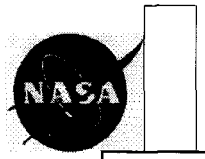



JPL

# Multi-Mission Sequencing Software

L. Needels

Jet Propulsion Laboratory

California Institute of Technology

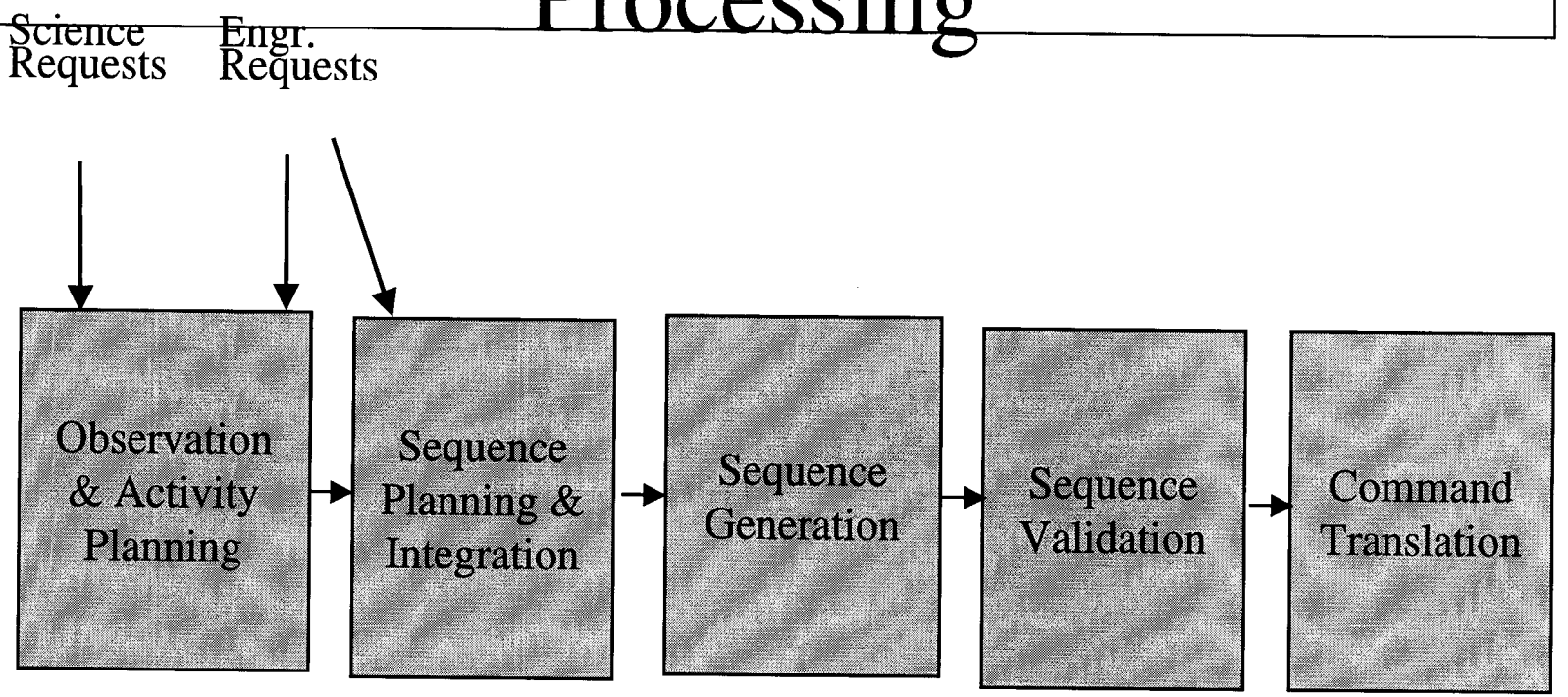
# Agenda

- Description of the sequencing process
- Overview of current sequencing architecture
- Examples of Core/Adaptation split
- Multi-Mission aspects of the Adaptation
- Advantages and disadvantages of this architecture

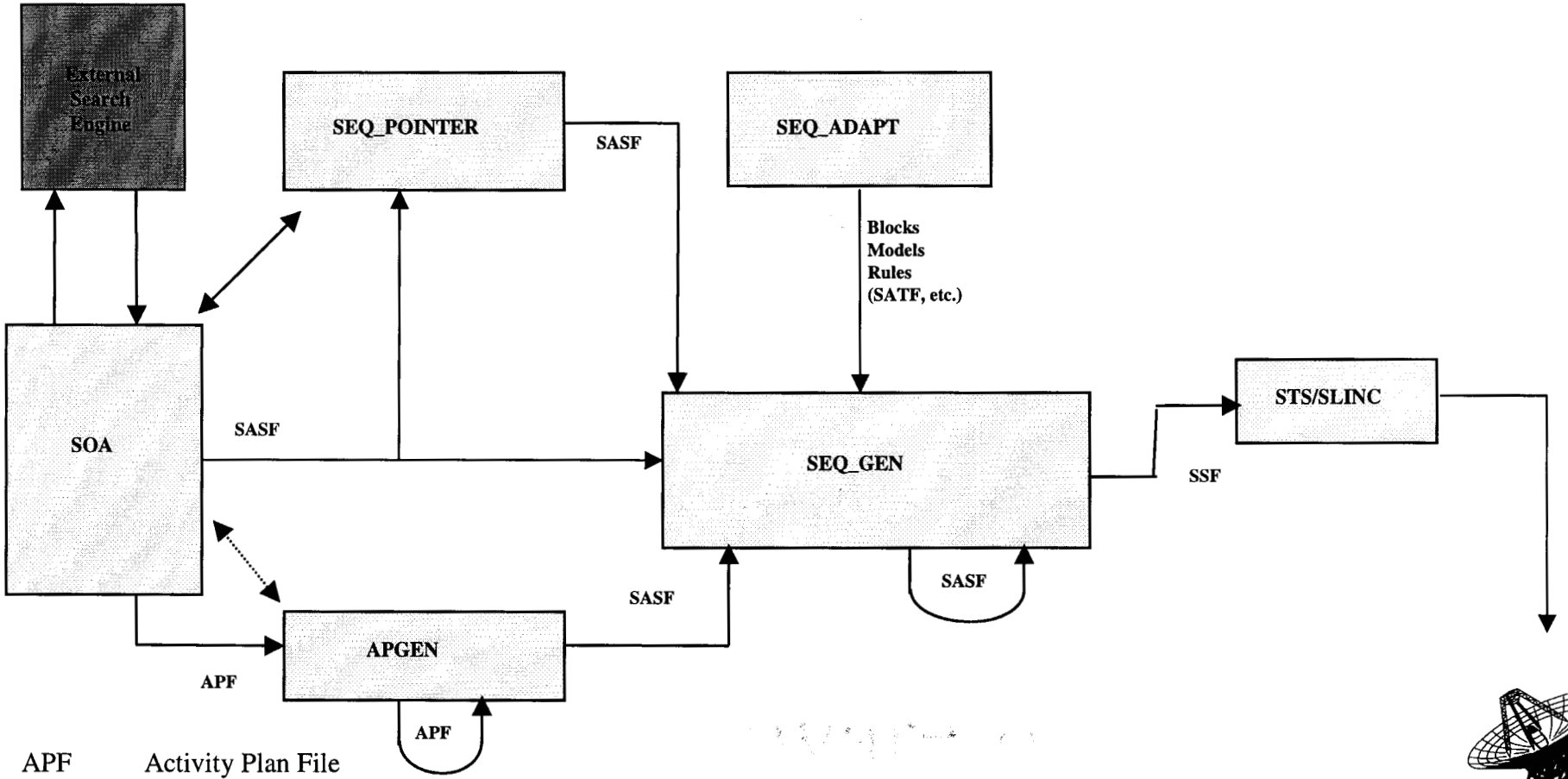
# Sequencing Process

- The spacecraft commands are generated using the following steps:
  - Planning and creation of science and engineering activities
  - Command syntax checking
  - Mission and Flight Rule Checking
  - Command translation
  - Uplink

# Functional Steps of Sequence Processing

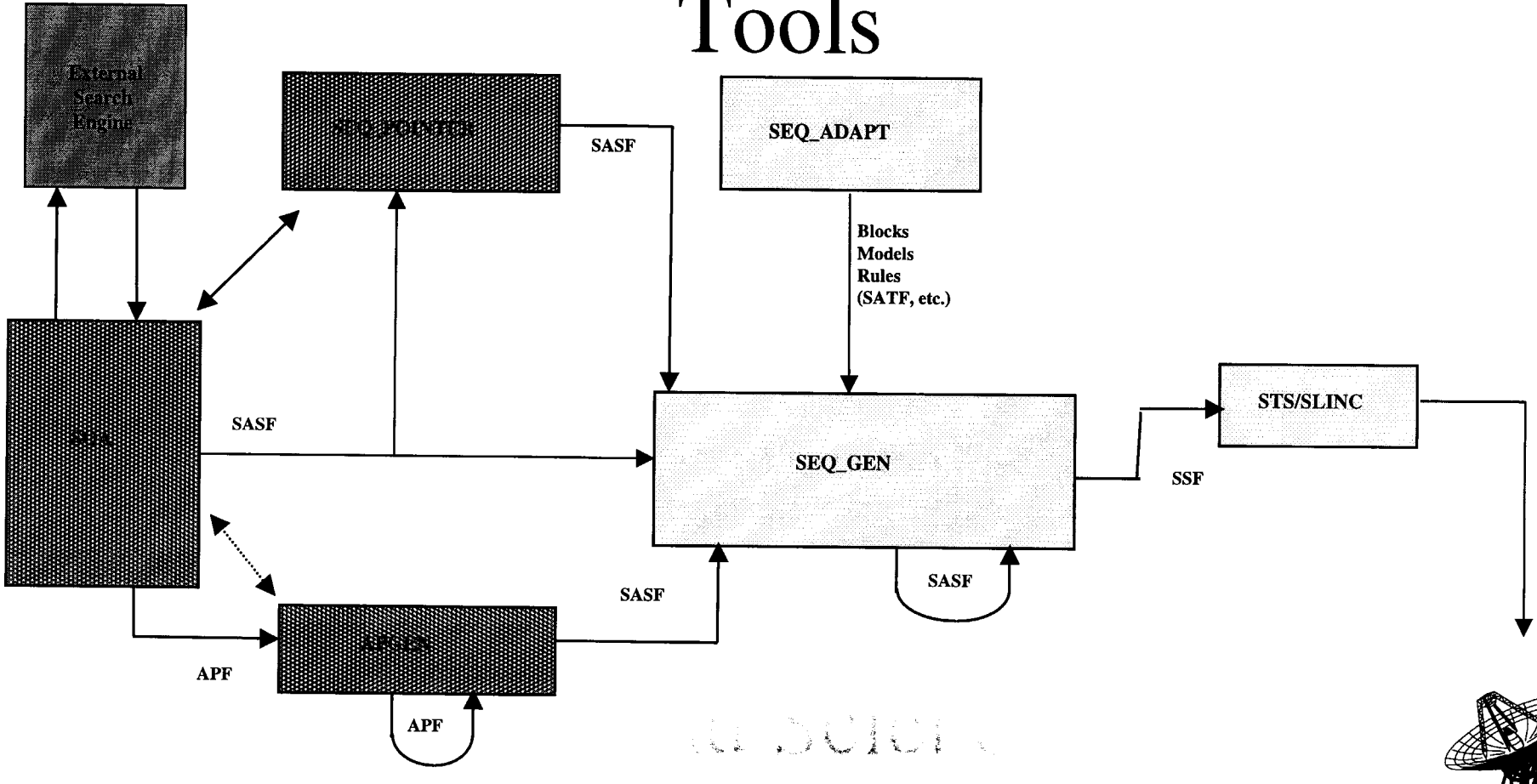


# SEQ Software Flow




| | |
|---|---|
| APF | Activity Plan File |
| PEF | Predicted Events File |
| SASF | Spacecraft Activity Sequence File |
| SATF | Spacecraft Activity Type File |
| PEF | Spacecraft Sequence File |
| SSF | Spacecraft Sequence File |

# Engineering and Science Planning Tools




JPL

# Engineering and Science Planning Tools

- Science Opportunity Analyzer (SOA) is an observation planning tool that works at the activity level
  - Interfaces with external search engines that will locate times of interest (flybys, bow-shocks, occultations, periapsis, etc.)
  - Design of a specific observation (Continuous Scan, Roll Scan, Start/Stop Mosaic, Stare, etc.)
  - Constraint Checking (activity duration, distance, exclusion zones, hardware limits, etc.)

# Engineering and Science Planning Tools

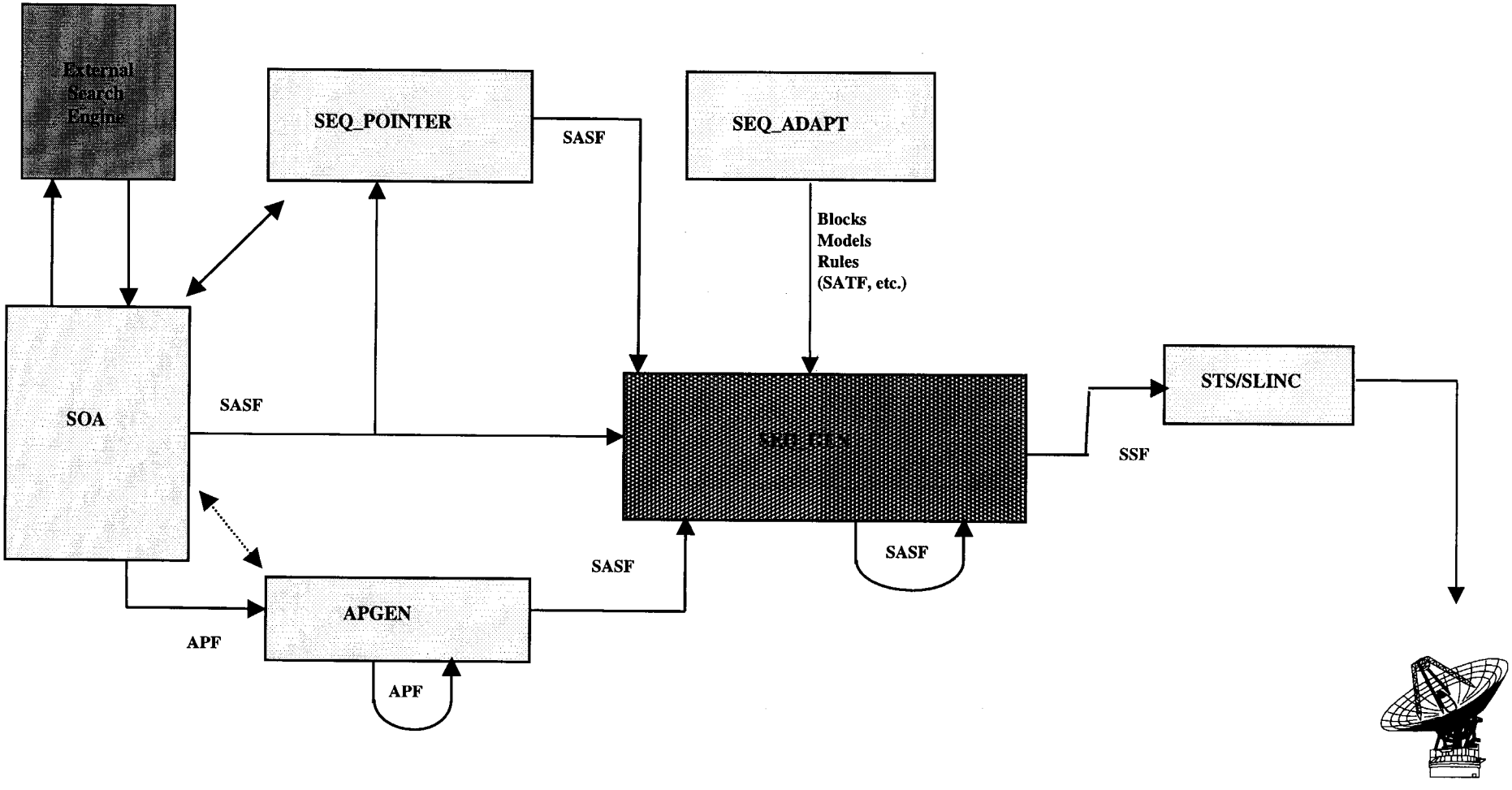
- Planetary Observation Instrument Targeting and Encounter Reconnaissance (POINTER) is an observation planning tool that works at the command level
  - More detailed calculations
  - More detailed constraint checking


JPL

# Engineering and Science Planning Tools

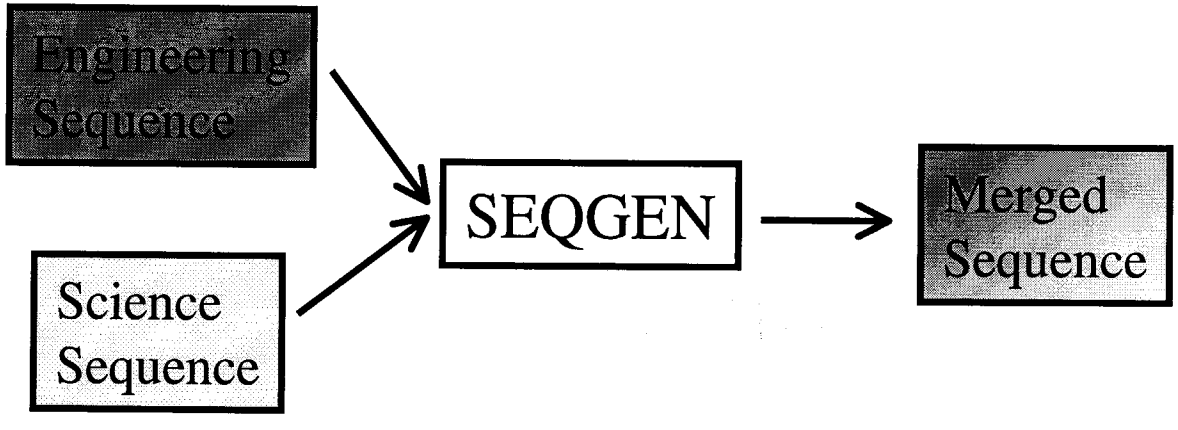
- Activity Plan Generator (APGEN) is used to plan engineering activities
  - Activities (DSN contacts, science activities, general engineering activities, etc.) are monitored against resource constraints (solid state recorder space, propellant, battery state of charge, etc.).
  - Scheduling of activities can also be completed.

# Sequence Generation Tool

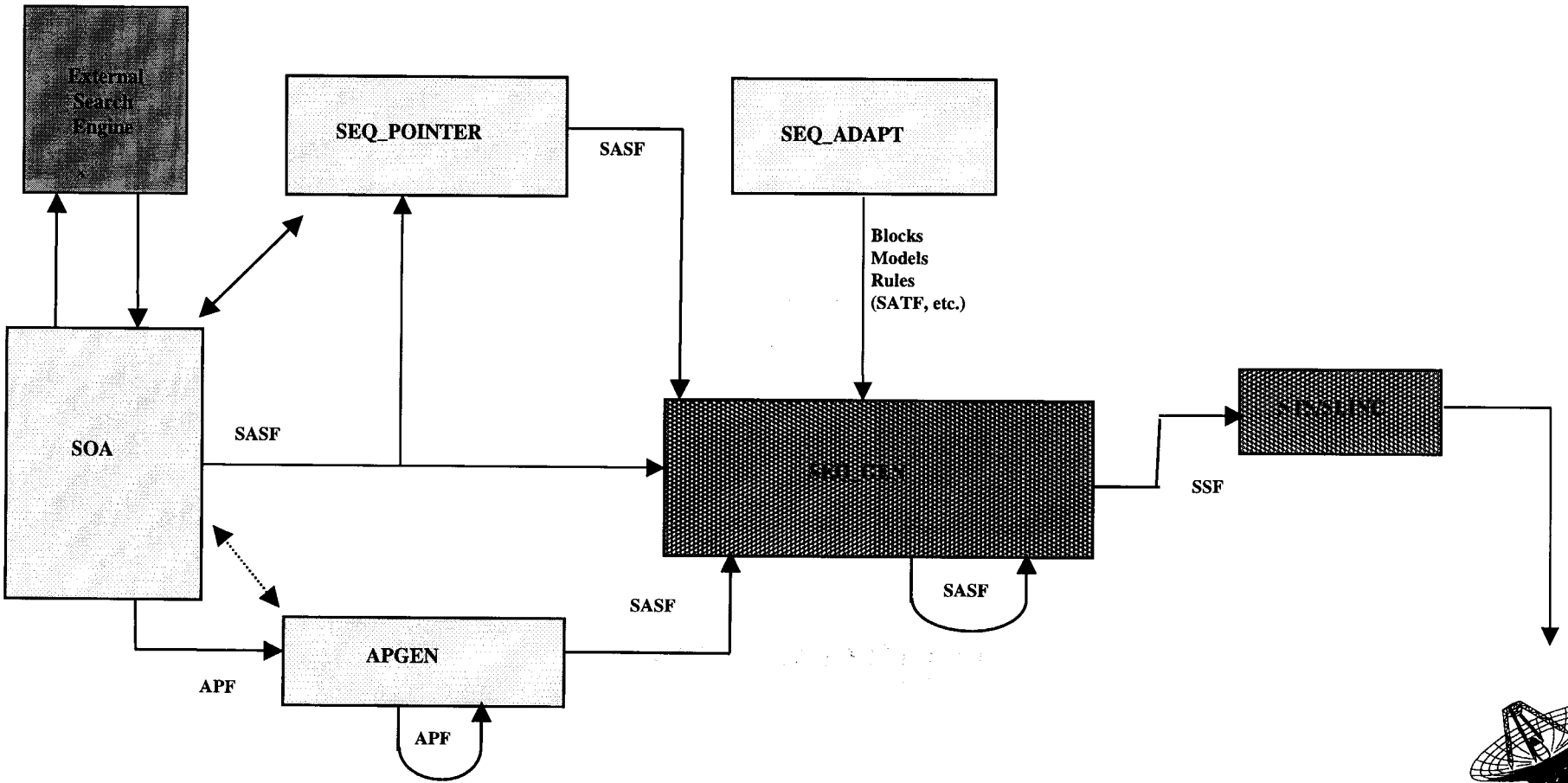


# Sequence Generation

- Sequence Generation (SEQGEN) is used to generate and integrate sequence commands
  - Sequences may have been generated by SOA, POINTER, APGEN, SEQGEN or other tools
  - Sequence integration is used to develop the overall sequence

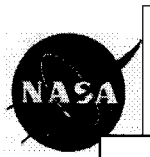


# Sequence Verification



# Sequence Verification

- Sequence Generation (SEQGEN) is also used for sequence verification.
  - Command syntax
  - Mission and Flight rules
  - Some types of parameter checking
- Spacecraft Language Interpreter and Collector (SLINC) packetizes and translates the commands into binary format.
  - Out of Range checking for all types of parameters is done.

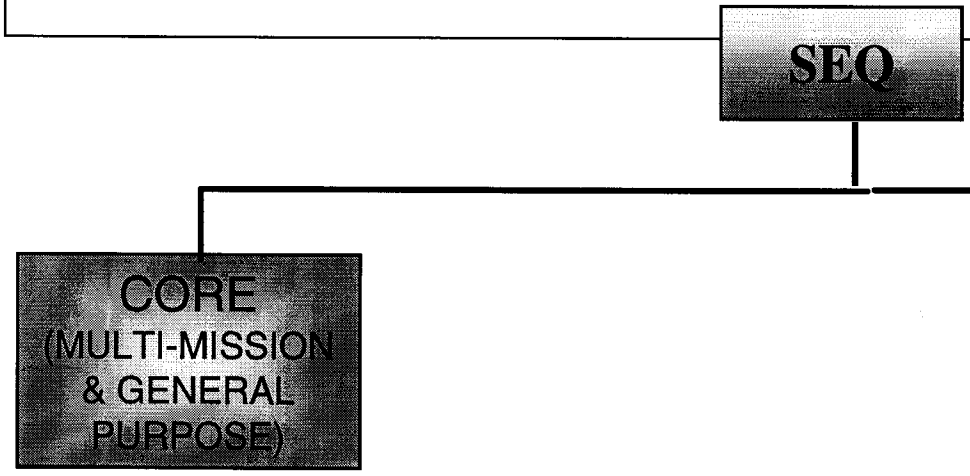



JPL

# Change in Sequence Architectures

- Until the early 1990s, each spacecraft developed its own individual sequencing system/programs. Sequencing systems are quite expensive to develop
- In an effort to reduce cost and development time, a "Multi-Mission" Sequencing architecture was developed which promoted reuse of sequencing system components.

# Current Sequence Architecture




**TOOLS FOR:**

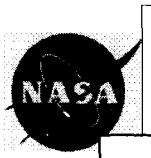
- **SCIENCE OPPORTUNITY ANALYSIS**
- **SEQUENCE AND ACTIVITY PLANNING**
- **SEQUENCE DESIGN**
- **SEQUENCE INTEGRATION & VERIFICATION**
- **COMMAND GENERATION**

**DATABASES AND SCRIPTS FOR PROJECT-SPECIFIC:**

- **COMMANDS**
- **FLIGHT RULES**
- **MODELS**
- **OPERATIONS PROCESSES**
- **PROCESS AUTOMATION (ASP)**

**TOOLS AND SCRIPTS FOR PROJECT-UNIQUE:**

- **FILE CONVERSIONS**
- **DATA INTERFACES**
- **OTHER PROCESSES**



# Core/Adaptation

- "Core" software contains the ability to perform a specific function
- "Adaptation" is the process of adding project specific components that rely on Core functionality.

# SOA

- CORE Functionality
  - SOA shall read a configuration file containing user specified file names and default values for SOA initialization.
  - SOA shall have the Parameter and Model data from the User Interface Component for interprocess communications.
- Adaptation Tasks
  - Define the configuration file to contain the correct kernel files, rule files, previously loaded queries, search engine information, model files, default times, default bodies, etc.
  - Define the mapping between the parameter and model variables used in SOA and other programs SOA communicates with.

# POINTER

- CORE Functionality
  - Provide the capability to display error and warning messages directly to the uses and/or the event listing hardcopy.
  - The time formats written to a sequence file shall be Greenwich Mean Time (GMT), epoch plus GMT, absolute clock time, epoch plus relative clock time.
  - During adaptation, read the Spacecraft Activity Type File for definitions of the SEQ_POINTER-applicable activities.
  - The operator shall be able to override the default workstation desktop color assignments and fill patterns by editing the Application Resource File.
  - Perform range checking on each parameter value.
- Adaptation Tasks
  - Implement the error and warning messages, and the models needed to trigger them.
  - Define a set of useful epoch times.
  - Provide a Spacecraft Activity Type File that contains definitions for the activities that will be used during modeling.
  - Provide a project adapted Application Resource File if the default configuration is not acceptable.
  - Provide ranges for parameter values.

# APGEN

- CORE Functionality
  - ApGen shall be delivered complete with an installation program.
  - Activity types shall have a mechanism to specify activity resource usage.
- Adaptation Tasks
  - Provide an architecture which supports ApGen usage and the environment variables and files needed by the program.
  - Provide the resource usage for each of the steps of a given activity. Define the resources that will be monitored.

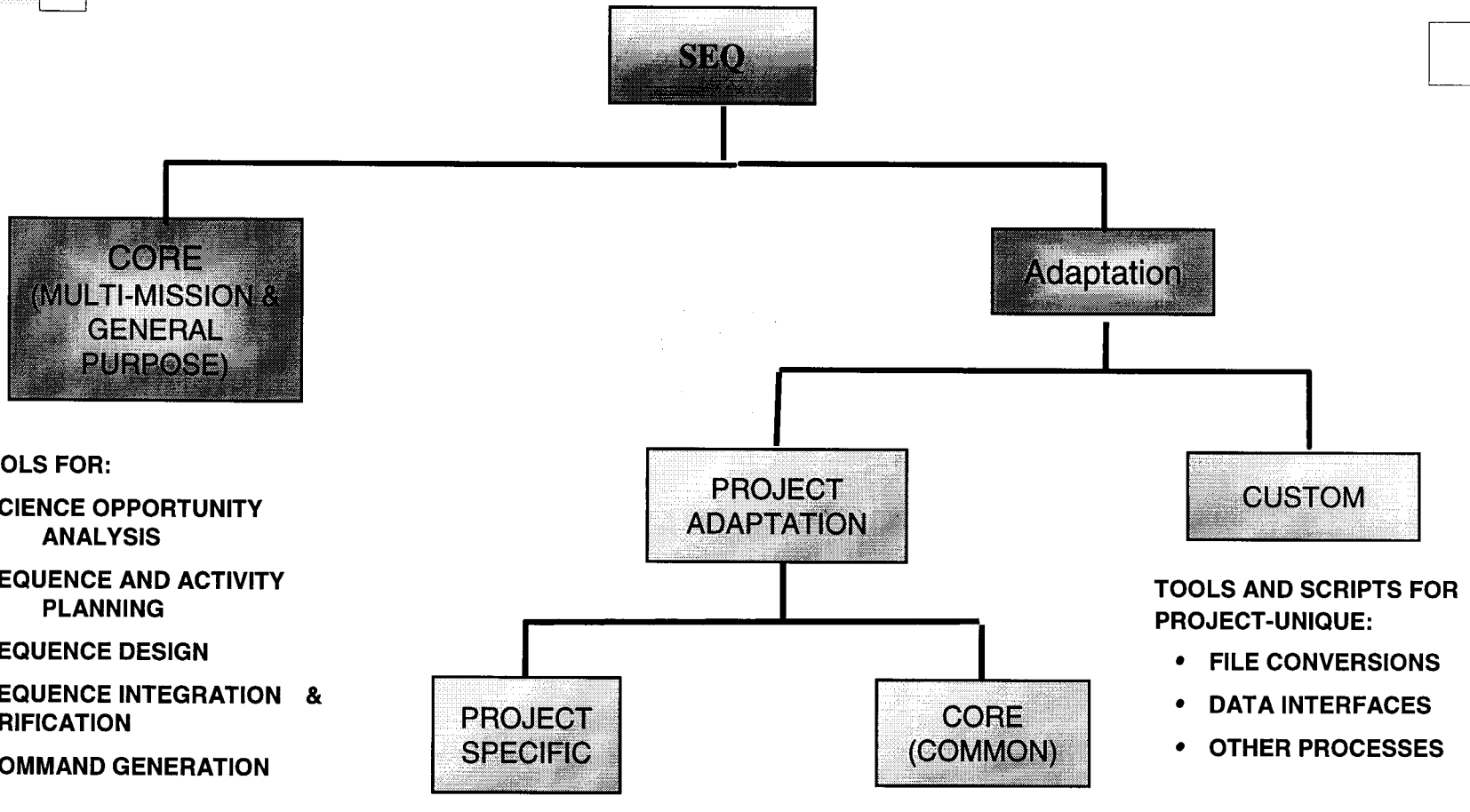
# SEQGEN

- CORE Functionality
  - SeqGen shall display an indication of each rule violation, either in the window of the subsystem involved in the rule, or in the timeline near a request causing a the rule violation.
  - Any textual display of time shall be either in UTC, or JPL local time or user local time including automatic conversion to daylight savings time, or the flight projects Spacecraft Clock units or epoch relative at the option of the user.
  - There shall be a command to allow the user to add a phrase of up to 1000 characters into the runlog. SeqGen shall append the time of day to the phrase.
  - SeqGen shall convert "activities" into "steps".

- Adaptation Tasks
  - Coding of the rules that are checked by SeqGen.
  - Define a useful set of epoch times.
  - Generate useful messages that will be included in the runlog. These messages will often include information about rule violations, comments about the beginning or ending of a block, or messages about parameters.
  - Define the steps needed to complete an activity

# Extensions of the Multi-Mission Concept into Adaptation

- In the most widely used of the Core tools, SEQGEN, the Adaptation effort has evolved so that certain aspects of the adaptation are developed using common models.




**TOOLS FOR:**

- **SCIENCE OPPORTUNITY ANALYSIS**
- **SEQUENCE AND ACTIVITY PLANNING**
- **SEQUENCE DESIGN**
- **SEQUENCE INTEGRATION & VERIFICATION**
- **COMMAND GENERATION**

**TOOLS AND SCRIPTS FOR PROJECT-UNIQUE:**

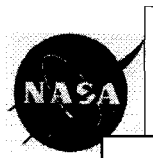
- **FILE CONVERSIONS**
- **DATA INTERFACES**
- **OTHER PROCESSES**

**DATABASES AND SCRIPTS FOR PROJECT-SPECIFIC:**

- **COMMANDS**
- **FLIGHT RULES**
- **PROJECT MODELS**
- **OPERATIONS PROCESSES**
- **PROCESS AUTOMATION (ASP)**

**DATABASES AND SCRIPTS MODELS:**

- **ORBIT PROPAGATION AND TIMING GEOMETRY (OPTG) MODEL**
- **DEEP SPACE NETWORK (DSN) ANTENNA MODELS**
- **PROCESS AUTOMATION (ASP)**



# Adaptation

- Project specific Adaptation
  - The model response to commands. This includes checking rules, changing modeling states, etc.
  - Designation of commands that are hardware commands versus commands that are to be handled by the flight software.
  - Flight and mission rule implementation
  - Implementation of project specific modeling for states or resources (modeling the Pointing and Control Subsystem, power models, etc.)
  - Implementation of blocks (activities) which are a canned series of commands that will be expanded for the user

# Adaptation

- Core (Multi-Mission) Adaptation
  - Modeling needed to read configuration files that are used to control parameter values on the spacecraft
  - Interpretation of Orbit Propagation and Timing Geometry (OPTG) files and models. These models take the keywords and data in an OPTG file and use it to model when eclipse, occultation, and other orbit and timing events occur.
  - Descriptors for DSN view period files that contain information needed to model the range of antenna visibility.
  - Modeling needed to support the use of DSN antennas and equipment, including generation of messages needed to generate keyword files used at the DSN.


JPL

# Advantages of the Multi-Mission Architecture

- Time savings
  - Since the basis for the sequencing system already exists, simple commands (NOOP) can be passed through the "real" sequencing system in less than a week.
- Cost savings
  - The "functionality" correctness is checked once. Projects need to verify only the adaptation.
  - Pool of adapters who are familiar with the sequencing software. Adapters can easily transition from one project to another.

JPL

# Disadvantages of the Multi-Mission Architecture

- Having to balance the needs across multiple customers
  - Conflicting desires in software functionality
  - The continued support of legacy needs
  - Balancing delivery schedules to meet needs of new customers or during critical mission events.