad (2.21)$$

$$u = \frac{\ln(xe^{vd})}{d} \qquad (2.22)$$

$$D = \frac{H-H_0}{\tan(E)}, \quad E >= 10° \qquad (2.23)$$

$$D = (r_e + H_0)y, \quad E < 10° \qquad (2.24)$$

$$y = \sin^{-1}\left[\frac{g \cos(E)}{r_e+H}\right] \quad E < 10° \qquad (2.25)$$

$$L = \frac{D}{\cos(E)}, \quad E >= 10° \qquad (2.18)$$

$$L = -(r_e+H_0)\sin(E)+\sqrt{(r_e+H_0)^2\sin^2(E)+2r_e(H-H_0)+H^2-H_0^2}\ , \quad E < 10° \qquad (2.27)$$

$$L_r(dB) = \frac{aR_p^bL}{D}\left[\frac{e^{ubD}-1}{ub}\right], \quad 0 \le D \le d \qquad (2.28)$$

$$L_r(dB) = \frac{aR_p^bL}{D}\left[\frac{e^{ubd}-1}{ub} - \frac{x^be^{vbd}}{vb} + \frac{x^be^{vbD}}{vb}\right], \quad d \le D \le 22.5 \qquad (2.29)$$

where $L_r(dB)$ is the rain-induced attenuation experienced by the system.

Additionally, downlink rain attenuation increases the effective sky noise temperature, which adds to the system noise temperature of the downlink terminal receiver. The equation governing this relationship is:

$$\Delta T = 273\left\{1-\left(\frac{1}{\log^{-1}\left[\frac{L_r(dB)}{10}\right]}\right)\right\} \qquad (2.30)$$

c. Free-space path loss

The free-space path loss is proportional to the squares of the frequency and distance and is:

$$L\ (dB) = 20\log_{10}(s) + 20\log_{10}(f) + 20\log_{10}\left[\frac{4\pi}{c}\right] \qquad (2.31)$$

where s is the slantrange in kilometers, f is the frequency in GHz, and c is the speed of light, 2.997925×10^8 m/s. [Ref. 2:p. 327]

d. Antenna G/T

The system noise temperature referred to the input of the low-noise amplifier is in equation 2.32 :

$$T = \frac{T_A}{L_1} + \frac{L_1 - 1}{L_1} T_0 + T_{e2} + \frac{T_{e3}}{G_2} \tag{2.32}$$

where T_A is the antenna noise temperature, L_1 is the waveguide loss, T_0 is the ambient temperature, T_{e2} is the LNA equivalent noise temperature, T_{e3} is the downconverter equivalent noise temperature, and G_2 is the antenna gain. [Ref. 3:p. 87]

The system noise temperature in dB is 10 log(T) = T (dB). (2.33)
The G/T is then G (dB) - T (dB), where G (dB) is the antenna gain in dB.

e. Uplink C/N

The uplink carrier-power-to-thermal-noise power ratio, C/T, is (dB equation):

$$C/T = EIRP + G/T - L_{path} \tag{2.34}$$

where L_{path} is the total path loss and is the sum of the rain attenuation, free-space path loss, atmospheric loss, and propagation effect loss.

The uplink carrier-to-noise power ratio, C/N, is:

$$C/N = C/T + 228.6 - BO - B_N \tag{2.35}$$

where 228.6 is the reciprocal Boltzmann constant in dB(W/Hz)/K, BO is the satellite input back-off in dB, and B_N is 10 log(noise bandwidth in Hz).

f. Uplink illumination level at the satellite

$$\Omega\ (\mathrm{dBW/m^2}) = \mathrm{EIRP} - 163.3 + R - L_{path} \qquad (2.36)$$

where 163.3 is the maximum loss from the edge of the earth in dB/m^2 and R is the range correction factor:

$$R = 20 \log\left(\frac{41680}{\text{slantrange (km)}}\right) \qquad (2.37)$$

g. Intersatellite path loss

The intersatellite path attenuation is given by equation 2.38:

$$L_x\ (\mathrm{dB}) = 191.0 + 20 \log_{10}\left[\sin\left(\frac{1}{2}\alpha\right)\right] + 20 \log_{10}(f_x) \qquad (2.38)$$

where α is the longitudinal separation (in radians) of the two geosynchronous satellites and f_x is the intersatellite transmission frequency. [Ref. 2:p. 32]

h. Intersatellite EIRP, G/T, and C/N

The first satellite's EIRP and the second satellite's G/T are computed using the same equations as for the uplink, i.e., equations 2.14, and 2.32.

The intersatellite carrier-power-to-thermal-noise power ratio, C/T, is:

$$C/T_x = EIRP_x + G/T_x - L_x \tag{2.39}$$

where $EIRP_x$ is the crosslink EIRP from the first satellite to the second, G/T_x is the satellite antenna G/T of the second satellite, and L_x is the intersatellite path attenuation in dB.

The intersatellite carrier-to-noise power ratio, C/N, is:

$$C/N = C/T + 228.6 - BO - B_N \tag{2.40}$$

where 228.6 is the reciprocal Boltzmann constant in dB(W/Hz)/K, BO is the satellite input back-off in dB, and B_N is 10 log(noise bandwidth in Hz).

i. Downlink equations

The equations for the link from the satellite (or the second satellite, if there are two satellites) to the downlink earth terminal for EIRP, free-space path loss, rain attenuation, illumination level at the downlink antenna, earth terminal receiver antenna gain, G/T, and C/N are the same as above; only the input is different, e.g., downlink frequency and slantrange. Two additional equations are provided for analysis of the downlink, the electric field strength and the receiver input signal level.

The electric field strength is given by:

$$E\ (\mathrm{dB\mu V}) = \Omega\ (\mathrm{dBW/m^2}) + 85.7 \tag{2.41}$$

where Ω is the illumination level at the downlink earth terminal antenna in $\mathrm{dBW/m^2}$, and 85.7 is a constant.

The receiver input signal is given by:

$$W\ (\mathrm{dB\mu V/m}) = E\ (\mathrm{dB\mu V}) + G\ (\mathrm{dBi}) \tag{2.42}$$

where G is the receiver antenna gain in dBi.

j. Total C/N

The equation for the total link carrier-to-noise power ratio is:

$$\left[\frac{1}{(C/N_T)}\right] = \left[\frac{1}{(C/N_U)}\right] + \left[\frac{1}{(C/N_X)}\right] + \left[\frac{1}{(C/N_D)}\right] \tag{2.43}$$

where C/N_U is the uplink carrier-to-noise ratio, C/N_X is the crosslink carrier-to-noise ratio, and C/N_D is the downlink carrier-to-noise ratio, and all are absolute ratios, not dB values.

E. PART 3: THE LINK BUDGET TABLES

1. General

The link budget tables are designed to provide the user with a comprehensive set of tabular data regarding the key aspects of the system. There are two pages each for the uplink and downlink tables and a separate page for the intersatellite link, if required. A table line item may contain a zero if a derived parameter was entered instead of the basic parameters. A zero will also appear if a particular line item was not considered in the analysis or design.

2. The Uplink Budget Tables

The first page of the Uplink Budget Tables appears as shown in Figure 24.

Uplink Budget

Earth station longitude: 157°W	Azimuth angle: 102°	Antenna diameter: m
Earth station latitude: 23°N	Elevation angle: 17°	Uplink beam:
Uplink frequency: 6 GHz	Slant range: 39826 km	Satellite: 95°W

Earth Station

Transmitter saturated power rating at output flange (dBW):	0.00
Reserve for end-of-life loss (dB):	0.00
Output back-off for carriers (dB):	0.00
Net power into transmission line (dBW):	0.00
Transmission line loss (dB):	0.00
Other feeder losses (directional couplers, switches, filters, antenna feeds, and radomes) (dB):	0.00
Antenna gain on axis (dBi):	0.00
Nominal EIRP of earth station (dBW):	0.00
Maintenance margin (dB):	0.00
Antenna pointing loss (wind, snow, and foundation settling) (dB):	0.00
Antenna loss due to satellite motion (dB):	0.00
Worst case EIRP (dBW):	93.00

Earth-to-Space Path

Clear sky free-space path loss (dB):	-200.02
Atmospheric loss (dB):	-.50
Precipitation losses for 0.05 % of a year & propagation effect losses (scintillation, etc) (dB):	-2.72
Nominal uplink losses (dB):	-203.24

Quit　　Print Screen　　Next Page

Figure 24. Uplink Budget Table, page 1

The second page of the Uplink Budget Tables appears as shown in Figure 25.

Uplink Budget (Continued)	
Satellite	
Satellite antenna peak gain (dBi):	0.00
Receiving system noise temperature (dBK):	0.00
Satellite G/T (dB/K):	-18.60
Off-beam center loss (dB):	0.00
Pointing error (attitude control, thermal misalignments, etc.) (dB):	0.00
Nominal G/T (dB/K):	-18.60
Summary	
Earth station EIRP (dBW):	93.00
Up-link losses (dB):	-203.24
Satellite G/T (dB/K):	-18.60
C/T at satellite receiver output (dBW/K):	-128.84
1/Boltzmann constant (dB(W/Hz)/K):	228.60
Satellite transmitter input backoff (dB):	0.00
C/kT at transmitter input (dBHz):	99.76
or	
Earth station EIRP (dBW):	93.00
Range correction factor (dB):	0.40
Illumination level at the satellite (dBW/m^2):	-72.87
Quit · Print Screen · Intersatellite Link Tables	

Figure 25. Uplink Budget Table, page 2

3. The Intersatellite Link Budget Table

The Intersatellite Link Budget Table appears as shown in Figure 26.

Inter-satellite Link Budget

Orbital separation of satellites in degrees of longitude: 138° Separation in km: 78773.85
Crosslink transmission frequency 18 GHz

Transmitter

Transmitter saturated power rating (dBW):	0.00
Reserve for end-of-life loss (dB):	0.00
Losses due to switches, transmission line, etc. (dB):	0.00
Antenna gain (dBi):	0.00
Saturated EIRP (dBW):	0.00
Output back-off for carriers (dB):	0.00
Effective EIRP (dBW):	0.00
Pointing error (attitude control & margin) (dB):	0.00
Nominal EIRP (dBW):	0.00

Path

Path loss (dB):	182.57

Receiver

Figure of merit (G/T) (dB/K):	2.00
Pointing error (attitude control & margin) (dB):	0.00
C/T at reveiver output (dBW/K):	-154.07
1/Boltzmann constant (dB(W/Hz)/K):	228.60
C/kT at receiver output (dBHz):	74.53

Quit | Print Screen | Downlink Budget Tables

Figure 26. Intersatellite Link Budget Table

4. The Downlink Budget Tables

The first page of the Downlink Budget Tables appears as shown in Figure 27.

Downlink Budget

Earth station longitude: 72°E	Azimuth angle: 255°	Antenna diameter: m
Earth station latitude: 7°S	Elevation angle: 61°	Downlink beam:
Downlink frequency: 4 GHz	Slant range: 36481 km	Satellite: 48°E

Satellite

Transmitter saturated power rating (dBW):	0.00
Reserve for end-of-life loss (dB):	0.00
Losses due to multiplexer, filters, couplers, switches, transmission line hybrids, & feeds (dB):	0.00
Antenna gain (dBi):	0.00
Saturated EIRP (dBW):	26.50
Output back-off for carriers (dB):	-10.00
Effective EIRP (dBW):	16.50
Off-beam center loss (dB):	0.00
Pointing error (attitude control, thermal misalignments) (dB):	0.00
Nominal EIRP (dBW):	16.50

Space-to-Earth

Clear sky free-space path loss (dB):	-199.25
Atmospheric loss (dB):	-.50
Precipitation loss margin for 2.0 % of year (dB):	-.15
Other propagation effect losses (scintillation, polarization coupling, etc.) (dB):	-1.20
Increase in sky noise temperature due to precipitation (dB):	0.00
Total, nominal case (dB):	-212.09

Quit | Print Screen | Next Page

Figure 27. Downlink Budget Table, page 1

The second page of the Downlink Budget Tables appears as shown in Figure 28.

Downlink Budget (Continued)

Earth Station

Earth station clear sky G/T (dB/K):	30.99
Pointing error (due to satellite motion) (dB):	0.00
Earth station maintenance margin (dB):	0.00
Earth station performance (dB/K):	30.99

Summary

Satellite EIRP toward earth station (dBW):	26.50
Total downlink losses (dB):	-201.10
Earth station G/T (dB/K):	30.99
C/T at earth station receiver output (dBW/K):	-143.61
1/Boltzmann constant (dB(W/Hz)/K):	228.60
C/kT at receiver output (dBHz):	84.99
or	
Satellite EIRP toward earth station (dBW):	26.50
Range correction factor (dB):	1.16
Illumination levels (dBW/m^2):	-136.29
Constant (dB):	85.70
Electric field strength (dBµV):	-50.59
Earth station receiver antenna gain (dBi):	55.00
Receiver input signal (dBµV/m):	4.41
Total system C/N ratio (dB):	-1.42

Quit | Print Screen | Change Parameters | Equations

Figure 28. Downlink Budget Table, page 2

F. EQUATIONS

The user can review key equations used by the program to perform its calculations by clicking on the desired equation from the screen displayed in Figure 29.

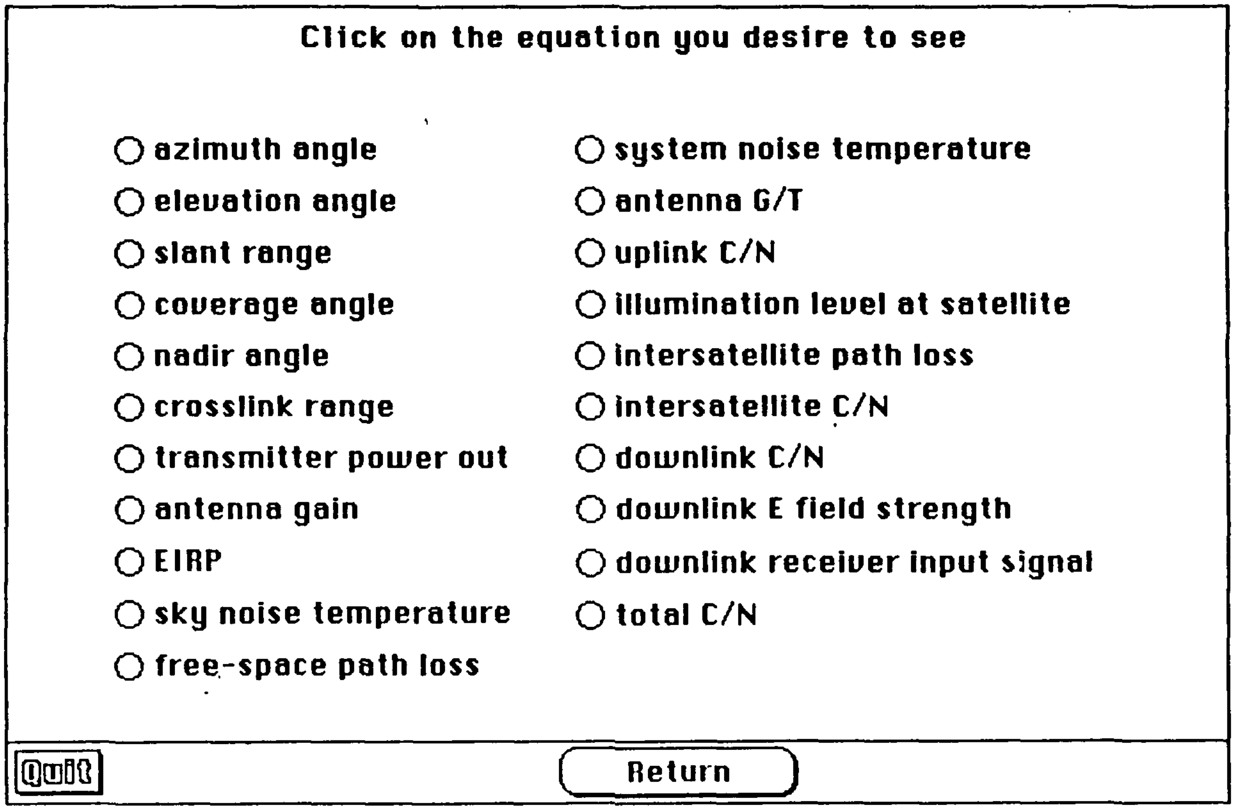

Figure 29. Equations

G. CHANGING KEY PARAMETERS

After reviewing and printing the link budget tables the user now has the option to change selected key parameters and re-run the program. By clicking on the "Change parameters" button displayed on the screen at the end of page two of the Downlink Budget Tables, as shown in Figure 28, the user will display the screen shown in Figure 30.

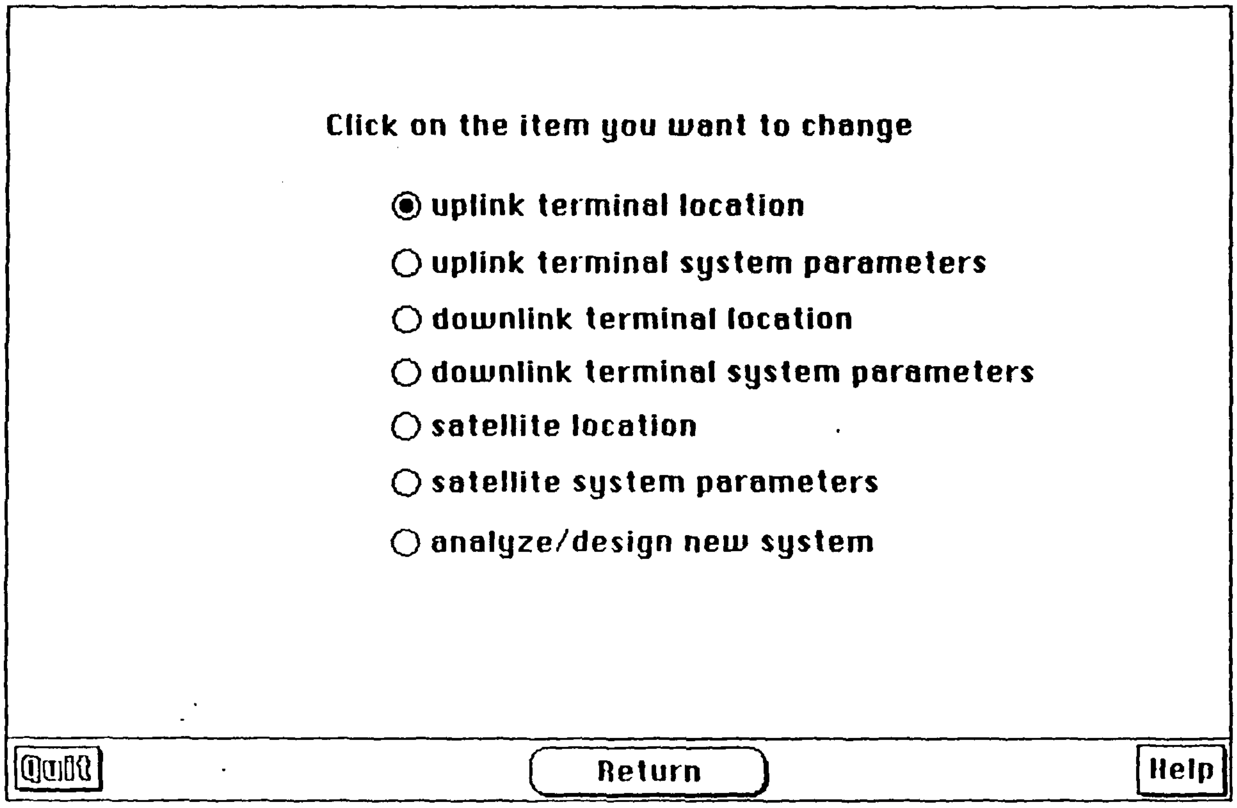

Figure 30. Key parameters that can be changed

When the user clicks on one of these buttons, the program takes him to the part of the program where that item was originally entered and queries the user to enter the new data in the same manner as the original data. Thus, the user is provided a consistent interface for data entry.

After the user has re-run the program with the new data, he can print out the new results and repeat the process.

Finally, the user exits the program by clicking on the "Quit" button, located in the lower left corner of most of the screens.

III. SUGGESTIONS FOR FOLLOW-ON THESIS WORK

A. GENERAL

The one megabyte of RAM limitation of the standard Macintosh precluded the inclusion of several capabilities in this project. For users with extended RAM memory, the following improvements can be implemented.

B. MORE ACCURATE MAPS

As stated earlier, maps are extremely memory intensive. It is acknowledged, however, that a mouse click on the world map used in this program can approximate the true location only to within about one degree of accuracy in longitude and latitude. The inclusion of other maps, which are accessed by clicking on a region on the world map, would allow much more precise specification of earth terminal locations.

C. LOW EARTH ORBITING (LEO) SATELLITES

The program, as it currently exists, addresses only geosynchronous satellites. Modifying the program so that it can analyze the communications and surveillance capabilities of low earth orbit, supersynchronous, and elliptical orbit satellites would greatly enhance its utility.

D. ITERATIVE CALCULATIONS

At the conclusion of the running of the program, the user has the opportunity to change specified parameters. All output is then re-calculated and a new set of link budget tables is produced. While this procedure allows the comparison of one system design with another, it does not lend itself to the rapid determination of how changing one parameter over a range of values will affect the system. Modification of the program to allow iterative calculations to determine the best value from a range of values would be a significant enhancement.

IV. CONCLUSIONS

As educators and academicians search for ways to use computers to enhance their students' "learning experience," innovative software engineering tools provide one means of achieving this goal. The object-oriented HyperCard language combines ease of use with sophisticated graphics tools to facilitate the development of visually-oriented, user-friendly interactive software. Products developed in this environment can aid in visualizing abstract concepts, thereby enhancing student understanding.

The success of this program validates the choice of HyperCard as the appropriate software development environment for this application.

APPENDIX A: HYPERCARD OVERVIEW

I. BACKGROUND

The choice of platforms for this thesis was guided by these initial requirements:

1) the program must be graphics-oriented/menu driven,

2) the user must not have to purchase additional software to use this program,

3) the user must not have to learn any language or command sequence(s) to use the program productively, and

4) the user must be able to modify the program with minimal effort.

The first requirement implies high quality graphics. While many of the personal computers available on the market in 1989 could have met this requirement, the inherent high quality graphics capabilities of the Apple Macintosh made it an obvious candidate.

The second requirement implied that the development language chosen for the program either had to be included (bundled) with the hardware system, if the program was to run in an interpreted mode, or the entire program had to be provided to the user in compiled form. The fourth requirement, that the program be easily modifiable by the user, eliminated the compiled program since this form of the program requires a language compiler to enable the user to modify the program. The only interpretive language that usually comes bundled with a hardware system is some version of the BASIC programming language. As BASIC has fallen from favor as an application development language and computers have advanced in power and complexity, even this bundling has become less frequent. The lone exception to this trend is Apple's bundling of HyperCard with each of its Macintosh computers since August 1987 [Ref. 4:p. 11].

The third requirement was easily satisfied by the Macintosh, since all essential interactions between the user and the computer (except for data entry) are accomplished via the mouse, that is, by pointing and clicking on the desired icon or menu selection. IBM personal computers, on the other hand, generally require a minimal proficiency in the Disk Operating System (DOS) before any application can be used.

The fourth requirement, as previously stated, eliminated compiled languages since any modification would require both ownership of the compiled language as well as programming fluency in it. HyperCard, with its English-like syntax, is comparatively easy to learn. Furthermore, mastery of HyperCard is not required to make significant modifications to an existing program.

Thus, the Apple Macintosh was selected as the development platform for this project, with HyperCard as the implementation language.

II. WHAT IS HYPERCARD?

HyperCard is an authoring system which implements hypertext concepts in an object-oriented programming environment.

A. Authoring Systems

An authoring system provides the programmer with an integrated set of software tools with which to create interactive applications that communicate knowledge [Ref. 5:p. 71].

HyperCard's integrated toolkit includes digitized sound, pixel-level control of bit-mapped graphics, special visual effects, variable text styles and fonts, a library of built-in mathematics functions and common programming language constructs (e.g., If...then...else, Repeat..until), and extensibility through routines written in a general purpose programming language (e.g., Pascal). Additionally, HyperCard supports a range of authoring levels, which vary according to the author's expertise.

B. Hypertext Concepts

At its most basic level hypertext is a database management system which associatively links screens of information. Each screen, or *node*, represents a single idea or concept. One node is connected to another via a link, which usually originates at a single *point*. A point usually identifies a link via an icon or text string, which pictures or names the destination node. [Ref. 6:p. 237]

HyperCard nodes are called *cards* and can contain any combination of text, graphics, and audio/video data. Cards are linked via points, called *buttons*. Buttons can be various sizes and shapes, appear as icons, or be invisible. The HyperCard user traverses links between cards by clicking on buttons.

C. Object-Oriented Programming (OOP) Environment

While most high-order programming languages (e.g., FORTRAN, PASCAL) are procedural, i.e., active procedures act on passive data passed to them, in an OOP environment objects (data) perform operations on themselves. Computations are performed by sending a message to an object, which invokes a procedure hidden inside the object. OOPs are characterized by four elements: information hiding, data abstraction, inheritance, and polymorphism. Information hiding involves the manipulation of data within a module (subroutine or procedure) such that the status of internal data and variables is hidden and only the output of the module is known. Data abstraction allows the programmer to define an abstract data type with an internal representation and a set of procedures to access and manipulate the data, thereby hiding information. Inheritance embodies the concept of lower level objects in a hierarchy inheriting properties from higher level objects. Polymorphism permits the same message to elicit a different response from different objects to which it is sent. [Ref. 7:p. 372]

There are five types of objects in HyperCard: buttons, fields, backgrounds, cards, and stacks. Information hiding occurs in the hidden *scripts* of all five types of objects, which, when activated by a system message, execute procedures which determine the interactions between the objects. Programmer-defined global and local variables are abstract data types used inside the scripts to manipulate data. The hierarchical ordering of the five types of objects is shown in Figure 31. Each level up on the hierarchy is more general than the one below it, i.e., it includes all of the objects on the levels below it. This hierarchy determines a system message's inheritance path, i.e., how the message will be passed up the hierarchy. Polymorphism allows the *Print* command to be executed by each type of object in the proper way.

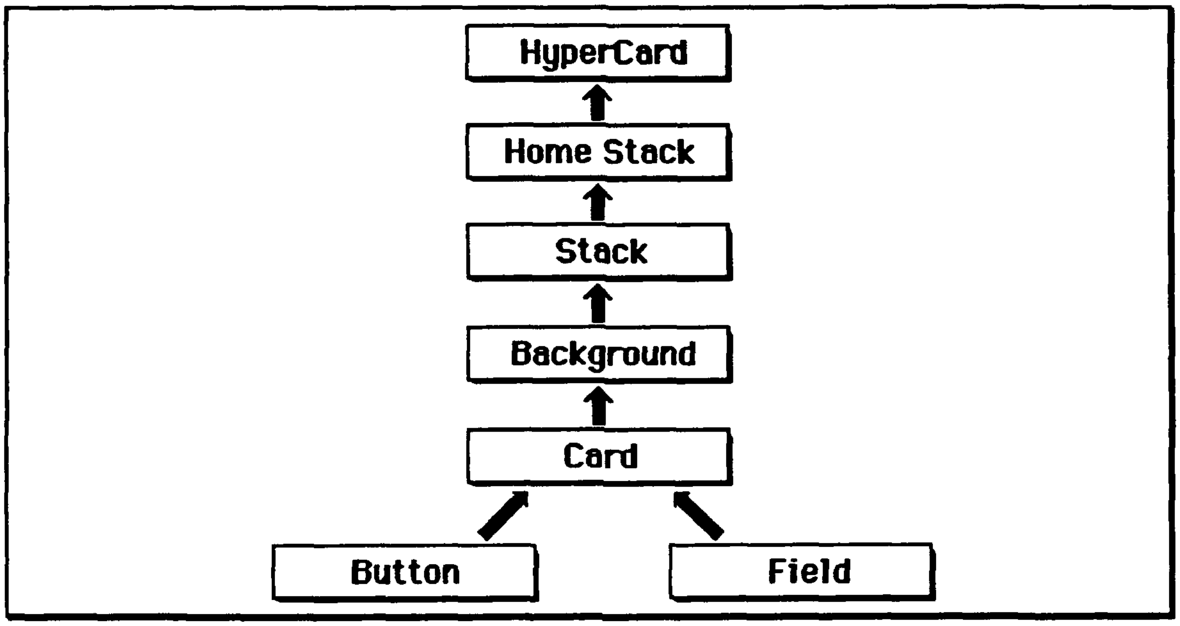

Figure 31. HyperCard Object Hierarchy [Ref. 4]

To the HyperCard user each of the five types of objects has a readily observed, practical role. Buttons allow the user to navigate throughout the program and make desired selections. Fields allow the user to enter text and data into the program for manipulation and for calculated output to be displayed. Backgrounds provide a common graphical context for the cards. Cards provide the hypertext nodes which are linked associatively to comprise the stack. The stack is the collection of cards.

III. HOW DOES HYPERCARD WORK?

The catalyst for all HyperCard actions is the *system message*. HyperCard generates a system message and sends it along the hierarchical path to an object(s) whenever certain events take place. Clicking the mouse generates both *mouseDown* and *mouseUp* system messages. Moving to a new screen (card) generates a *closeCard* message, which is sent to the old card, and an *openCard* message, which is sent to the new card. [Ref. 8:p. 2]

The programmer creates an object and writes a script for the object containing *message handlers*, which define how that object responds to a particular message. The message handler is analogous to a procedure or subroutine; when the name is invoked, the commands inside the handler are executed sequentially. All HyperCard message handlers begin with *on* followed by the name of the message. All message handlers terminate with *end* followed by the message name. The following script would generate a beep, then jump to the next card in the stack when the mouse was clicked:

```
on mouseUp
   beep
   go to next card
end mouseUp
```

HyperCard uses the object hierarchy for passing system messages from the lowest level object (a button or field on a card) up through the hierarchy of objects (the card, background, and stack) as shown in Figure 32. This message passing continues until an appropriate message handler is located in the script of an object, or until the top of the hierarchy (HyperCard) is reached. When a matching message handler is found anywhere along this hierarchical path, the message is trapped (intercepted) and executed.

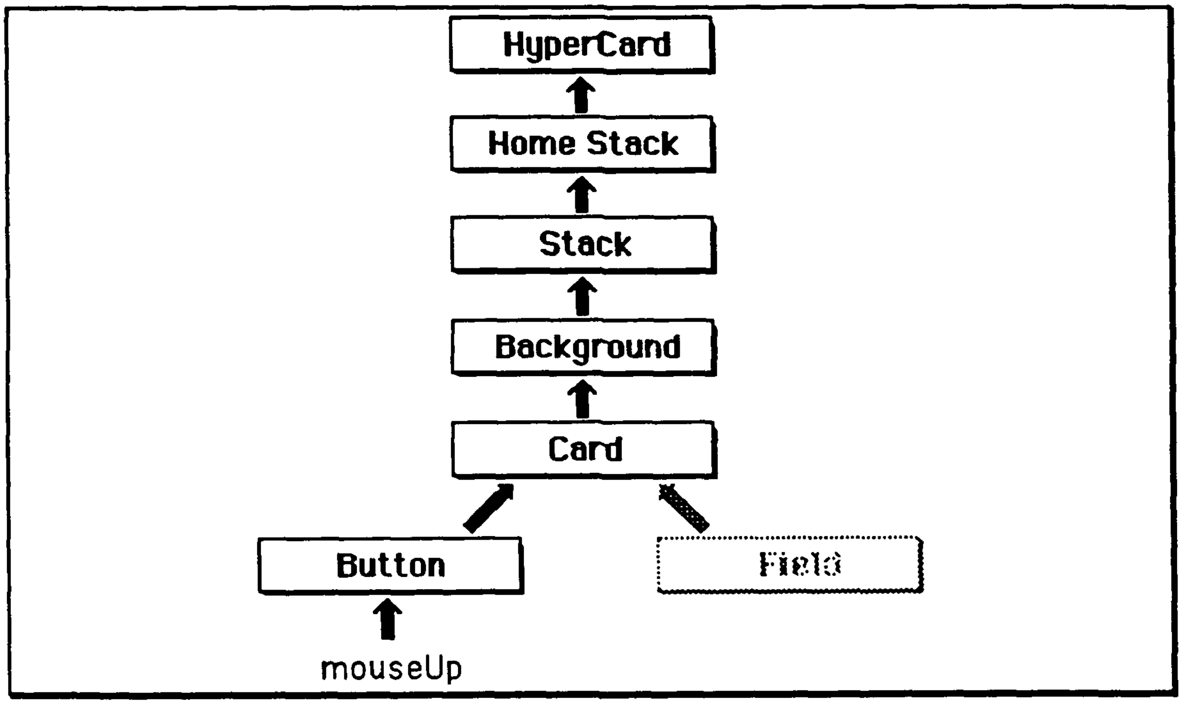

Figure 32. Passing System Messages [Ref. 4]

All HyperCard objects are placed in layers as they are created. The card's background serves as the bottom layer or base. The last object created resides on the top layer. The layer in which an object resides is important since many scripts in the hierarchy may contain similar message handlers.

Much of HyperCard's power lies in the fact that the programmer is not limited to using only the system messages generated by HyperCard. The programmer can create task-specific messages and message handlers and, by placing the handlers in an object high in the hierarchy, make them available to numerous buttons on many different cards.

APPENDIX B: SPECIFIC PROGRAMMING CHALLENGES

I. GENERAL

The HyperCard object and message handling hierarchy presented unique advantages and challenges. If, for example, the last object created is a card button with an *on mouseUp* message handler, then this handler is executed only when this button is clicked. Conversely, when this button is clicked, the handlers of other objects (in all layers) *may* be bypassed when the *on mouseUp* is executed. If a small button overlays a larger button on a card and both have an *on mouseUp* message handler in their scripts, when the small button is clicked, it will intercept the message since it is the button in the top-most layer. The *on mouseUp* message handler in the script of the larger button, which is on a layer beneath the smaller button, will be bypassed. Certain other message handlers (if they exist), e.g., *on closeCard* , may be executed, however, even though they lie in layers under the layer holding the clicked button. Thus, the programmer must maintain cognizance of these handlers and the actions they will generate.

II. CRANE GLOBAL RAIN CLIMATE REGIONS

One design goal which proved unexpectedly difficult to achieve was for one click of the mouse on the world map to result in the determination of the longitude, latitude, and rain climate region. The longitude determination was straightforward; use of the conformal mercator projection map yields a linear relationship between the screen horizontal pixel location and a longitude. Since ranges of latitudes are not equal on this type of map projection, correction factors had to be calculated for each range of ten degrees of latitude.

More difficult was the determination of rain climate region. There are four main models used to predict rain attenuation. The Crane global model was chosen for this project because of its accuracy and common use in student textbooks. The user, therefore, is more likely to be familiar with this model.

Several approaches were tried before success was achieved. The first effort involved drawing the rain climate regions in the background graphics, i.e., on a world map on the background layer, with the same world map (without the rain regions) superimposed on top of the graphics layer to hide the rain climate region graphics. Each rain region was shaded with a different pattern from the standard palette of HyperCard patterns. While HyperCard has a built-in function which can detect the pattern which exists where the mouse is clicked, this procedure results in the appearance of a dialog box, which interrupts the program flow and queries the user for an answer that he cannot provide. The second approach involved approximating the irregular shapes of the rain regions with various sizes of rectangular buttons. This approach was discarded because of the large number of small buttons that were required to accurately approximate each of the many irregular shapes. The solution to this problem was found in the program PolyButtons [Ref. 9:p. 137], a HyperCard script which creates and permits the use of polygonal buttons. PolyButtons was used to create the irregularly-shaped rain climate regions, then the section of PolyButtons which recognizes the polygonal buttons was extracted and modified for inclusion in the stack script.

III. MEMORY SPACE-SAVERS

An important constraint in developing any HyperCard stack is the one megabyte random access memory (RAM) limitation of the standard compact

Macintosh. Since the HyperCard application requires 387 kilobytes of RAM and the System and Finder typically occupy another 350 kilobytes of RAM, the stack cannot exceed about 250 kilobytes. A larger stack runs the risk of exceeding the available RAM, in which case an error message will appear and the stack will not run. The following are several examples of strategies employed to minimize the size of the stack.

A. Surface Point Rain Rate

There are ten rain region designations in the Crane global rain model. The original approach used was to build a separate card for each of these ten regions, with the stack jumping to the appropriate card (region) based upon the location of the earth terminal. Each card had 12 surface point rain rates (percentages) from which the user chooses one. Each point rain rate corresponded to a number, which varied from region to region. When the user selected a point rain rate from a region, the corresponding number was loaded into a global rain rate variable. Since all ten cards had the same 12 surface point rain rates (percentages) and only the numbers corresponding to each point rate changed based on the region, one card could be designed with 12 fixed point rain rates and a field corresponding to each rate. When the stack jumped to this card, numbers were placed into each field corresponding to each of the 12 point rain rates based on the rain region. This reduction from 10 cards to one card saved 100k -12k = 88k of stack size. The single card is 2k larger than the other 10 cards due to the large script required to implement this strategy.

B. World Map

The original approach to having the user click on the earth terminal and sub-satellite point locations involved three maps of the world with instructions appropriate to each. By creating a text field into which differing instructions could be placed based upon flags, one map was used to accomplish the same results as three. The addition of an invisible field, which held a long line and became visible only when the user was queried for the sub-satellite location, yielded an equator on the map. This reduction from three maps to one resulted in a savings of 45 -17 = 28k of stack size.

Another option considered early in the development of the stack was the use of multiple maps. In a multiple map program the user would click on a region on a global map; this action would result in another map appearing, which was a blow-up of the region first clicked. Subsequent clicks would traverse a series of maps of increasingly smaller geographic areas. While this approach was esthetically pleasing and permitted the user to gain greater accuracy in specifying a particular location, its memory requirements were prohibitive. A collection of 16 maps, scanned at minimally acceptable resolution of 72 dots per inch, occupied 160k of memory, fully two-thirds of the limit for the entire stack.

C. Placement of Shared Message Handlers

Several instances arose in the development of the program where the same calculations were needed by different parts of the program. By writing a message handler that performed the calculations and placing the handler in the script of an object higher in the hierarchy than any of the objects which send it messages, one message handler accomplished the calculations which would otherwise have required the complete handler to appear in the script of many lower level objects.

IV. INTERSATELLITE COMMUNICATIONS

The coding problem arises from the fact that all satellite positions are given with reference to the 0° longitude, i.e., 150° W or 30° E. If both satellites are in the same hemisphere as shown in Figure 33(a), the solution is to take their longitudinal difference and compare the result to 162.6°. If the satellites are in different hemispheres, but both longitudes are less than 90° as shown in Figure 33(b), the solution is to sum their longitudes and compare that sum to 162.6°. If the satellites are in different hemispheres, but both longitudes are greater than 90° as shown in Figure 33(c), the solution is to subtract the sum of the two longitudes from 360 ° and compare the result to 162.6°. The most difficult problem occurs when the satellites are in different hemispheres and one satellite's longitude is less than 90°, while the other's is greater than 90° as shown in Figure 33(d). In this case one must measure their longitudinal separation in both directions. The first simplifying step, however, is to convert all measurements to one reference direction by subtracting the larger longitudinal value from 360°. Thus, the direction of measurement is dictated by the smaller numerical longitude. If the difference between the longitudes is between 162.6° and 197.4°, the satellites will be hidden from one another. If the difference is greater than 197.4° in one direction, then it will be less than 162.6° when approached from the opposite direction and the satellites will be mutually visible.

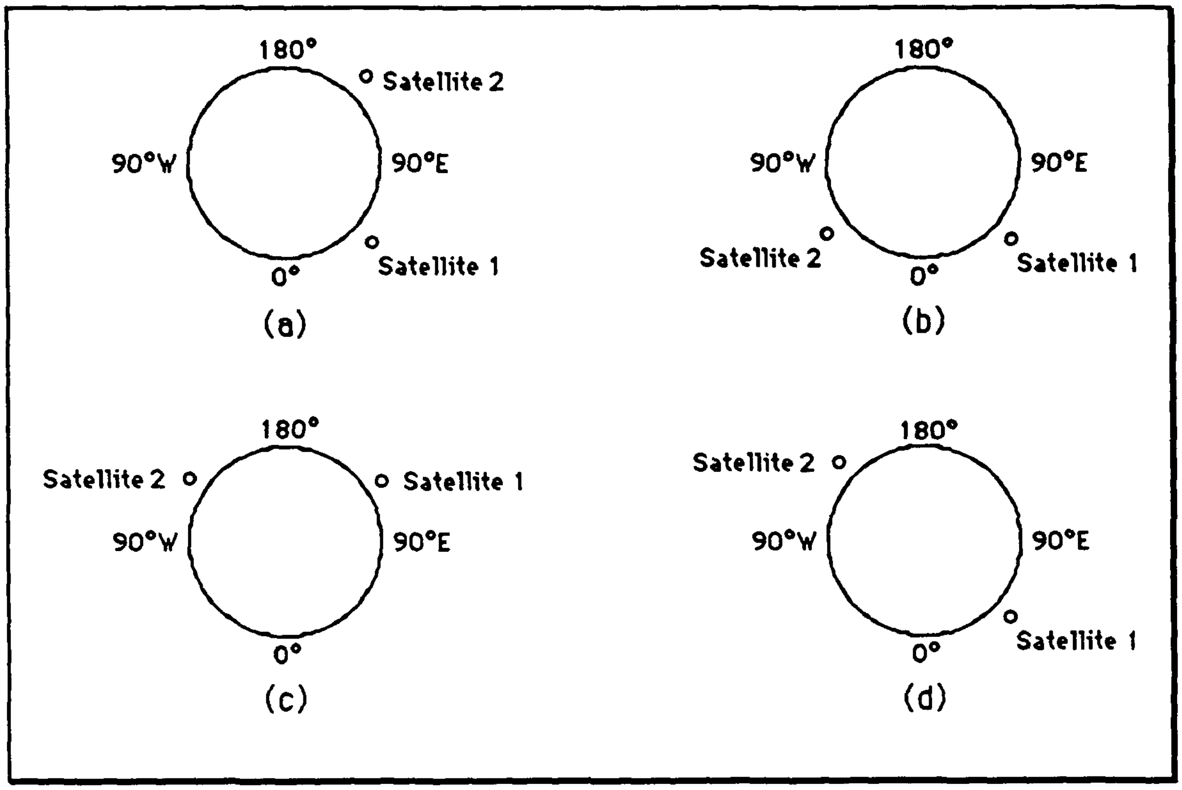

Figure 33. Inter-Satellite Communications Problem

V. INADVERTENT LINKAGES

The use of the same global variable for calculations in more than one card poses special risks. The results of a numerical calculation are often sent to another card for display in an output field or subsequent calculations. Many times, however, all output fields are emptied upon the opening of the card. Thus, the potential for an empty variable to appear in a calculation arises. This results in a NAN (Not A Number) or INF (Infinity) error message. An interactive de-bugging program was used to trace through the program a line at a time and observe the changing of the value of the particular variable.

VI. POP-UP HELP FIELDS

The desire to assure "user-friendliness" necessitated the inclusion of some form of on-line Help function, which would provide specific assistance regarding the task at hand. The original plan envisioned a single large scrolling field with the help topics and their corresponding explanations, instructions, and examples. The Help button used to activate this field was placed in the background to provide accessibility from all cards and eliminate the need for individual Help buttons on each card. While this concept used minimal memory space (i.e., only one button and one large field were required), the large Help field scrolled slowly, frustrating attempts to quickly obtain help on certain topics. It also limited the types of helpful information that could be provided in the limited available space of the scrolling field. Thus, the original concept was discarded in favor of individual Help buttons which could be tailored to any specific task.

Individual pop-up Help fields and on-screen notes to the user were chosen as the solutions. Implementing this strategy generally involved a minimum of two buttons and one field on cards which did not provide an on-screen note to the user. The first button, or Help button, was placed in the lower right corner of each card for consistency. Upon interception of the *mouseUp* message, this button hid itself, showed a "Hide Help" button in its place and made the help field visible. The "Hide Help" button simply reversed the process. This strategy permitted help fields of variable design, i.e., the type, size, and shape of the field was tailored to the specific type of help information to be conveyed.

LIST OF REFERENCES

1. World map, Prototype Analysts Workshop (PAWS), Navy Exercise Support Terminal (NEST).

2. Morgan, W.L., and Gordon, G.D., *Communications Satellite Handbook*, pp. 222-225, John Wiley & Sons, 1989.

3. Ha, T., *Digital Satellite Communications*, Macmillan, 1986.

4. Goodman, D., *The Complete HyperCard Handbook*, p. 11, 2nd Ed., Bantam Books, 1988.

5. Dear, B.L., "HyperCard - what is it?," *Byte*, pp. 71-80, 1988 Mac Special Edition.

6. Fiderio, J., "A grand vision," *Byte*, pp. 237-244, October 1988.

7. Pascoe, G.A., "Elements of object-oriented programming," *Byte*, p. 139, August 1986.

8. Apple Computer, *HyperCard Script Language Guide: The HyperTalk Language*, p. 2, Addison-Wesley, 1988.

9. The Waite Group, *Tricks of the HyperTalk Masters*, pp. 137-144, Hayden Books, 1989.

BIBLIOGRAPHY

Agrawal, B.J., *Design of Geosynchronous Spacecraft,* Prentice-Hall, 1986.

Apple Computer, *HyperCard Stack Design Guidelines*, Addison-Wesley, 1989.

Bond, G., *XCMD's for HyperCard*, MIS Press, 1988.

Crabb, D., "Learn on me," pp. 143-146, *Byte*, July 1989.

Frisse, M., "From text to hypertext," pp. 247-253, *Byte*, October 1988.

Harvey, G., *HyperTalk Instant Reference*, Sybex, 1988.

Heid, J., "Getting started with HyperCard," pp. 243-253, *MacWorld*, May 1989.

Hindus, L.A., "Hide and seek data," pp. 207-210, *MacUser*, July 1988.

Lasky, R.D., "HyperTalk program design," pp. 205-211, *Byte*, August 1989.

Loeb, L.H., "Hypercard - how does it work?" pp. 78-80, *Byte*, 1988 Mac Special Edition.

Rall, K., and others, *Wild Things*, Language Systems Corporation, 1989.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943-5100	2
3.	MAJ Charles C. Howard 166 Beechwood Drive Oakland, California 94618	2
4.	Prof. Donald v. Z. Wadsworth Code EC/Wd, Naval Postgraduate School Monterey, California 93943	1