

Figure 1. Satellite Communications Simulation.

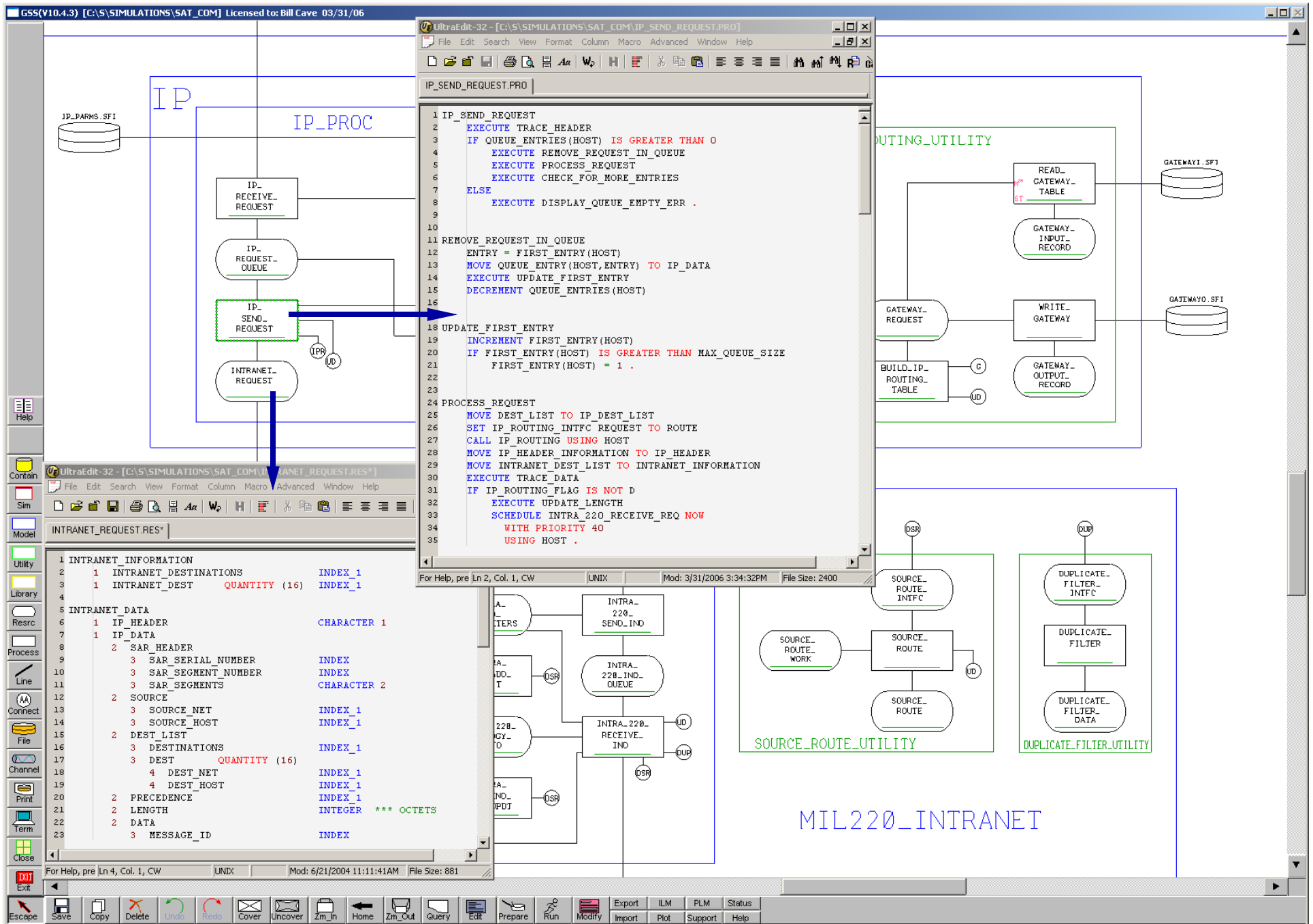


Figure 2. SAT_COMM Simulation with resource INTRANET_REQUEST and process IP_SEND_REQUEST open.

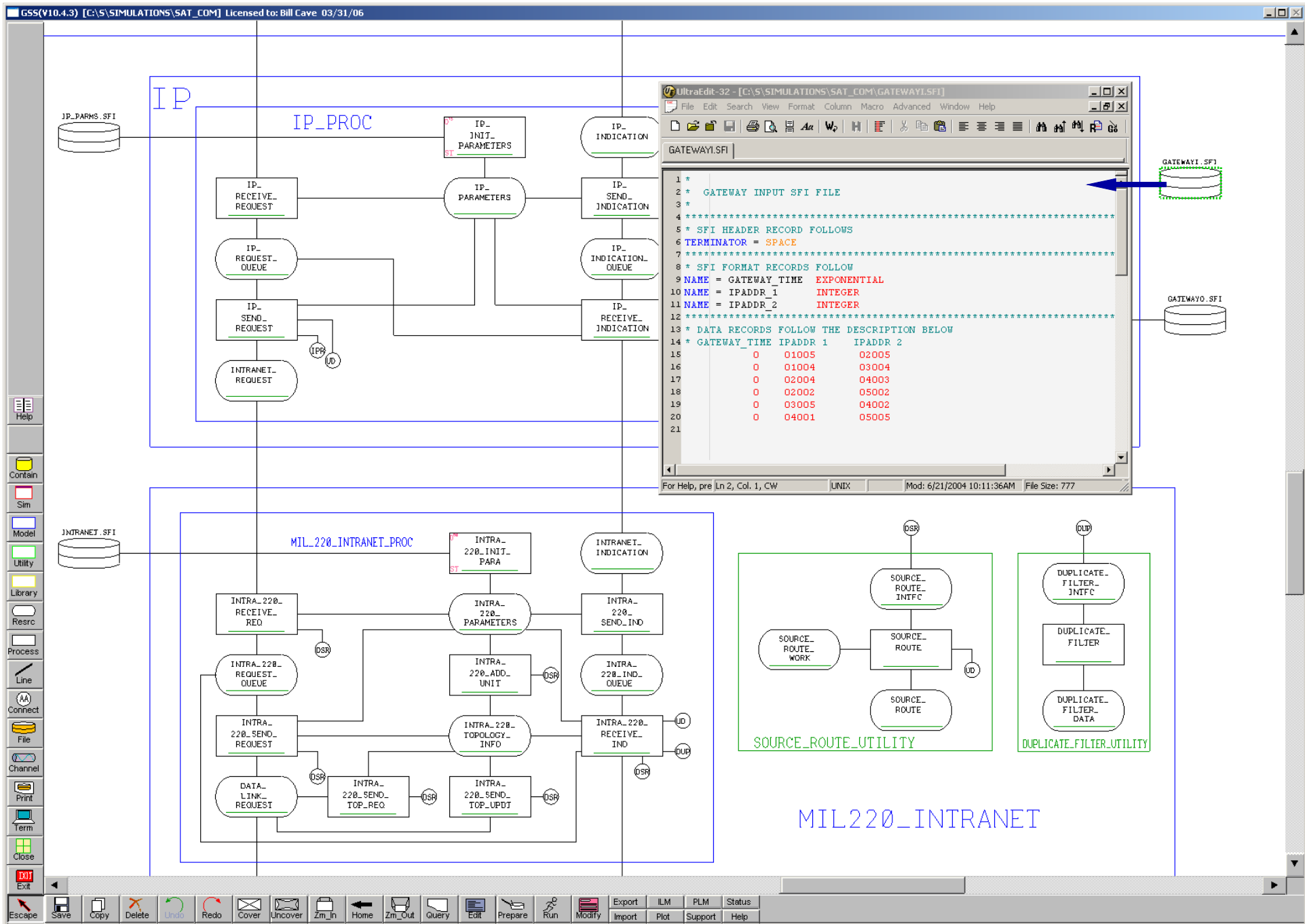


Figure 3. SAT_COMM Simulation, with file GATEWAY1.SFI open.

The image displays a simulation environment with two main windows. The left window, titled 'UltraEdit-32', shows the control specification file 'SAT_COMM.GSS'. The right window shows a network diagram with various protocol stacks and data links.

SAT_COMM.GSS Control Specification:

```

1 CONTROL SECTION
2 TITLE, SATELLITE SIMULATION
3 SIMULATE
4 ***TRACE
5
6 LIBRARY SECTION
7 C:\S\LIBS\FPPSLIB
8 C:\S\LIBS\SAT_LIB
9 C:\S\LIBS\RTG_DRAW
10 C:\S\LIBS\GENERAL
11
12 GRAPHICS SECTION
13 ACTIVATE
14 WORLD_SPACE LOWER_LEFT=(-35000,-11000),
15 UPPER_RIGHT=(100000,101000)
16 *** INITIAL_WINDOW LOWER_LEFT = (-12000, -8000), WIDTH = 100000
17 NVS/BDS = .2
18 ICON RADIO = TRUCK
19 ICON TANK = TANK
20 ICON SATELLITE = SATELLITE
21 ICON PLANE = PLANE
22 LINE_GATEWAY_LINK = COLOR GREEN, STYLE 2, THICKNESS 1
23 INST COLLISION PERCENT = THERM_BOX_VERTICAL,
24 LOW 0, HIGH 100, INITIAL_VALUE 0, COLOR GREEN
25
26 OVERLAY 1 = DRAW_CELL_CONTOURS IN CELLDRAW IN OVERLAYS
27 AT 0, 0, SCALE 20, 20, MENU CONTOURS
28 COLOR BROWN(40)
29
30 OVERLAY 2 = DRAW_GRID_LINES IN GRID_DRAW IN OVERLAYS
31 AT 0, 0, SCALE 1, 1, MENU GRID_LINES
32 COLOR LINE_GREEN
33
34 RTG_EVENT_HANDLER PROCESS_GRAPHICS_EVENT
35
36 DEFINITION SECTION
37 BANNER
38 READ_ORGANIZATION
39 INIT_MS_THREAD_TABLE
40 READ_MISSION_THREADS
41 READ_BER_LINE_TABLE
42 INIT_GRAPHICS
43 READ_NETWORK_DATA
44 RCP_CONVOLUTION_READ
45 SIP_READ_BER_TABLE
46 HQ_READ_BER_TABLE
47
48 SAT_COMM_MODELS
49 UDP_STACK
50 MIL_STD_188_220
51
52
53 DATABASE INPUTS
54 ASSIGN SFI 'BER_LINE.SFI' TO READ_BER_LNE_SFI_PROCESS
55 ASSIGN SFI 'SIP_BER.SFI' TO SIP_BER_TAB_SFI_PROCESS
56 ASSIGN SFI 'CONNECT.SFI' TO READ_CONNECTIVITY_SFI
57 ASSIGN SFI 'COUL.SFI' TO COUL_INIT_PARAMETERS *** USE FOR TCP
58 ASSIGN SFI 'DEFAULT.SFI' TO READ_DEFAULT_PARAMS
59 ASSIGN SFI 'DEPLOYIN.SFI' TO READ_DEPLOYMENT_SFI
60 ASSIGN SFI 'DLL_PARM.SFI' TO DLL_READ_PARAMETERS
61 ASSIGN SFI 'ENVIRNMT.SFI' TO READ_ENVIRONMENT_PARAMS
62 ASSIGN SFI 'GATEWAYI.SFI' TO READ_GATEWAY_TABLE
63 ASSIGN SFI 'HQ_BER.SFI' TO HQ_READ_BER_FILE

```

Network Diagram Components:

- TCP_UDP_STACKS:** A large container box containing several sub-diagrams:
 - TCP_STACK:** Includes sub-diagrams for EQUAL_PROXY, COUL, and TOP_PROC.
 - UDP_STACK:** Includes sub-diagrams for REL_DEST_ADDR, REL_SRC_ADDR, and SECURITY_HDR_READEN.
- MIL_STD_188_220:** A central data link component.
- MIL220_INTRANET:** A network layer component.
- MIL_220_DATA_LINK:** A data link component.
- MIL_220_PHYSICAL_LAYER:** A physical layer component.

The diagram shows a complex network topology with various nodes and connections, representing the simulation environment.

Figure 4. SAT_COMM Simulation with control specification open.

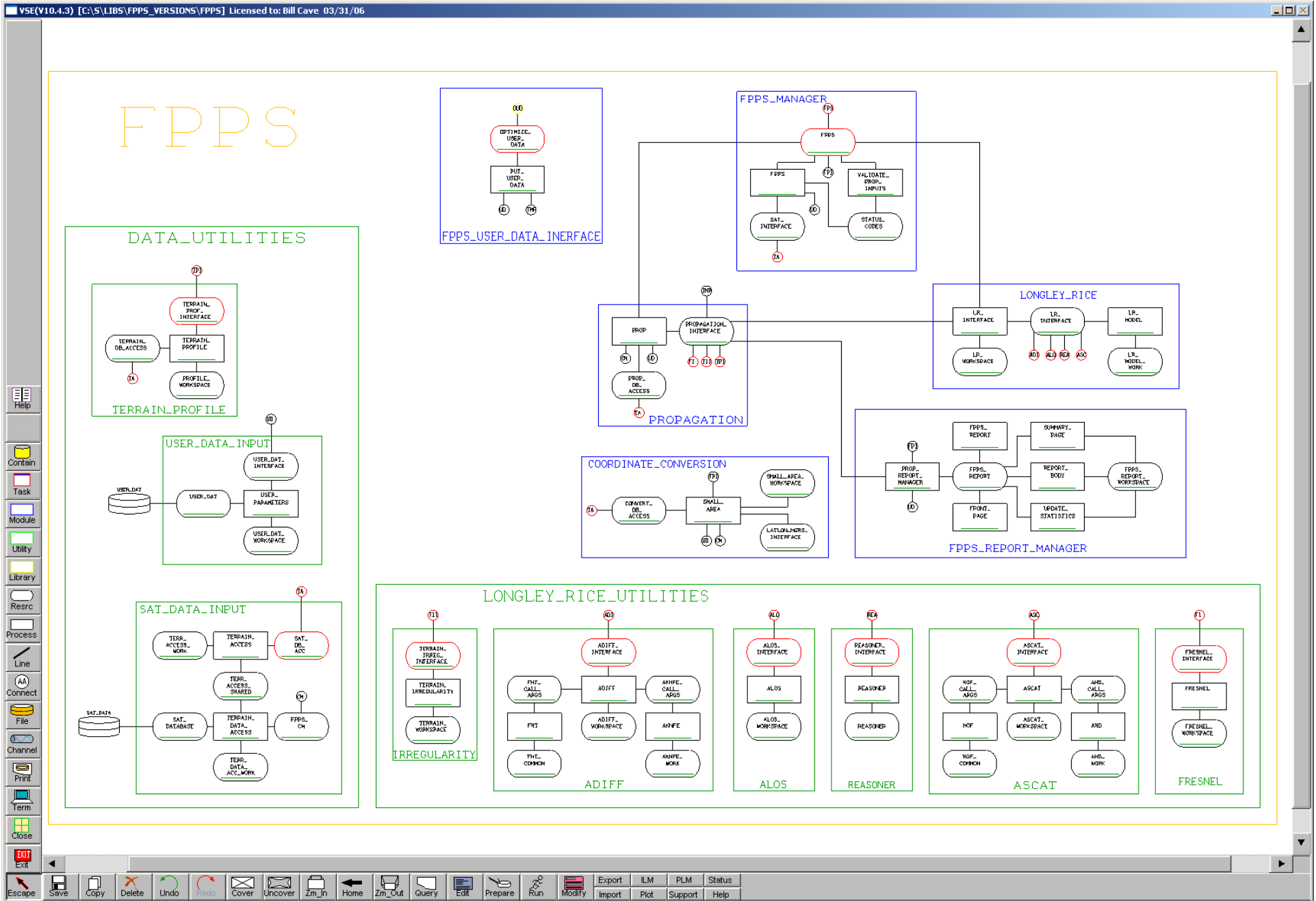


Figure 5. FPPS Library Module.

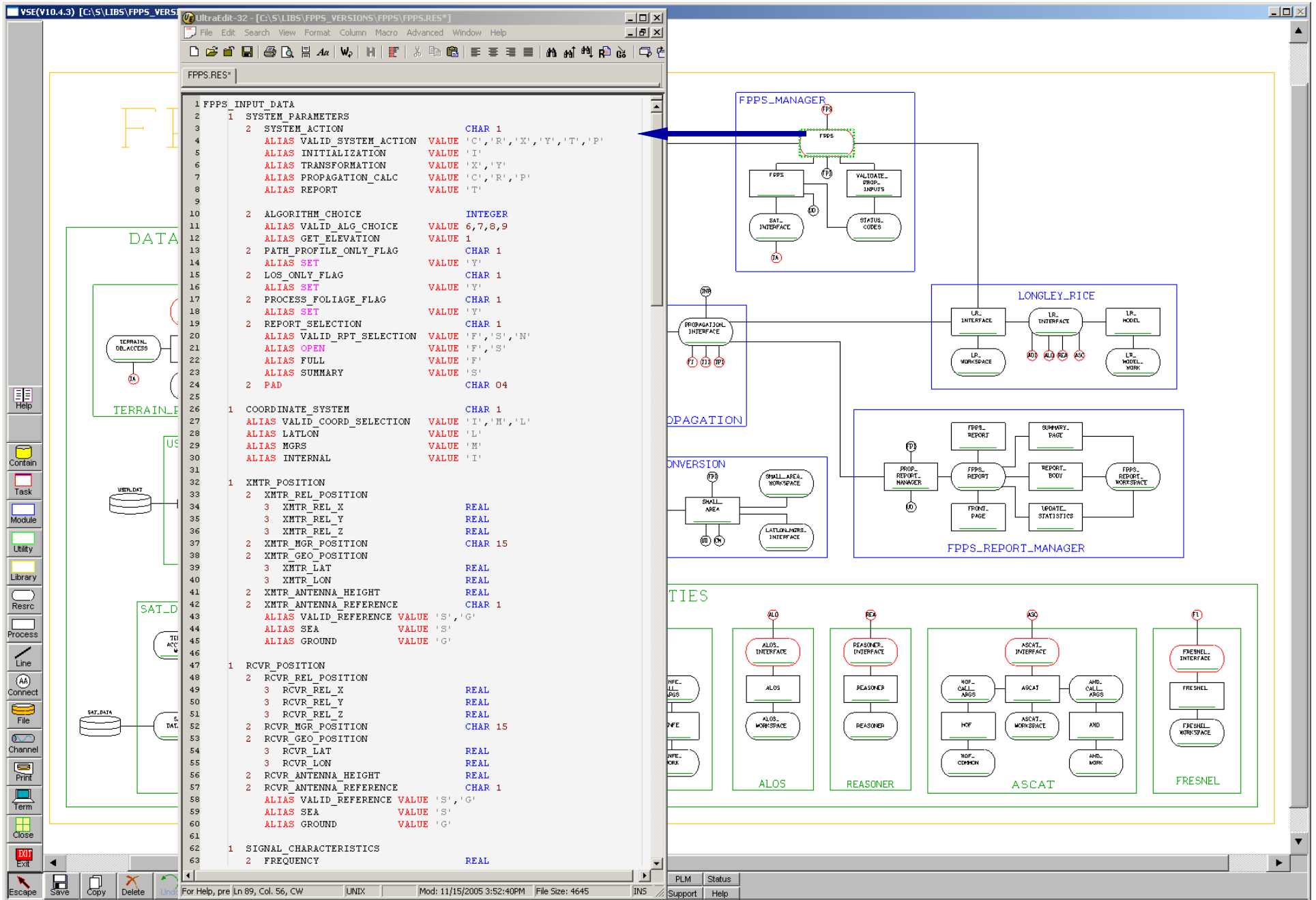


Figure 6. FPPS Library Module with aliased resource FPPS open.

The image displays the FPPS Library Module interface. On the left, a hierarchical tree structure shows the following sub-modules:

- DATA_UTILITIES** (containing TERRAIN_PROF_INTERFACE, TERRAIN_OB_ACCESS, TERRAIN_PROFILE, PROFILE_WORKSPACE)
- USER_DATA_INPUT** (containing USER_DATA_INTERFACE, USER_PARAMETERS, USER_DATA_WORKSPACE)
- SAT_DATA_INPUT** (containing TERRAIN_ACCESS_WORK, TERRAIN_ACCESS, SAT_DB_ACC, TERRAIN_ACCESS_SHARED, TERRAIN_DATA_ACCESS, FPPS_CH, TERRAIN_DATA_ACC_WORK)
- LONGLEY_RICE_UTILIT** (containing TERRAIN_IRREGULARITY_INTERFACE, TERRAIN_IRREGULARITY, TERRAIN_WORKSPACE, ADIFF_INTERFACE, ADIFF_ARGS, ADIFF_WORKSPACE, ADIFF_ARGS, ADIFF_WORK, ADIFF_COMMON)
- FRESNEL** (containing FRESNEL_INTERFACE, FRESNEL, FRESNEL_WORKSPACE)

The right pane shows the source code for the **PROP** process in UltraEdit-32. The code includes calculations for distance, elevation angle, and free space attenuation, along with logic for error handling and calling other sub-modules like **TERRAIN_PROFILE** and **ADIFF**.

```

180 DISTANCE = SQRT((XMTR_REL_X - RCVR_REL_X)**2
181             + (XMTR_REL_Y - RCVR_REL_Y)**2)
182 DISTANCE_Z = (END_TERRAIN_HEIGHT + END_ANTENNA_HEIGHT)
183             - (START_TERRAIN_HEIGHT + START_ANTENNA_HEIGHT)
184 DISTANCE_XYZ = SQRT(DISTANCE**2 + DISTANCE_Z**2)
185 ELEVATION_ANGLE = ATAN(DISTANCE_Z/DISTANCE)
186
187 FREE_SPACE_ATTENUATION = -27.55 + 20*LOG(FREQUENCY * DISTANCE_XYZ)
188
189 IF DISTANCE_XYZ IS EQUAL TO 0
190     ERROR_CODE = ERR_SAME_COORD
191     ***DISPLAY 'DISTANCE_XYZ ',DISTANCE_XYZ
192 ELSE
193 IF DISTANCE_XYZ IS LESS THAN 1000
194 OR DISTANCE_XYZ IS GREATER THAN MAX_FPPS_DISTANCE
195 THEN
196     WARNING = WRN_LR_MODEL
197 ELSE
198 IF FREE_SPACE_ATTENUATION IS GREATER THAN MAX_FPPS_LOSS
199     ERROR_CODE = NET_EXCEEDED_MAX_LOSS .
200
201 *****
202 BUILD TERRAIN_PROFILE
203 IF ERROR_CODE IS NOT UNDETECTED
204     EXIT THIS RULE .
205
206 MOVE LOW_VALUES TO PROFILE_DATA
207 MOVE PROCESS_FOLIAGE_FLAG TO FPPS_FOLIAGE_INCLUDED
208 MOVE LOS_ONLY_FLAG TO FPPS_LOS_ONLY
209 FPPS_DISTANCE = DISTANCE
210 STA_TERRAIN_GRID_SIZE = DBASE_SCALE
211
212 CALL TERRAIN_PROFILE
213
214 IF ERROR_CODE IS UNDETECTED
215     MOVE PROFILE_DATA TO PATH_PROFILE_DATA .
216
217 *****
218 INVOKE FOLIAGE_ANTENNA
219 START_ANTENNA_HEIGHT = XMTR_ANTENNA_HEIGHT
220 END_ANTENNA_HEIGHT = RCVR_ANTENNA_HEIGHT
221 MOVE XMTR_ANTENNA_REFERENCE TO START_ANTENNA_REFERENCE
222 MOVE RCVR_ANTENNA_REFERENCE TO END_ANTENNA_REFERENCE
223
224 START_FOLIAGE_DISTANCE = 0.0
225 END_FOLIAGE_DISTANCE = 0.0
226
227 CALL ANTENNA_FOLIAGE_CHECK
228     IN FOLIAGE_FPPSLIB
229     USING PROPAGATION_INTERFACE PROP_DB_ACCESS
230
231 *****
232 INVOKE FOLIAGE_MODEL
233 FOLIAGE_GRID = FOLIAGE_GRID_SIZE
234 FOLIAGE_ELEV = FOLIAGE_HEIGHT
235
236 FPPS_FREQUENCY = FREQUENCY
237
238 CALL FOLIAGE_MODEL
239     IN FOLIAGE_FPPSLIB
240     USING PROPAGATION_INTERFACE
241
242 FOLIAGE_ATTENUATION = FOLIAGE_LOSS
  
```

Figure 7. FPPS Library Module with process PROP open.