Command Management

Theresa Brandt

18 October 1995

CMS Hardware Mapping

Real Time Server

DMS
- Archiver
- Data Retriever
- Event Listener
- Event Handler
- Event Retriever

Analysis
- Clock Correlation
- RT Statistics
- SSR

COTS
- RTWorks

Data Server

CMS
- Schedule Controller
- Ground Schedule
- Load Catalog
- Spacecraft Model
- Command Model

DMS
- Queue Manager
- Playback Merge
- Event Processing
- Disk Cleaner
- SDPS I/F
- Data Retriever
- File Manager
- PDB Input
- PDB Edit
- PDB Reporting
- OD Generation

COTS
- Sybase
- DBTools

User Station

DMS
- Data Retriever
- Event Listener
- Event Handler
- Event Retrieve

Analysis
- Request Manager
- Offline Analysis
- DSS Advisor
- DSS DataServer

COTS
- Sybase
- DBTools
- RTWorks
- DBX
- IMSL
- Altair

IST

DMS
- Data Retriever
- Event Listener
- Event Handler
- Event Retriever

Analysis
- Request Manager
- Offline Analysis

COTS
- Sybase
- DBTools
- DBX
- IMSL

CMS Overview

Schedule Controller Process

- **Processes activity lists**

Command Model Process

- **Rule-based constraint**

Spacecraft Model Process

- **ATC buffer model**
- **RTCS buffer model**
- **Table model**
- **Ground images**
- **Memory dump processing**

Ground Schedule Process

CMS Overview

Load Catalog Process

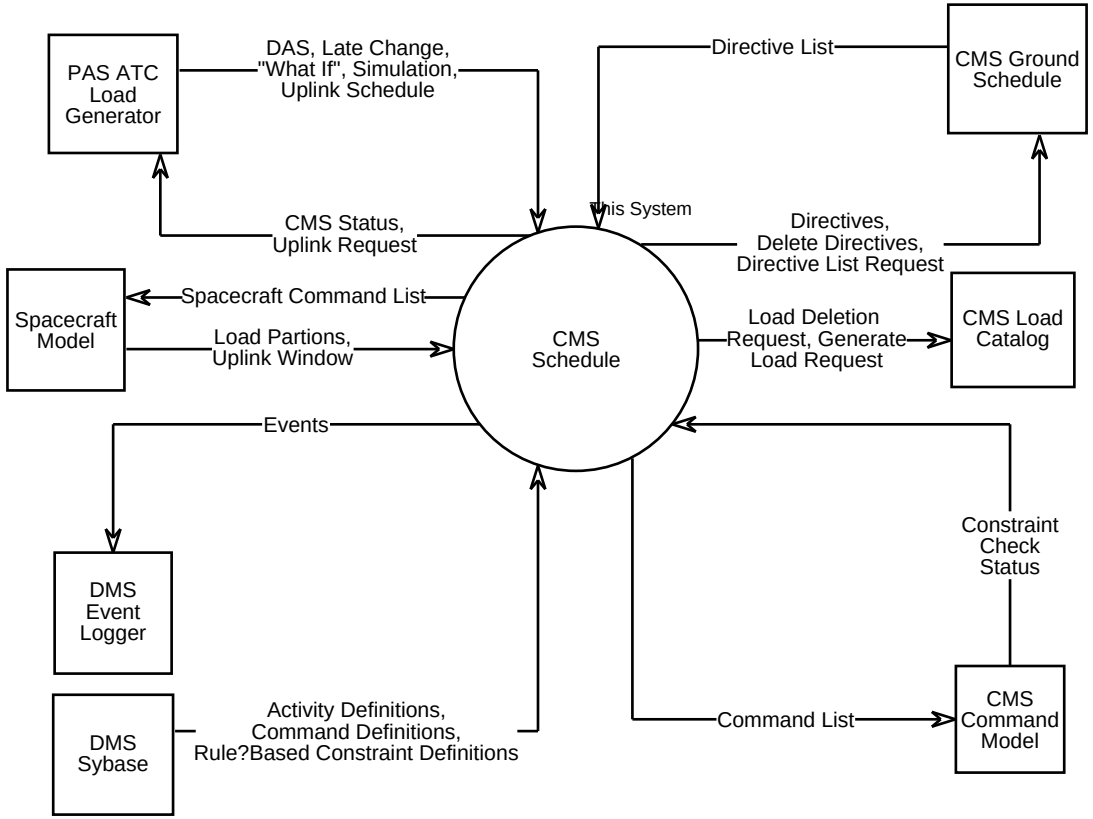
- **Load catalog**
- **Microprocessor loads**
- **RTCS loads**
- **FSW loads**
- **Table loads**

Schedule Controller Process

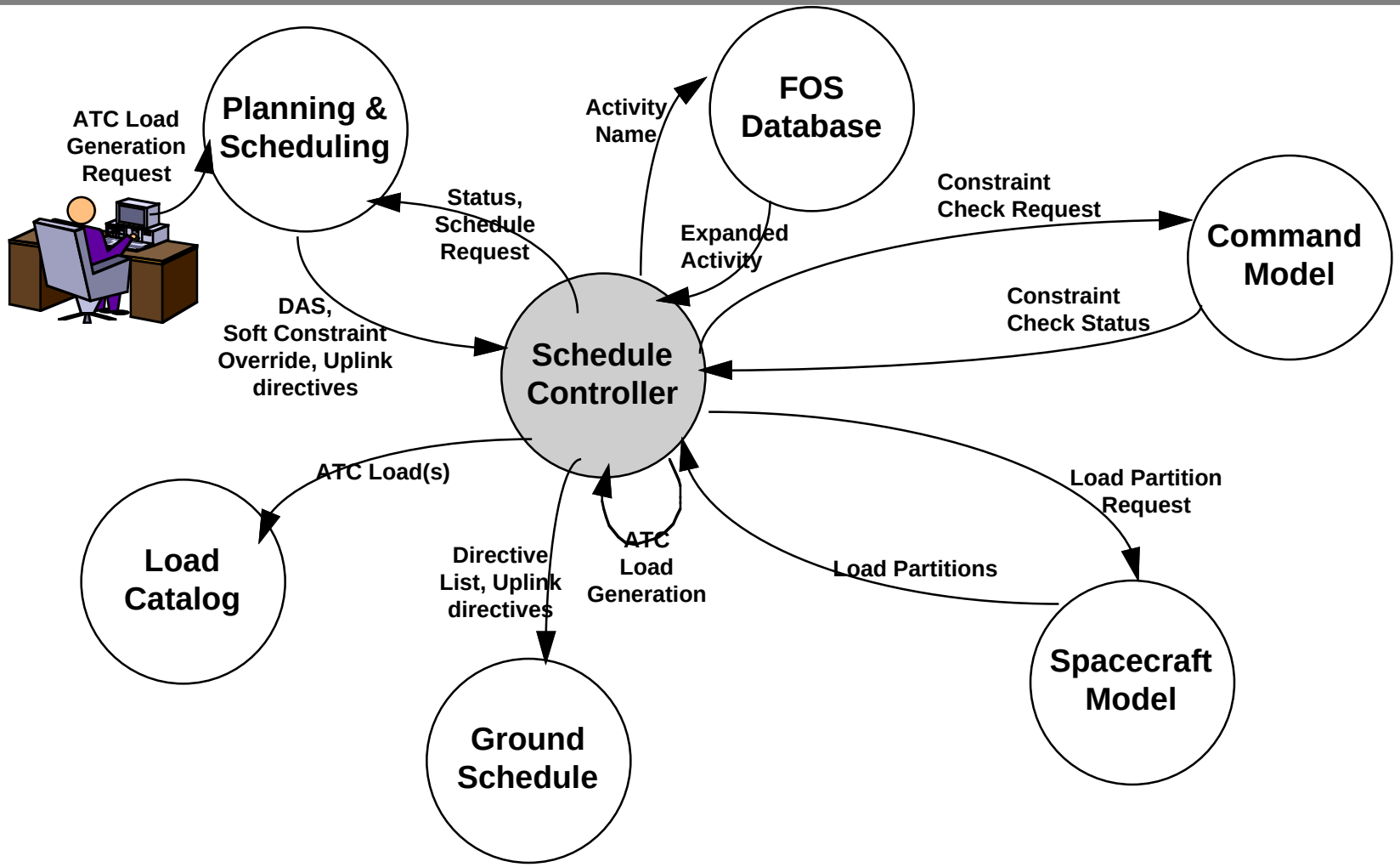
Process Description

- **Receives request to process an activity list from Planning and Scheduling (PAS)**
- **Manages processing of "What If" schedule**
  - **Expands activities**
  - **Requests constraint checking**
- **Manages processing of Detailed Activity Schedule (DAS) or late change**
  - **Expands activities into directive list**
  - **Requests constraint checking**
  - **Requests ATC buffer mapping and load generation**
  - **Requests updates to Ground Schedule**
- **Responds to PAS's request to process DAS by returning a status**

Schedule Controller Context Diagram



ATC Load Generation



Schedule Controller Process

Key Requirements

- **ATC load shall be generated from the same operational period as the Detailed Activity Schedule (DAS)**
- **Ground script shall be based on DAS**
- **Allow user to override soft constraint violations**

Key Features

- **Directive object shared with FUI**
  - **Each type of directive is a separate class**
    - **Simple to implement addition of new directive types**
    - **Special directive processing is isolated**
    - **Directive syntax change affects only one class**
- **Automatically requests uplink scheduling for all ATC loads**
  - **Improves reliability**

Schedule Controller Process

Design Implementation

- **Receive time ordered lists of activities which represents DAS**
  - **Ensures that ATC load and ground schedule will be based on the same operational period as DAS**
- **Expand Activities**
  - **Allow parameter substitution into expanded directives**
  - **Retrieve activity definition, command definitions and command constraint definitions from one database view (Sybase)**
  - **Build an integrated list of directives from ground commands, preplanned R/T spacecraft commands, and stored commands from the DAS**

    Provides unprecedented coordination between spacecraft and ground system

Activity Expansion

Activities

Green Activity	1:24 am	Green Command 1
	2:45 am	Green Command 2
	4:00 am	Green Command 3
Red Activity		Red Command 1
Blue Activity		Blue Command 1
Yellow Activity		

Time Ordered Expanded Command List

1:24 am	Green Command 1
2:45 am	Green Command 2
3:00 am	Red Command 1
3:30 am	Blue Command 1
4:00 am	Green Command 3
5:00 am	Red Command 2
5:30 am	Blue Command 2
6:00 am	Red Command 3
10:00 am	Yellow Command 1
11:00 am	Yellow Command 2
1:30 pm	Blue Command 3
4:30 pm	Blue Command 4
	Yellow Command 3

Schedule Controller Process

Design Implementation (cont.)

- **Constraint Processing**
  - **No constraint violations: processing of ATC load generation and updating ground schedule continue**
  - **Hard constraint violations cause processing to terminate: ATC load is not generated and ground schedule is not updated**
  - **If only soft constraint violations were found, processing is suspended until user response is received**

    Override response: processing continues

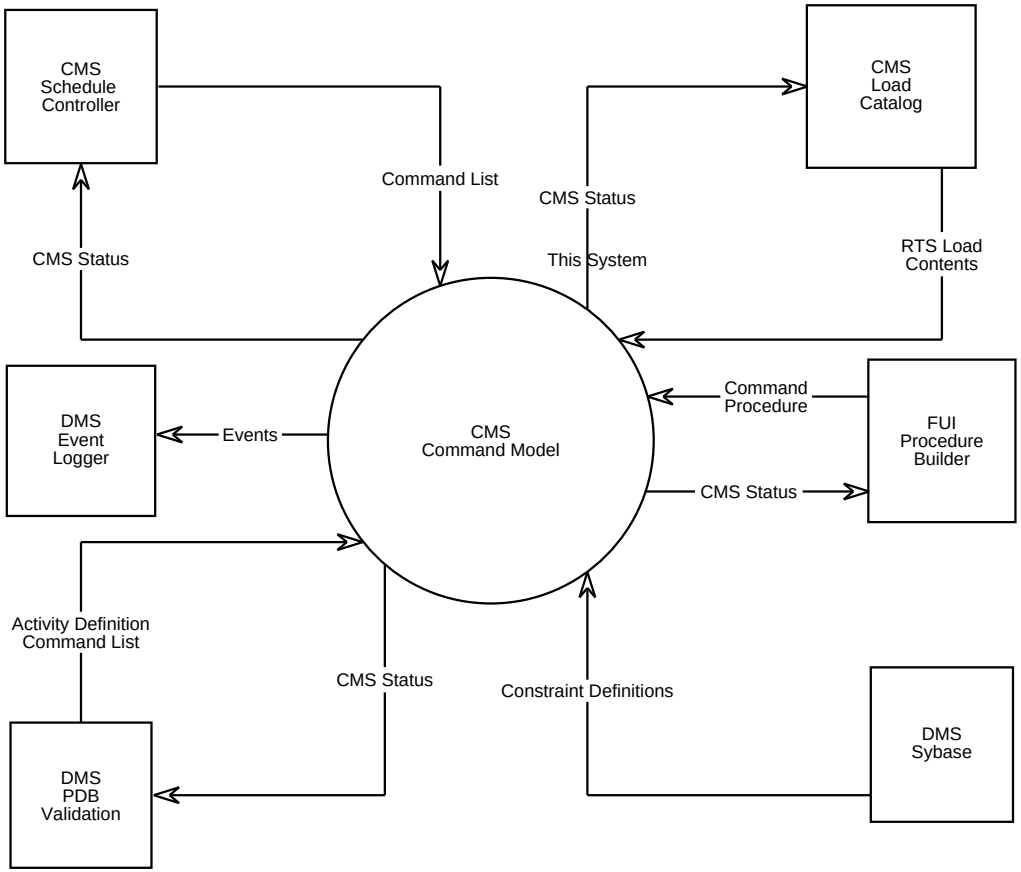
    Don't override response: processing is terminated
- **Uplink schedule request**

Command Model Process

Process Description

- **Receives constraint check requests**
  - **Command list from CMS Schedule Controller**
  - **Procedure definition from FUI Procedure Builder**
  - **RTCS load definitions from CMS Load Catalog**
  - **Activity definition list from DMS PDB Validation**
- **Returns status**
  - **Complete, no violations**
  - **Pending, only soft constraint violations found**
  - **Failed, at least one hard constraint violation**

Command Model Context Diagram



Command Model Process

Key Requirements

- **ATC commands, ground commands, RTCS loads and procedure definitions shall be constraint checked**

Key Analyses

- **Rule selection based on study of Wind/Polar CMS and PARR constraints**
- **PARR associated activity constraint definitions with the activities**

Command Constraint Rules

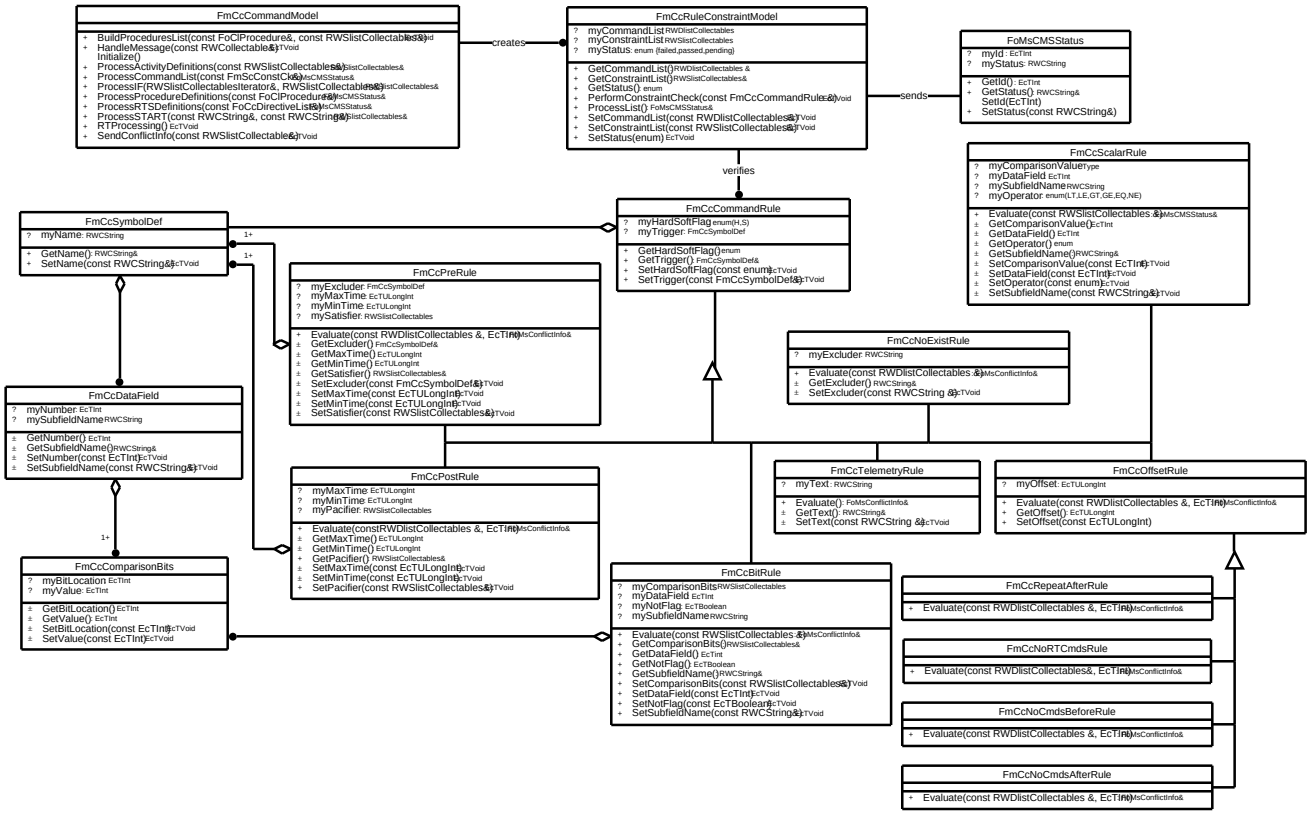
Command Rule Type	Definition
Pre-Rule	If Command A Occurs, Then Command B Must Have Occurred at Least X Time Earlier, And At Most Y Time Earlier. Command C (optional) Must Not Occur Between B and A
Post-Rule	If Command A Occurs, Then Command B Must Occur And Must Be At Least X Time Later, And At Most Y Time Later
Telemetry Rule	If Command A Occurs, Print an FOT-supplied Rule - Specific Text String in the Conflict Messages
No Exist Rule	If Command A Occurs, Then Command B Must Not Occur
Bit Rule	If Command A Occurs, One or More Bits in the Specified Data Fields Must Be (or Must NOT Be) set to Specified Values
Scalar Rule	If Command A Occurs, The Value Of A Particular Data Field Must Be Less Than, Greater Than, Equal To, Less Than OR Equal To, Greater Than Or Equal To, OR Not Equal to X
No Commands Before Rule	If Command A Then NO Commanding at Least X seconds before Command A
No Commands After Rule	If Command A Then No Commanding At Least X Seconds After Command A
No Commands to RT Until Rule	If Command A Is Sent To RT, Then No Commanding to That RT at Least X Seconds
No Repeat Rule	If Command A, Then Command A Must Not Be Repeated Before X Seconds

Command Model Process

Key Design Features

- **Reduce probability of constraint violations during ATC load generation**
  - **RTCS definitions, procedures and activity definitions are validated when defined**
- **Allows for multiple directive lists to be constraint checked independently**
  - **Spawns the rule constraint model process**
- **Constraint definitions are database defined and easily modified**
- **Addition of new rule types easily implemented**

Rule Constraint Object Model



Command Model Process

Design Implementation

- **Command model builds or receives a doubly linked command list for constraint checking**
  - **Forward/backward searching for rule satisfiers**

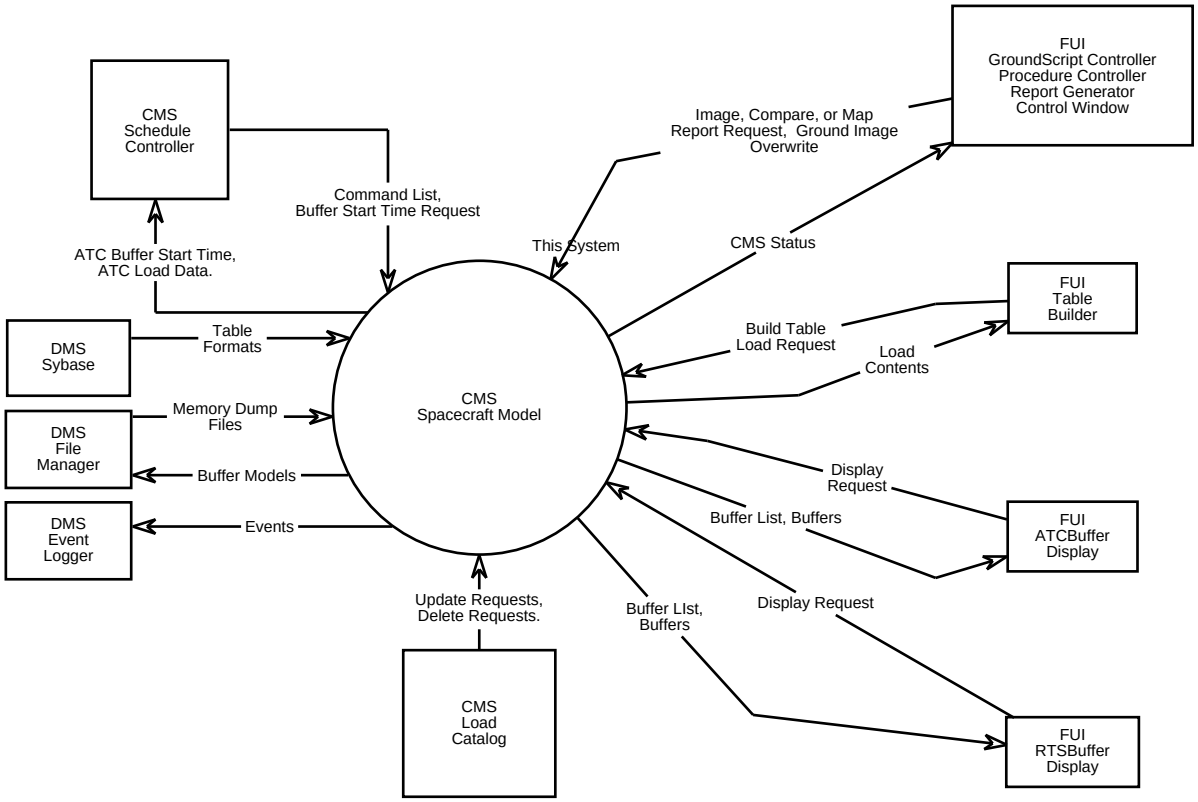
    Improves performance
- **Command constraints are associated with the command, i.e. they are included as part of the command directive object**
  - **Improves performance**
- **For each command that has a constraint, the constraint object performs the evaluation to determine if there is a violation**
- **Each conflict is put into a conflict list that is returned when the constraint checking is completed**

Spacecraft Model Process Description

Creates and Maintains

- **ATC buffer model**
- **RTCS buffer models**
- **Table models**
- **Ground images & memory dump images**

Spacecraft Model Process Context Diagram



Spacecraft Model Process

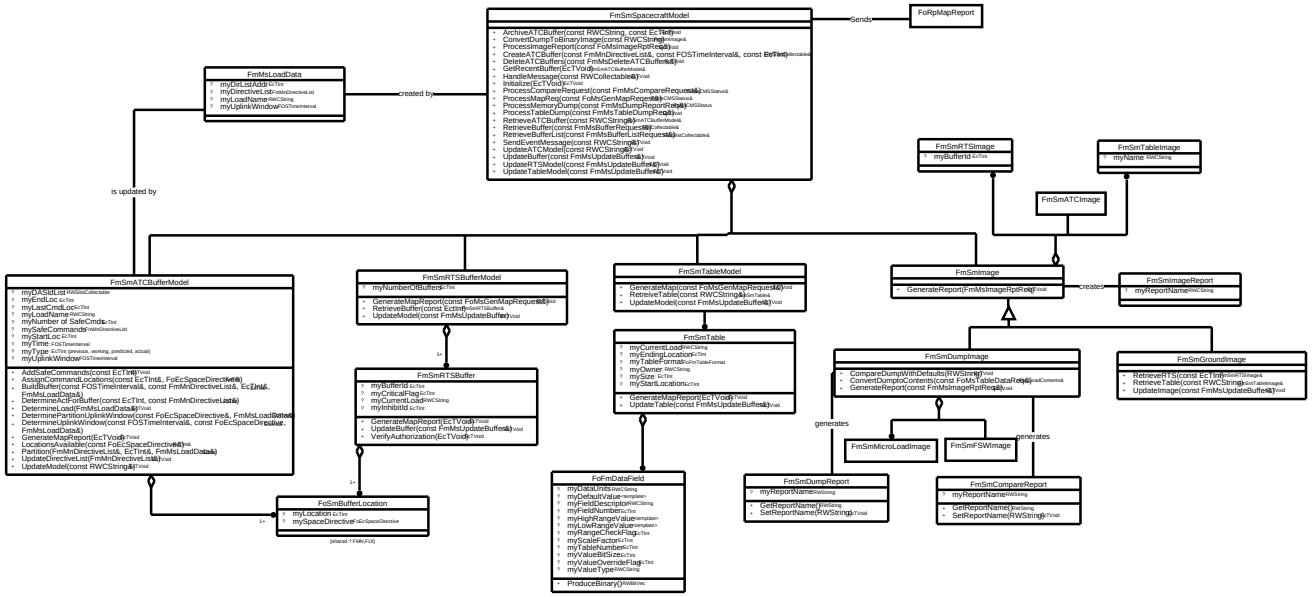
Key Requirements

- **Maintain ATC and RTCS command-to-memory maps**
- **Maintain table load maps**
- **Maintain ground and memory dump images**

Key Design Features

- **Provides reliability, error-free operations for ATC load processing**
  - **Automatically determines ATC load partitioning**
  - **Automatically determines ATC uplink Window**
  - **Ensures continuity between ATC loads**
- **Mission specific processing is isolated**
- **Simple to add or delete model or image for future missions**

Spacecraft Model Object Model



Spacecraft Model

Key Features (cont.)

- **Table models are database defined**
  - **Adding new model requires simple addition of database definitions**
- **Table dump conversion**
  - **Dumps are reverse engineered into table load contents, allowing the user to edit the dump and create a new table load**

Design Implementation

- **Ground images are updated after uplink notification or upon user request to overwrite the ground image**
- **Dump Conversion**
  - **Retrieves dump file from DMS**
  - **Reads the header record to determine dump type and/or format, number of words, and other identifying information**
  - **Strips CCSDS Header: results in binary image**

Spacecraft Model

Design Implementation (cont.)

- **Actual buffer models are updated after uplink notification**
- **Maintains numerous ATC buffers: actual, predicted, working, and previous**
  - **In the event of a late change the affected predicted buffer models are deleted**

ATC Buffer Model

Design Implementation

- **When command list is received from the schedule controller, retrieves the most recent predicted buffer and copies it to a working buffer**
- **Determines the uplink window based on user request**
  - **If ATC partition load, uplink window is automatically set after previous partition uplink window**
- **Calculates the number of available locations in the buffer, excluding safe commands**
- **Determines activity boundaries and maps commands into buffer**
  - **Maintain a list of activities currently being mapped into buffer**
  - **Reduces number of available locations by number of space directives in new “unmapped” activity**

Activity Expansion

Activities

Green Activity	1:24 am	Green Command 1
	2:45 am	Green Command 2
	4:00 am	Green Command 3
Red Activity		Red Command 1
Blue Activity		Blue Command 1
Yellow Activity		

Time Ordered Expanded Command List

1:24 am	Green Command 1
2:45 am	Green Command 2
3:00 am	Red Command 1
3:30 am	Blue Command 1
4:00 am	Green Command 3
5:00 am	Red Command 2
5:30 am	Blue Command 2
6:00 am	Red Command 3
10:00 am	Yellow Command 1
11:00 am	Yellow Command 2
1:30 pm	Blue Command 3
4:30 pm	Blue Command 4
	Yellow Command 3

ATC Buffer Prior to Uplink

Current ATC Buffer

Memory Loc.		
0001	4:00 am Day 1	Executed Command
0002	4:30 am Day 1	Executed Command
0003	5:00 am Day 1	Executed Command
0004	5:30 am Day 1	Executed Command
0005	6:00 am Day 1	Executed Command
0006	6:30 am Day 1	Executed Command
0007	7:00 am Day 1	Executed Command
0008	8:00 am Day 1	Executed Command
0009	9:00 am Day 1	Executed Command
0010	9:30 am Day 1	Executed Command
0011	10:00 am Day 1	Executed Command
0012	11:00 am Day 1	Executed Command
0013	1:30 pm Day 1	Executed Command
0014	6:30 pm Day 1	Unexecuted Command
0015	6:30 pm Day 1	Unexecuted Command
0016	2:00 am Day 2	Unexecuted Command
0017	3:45 am Day 2	Unexecuted Command
0018	4:00 am Day 2	Safe Command 1
0019	4:00 am Day 2	Safe Command 2

Uplink 4:00 pm Day 1 → 0014

ATC Load

Time Ordered Expanded Command List

1:24 am	Green Command 1
2:45 am	Green Command 2
3:00 am	Red Command 1
3:30 am	Blue Command 1
4:00 am	Green Command 3
5:00 am	Red Command 2
5:30 am	Blue Command 2
6:00 am	Red Command 3
10:00 am	Yellow Command 1
11:00 am	
1:30 pm	Blue Command 3
4:30 pm	Blue Command 4

ATC Load

1:24 am Day 2	Green Command 1
2:00 am Day 2	Existing Command in Buffer
2:45 am Day 2	Green Command 2
3:00 am Day 2	Red Command 1
3:30 am Day 2	Blue Command 1
3:45 am Day 2	Existing Command in Buffer
4:00 am Day 2	Green Command 3
5:00 am Day 2	Red Command 2
5:30 am Day 2	Blue Command 2
6:00 am Day 2	Red Command 3
10:00 am Day 2	Yellow Command 1
11:00 am Day 2	Yellow Command 2
1:30 pm Day 2	Blue Command 3
4:30 pm Day 2	Blue Command 4
6:00 pm Day 2	Yellow Command 3
6:15 pm Day 2	Safe Command 1
6:15 pm Day 2	Safe Command 2

ATC Buffer After Load Uplink

ATC Buffer After Uplink

Memory Loc.		
0001	3:30 am Day 2	Blue Command 1
0002	3:45 am Day 2	Existing Command in Buffer
0003	4:00 am Day 2	Green Command 3
0004	5:00 am Day 2	Red Command 2
0005	5:30 am Day 2	Blue Command 2
0006	6:00 am Day 2	Red Command 3
0007	10:00 am Day 2	Yellow Command 1
0008	11:00 am Day 2	Yellow Command 2
0009	1:30 pm Day 2	Blue Command 3
0010	4:30 pm Day 2	Blue Command 4
0011	6:00 pm Day 2	Yellow Command 3
0012	6:15 pm Day 2	Safe Command 1
0013	6:15 pm Day 2	Safe Command 2
0014	6:30 pm Day 1	Unexecuted Command
0015	6:30 pm Day 1	Unexecuted Command
0016	1:24 am Day 2	Green Command 1
0017	2:00 am Day 2	Existing Command in Buffer
0018	2:45 am Day 2	Green Command 2
0019	3:00 am Day 2	Red Command 1

Load Start

ATC Buffer Model

Design Implementation (cont.)

- When number of space directives in “unmapped” activity exceeds the number of available locations, the buffer is full and the directive list is marked for partitioning
- Remaining commands in activities currently being mapped are added to buffer
- Safe commands are added to buffer

• Builds command list for generating the ATC load, including safe commands

Ground Schedule Process

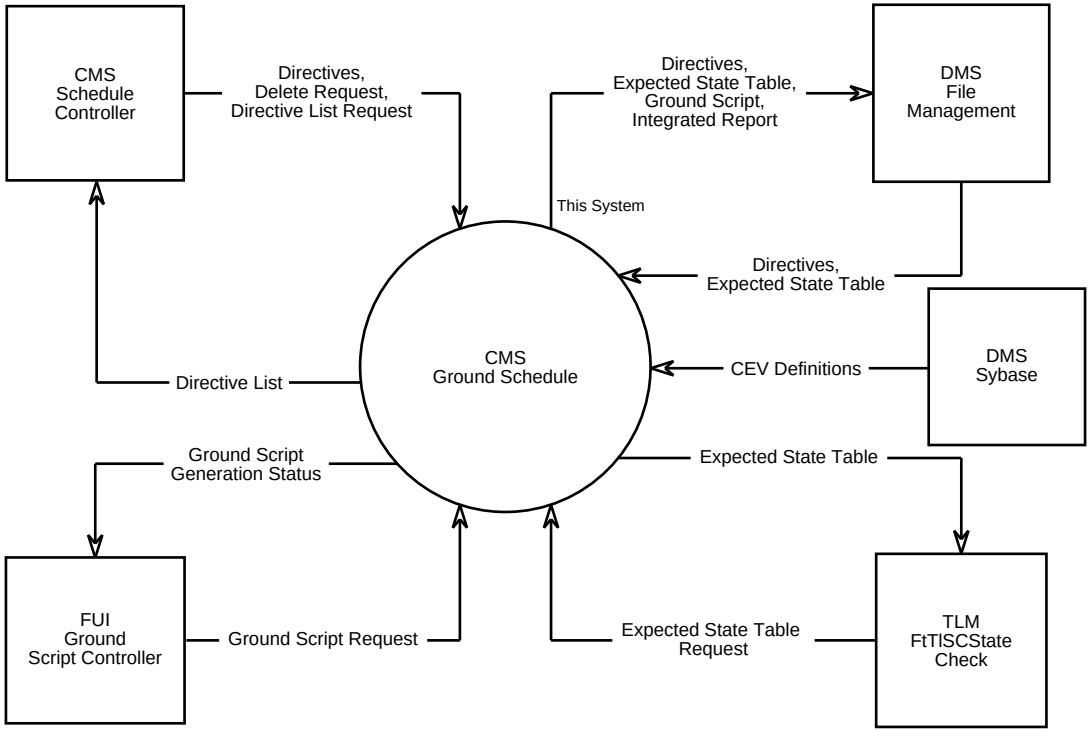
Process Description

- **Updates ground schedule**
- **Creates ground script**
- **Generates expected state table**
- **Generates integrated report**

Key Requirements

- **Produce planned state of spacecraft for command verification telemetry parameters**
- **Produce integrated report**
- **Include stored commands in ground script**
  - **Comments for display to FOT**
  - **During contacts stored commands are verified through telemetry**
- **Ground schedule includes orbital events, TDRS contacts, and all preplanned directives expanded from DAS activities**

Ground Schedule Context Diagram



Load Catalog Process

Process Description

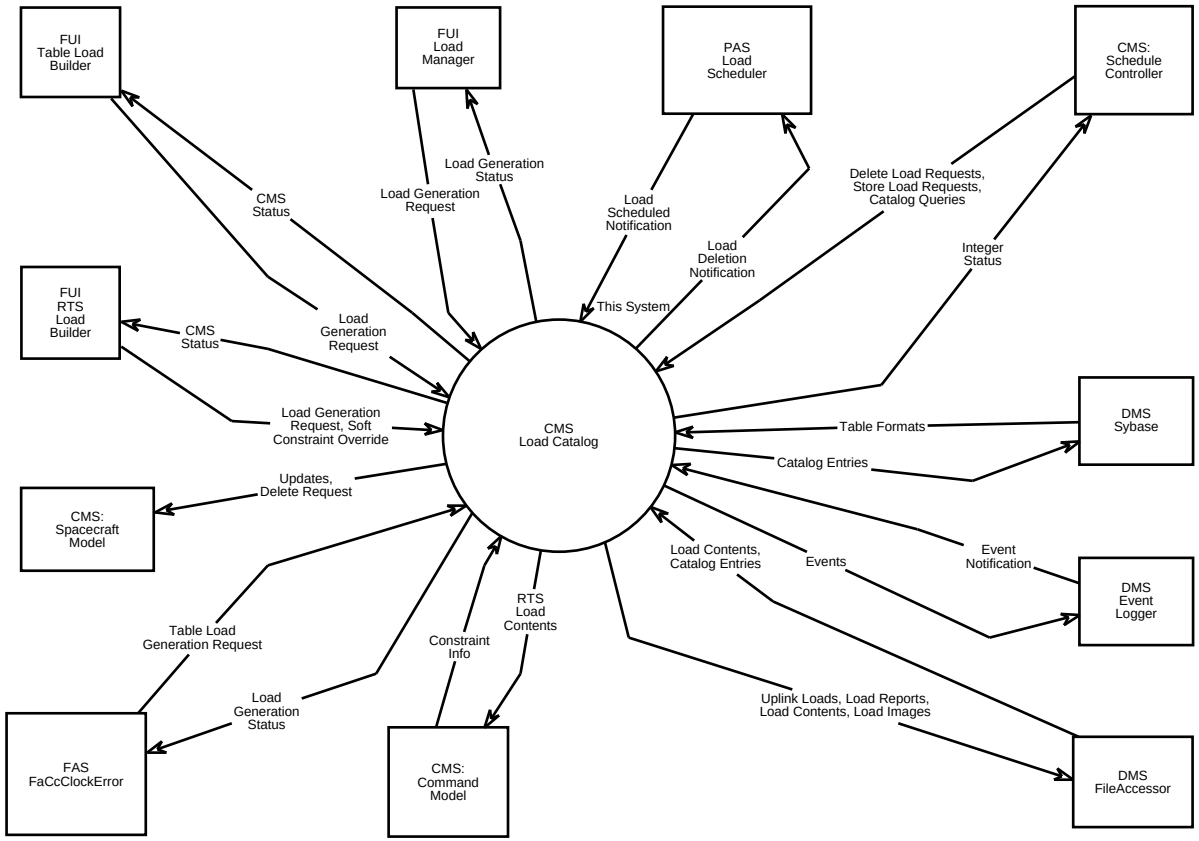
- **Maintains EOC load catalog**
  - **List of all valid loads available for uplink from EOC**

    ATC, RTCS, FSW, microprocessor, and table
- **Generates RTCS, FSW, microprocessor, and table loads**
- **Stores load with DMS**

Key Requirements

- **Maintain catalog of RTCS, FSW, microprocessor and table loads existing in EOC**
- **SCC loads must be partitioned into 4K bytes**

Load Catalog Context Diagram



Load Catalog Process

Key Design Features

- **Catalog entry directly accesses FOS database**
- **Catalog entry contains identifying information about the load**
  - **Load type, instrument or spacecraft subsystem, size of load, number of 4K pieces, filename and uplink time for each piece, valid uplink period, scheduling info, source, destination, and criticality indicator**
- **Catalog entry available to user via FUI and R/T command process which retrieves load for uplink from load catalog**
- **Notifies spacecraft model**
  - **Uplink notification**
  - **Late change deletion of predicted ATC buffer models**
- **Table load structures are defined in the database**
- **RTCS load validation allows user to override soft constraints**
- **Addition of new load types easily implemented**

Load Types Object Model

Load Manager Tool

Features

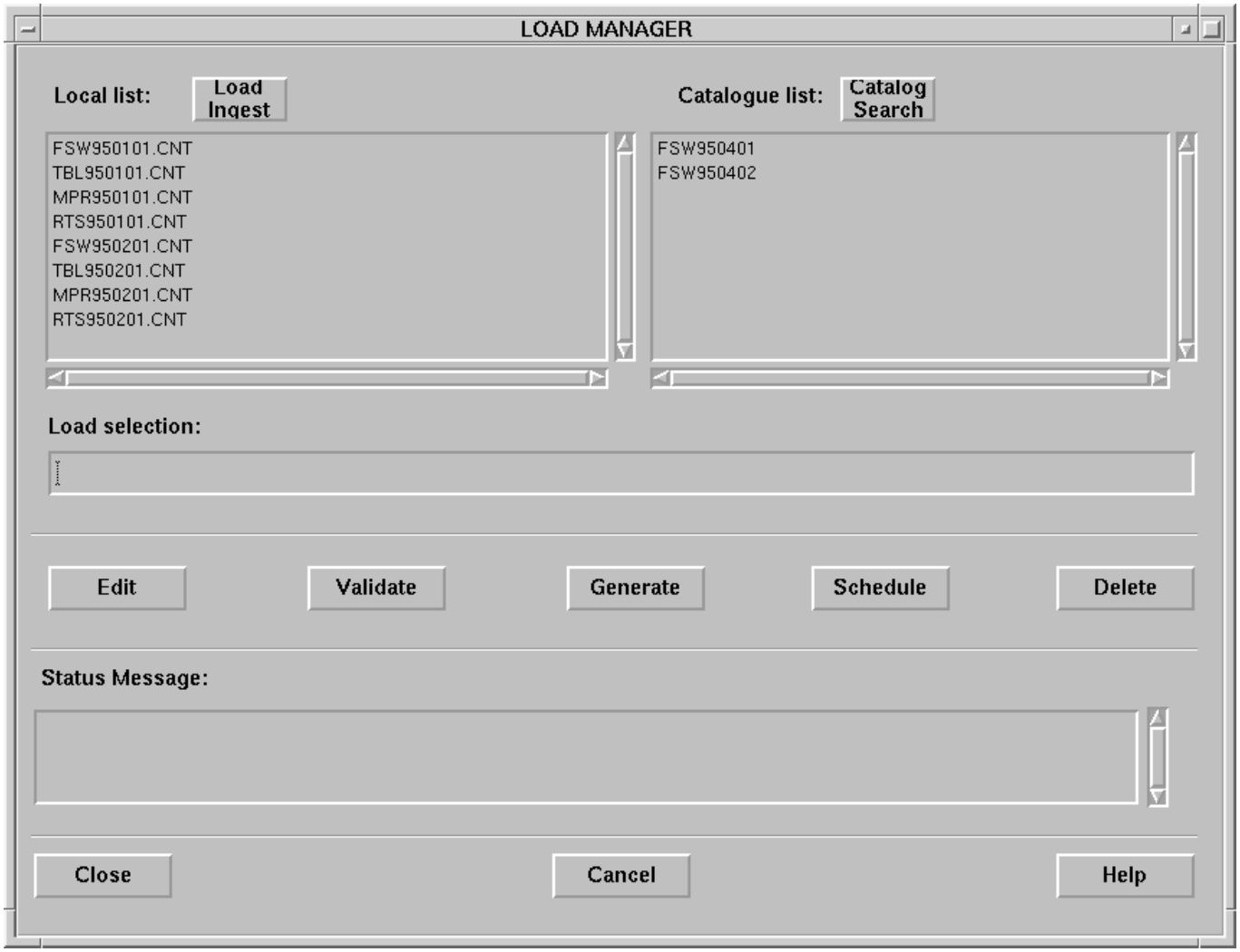
- **Provides access to RTCS load and table load builders**
- **Provides capability to validate and generate loads**
- **Provides capability to schedule load uplinks**
- **Ingests external load contents**
- **Displays load catalog entries**
- **Provides capability to delete valid loads**

Interfaces with FUI load builders, CMS load catalog, and PAS load scheduler

Design Implementation

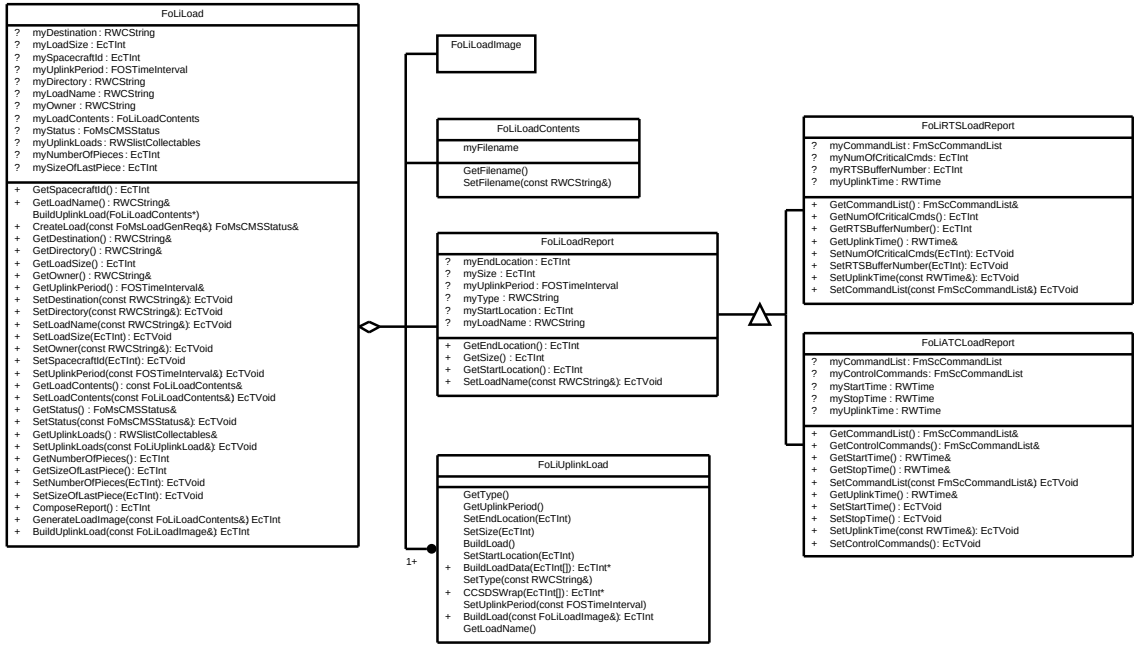
- **Table and RTCS load contents are in ASCII format**
  - **Edit, validate, generate and schedule**
- **FSW and microprocessor load contents are in binary format**
  - **Generate and schedule**

Load Manager Tool



Load Generation

Design Implementation

- **RTCS load generation**
  - **User selects or edits load contents**
  - **User selects generate which automatically invokes validation**
  - **Load manager tool sends generation request to CMS**
  - **CMS performs constraint checking of RTCS load contents**
  - **Validated load contents used by load object**
    - **Creates packetized binary uplink load**
    - **Creates binary load image for use in spacecraft model**
    - **Creates load report**
  - **Create catalog entry for new RTCS load**
  - **Store all load objects with DMS**
  - **User requests load uplink scheduling**

Load PaRTCS Object Model