Reduce Costs with Multi-mission Sequencing and a Multi-mission Operations System

David A. Bliss

Overview

- This presentation is broken into two parts.
  - Part 1 discuss cost savings realized by multi-mission sequence integration.
    - Part 1 then discuss how the principles behind multi-mission sequence integration can be extended to event timeline development and science sequencing.
  - Part 2 discusses how the principles behind multi-mission sequence integration can be extended to a mission operations system (MOS) which is multi-mission

Part 1: Multi-mission Sequencing

- Historically, sequence development occurs in four steps:
  - Skeleton timeline development
  - Science observation sequence development
  - Engineering activity sequence development
  - Mission sequence integration, command generation, and product review

Part 1: Multi-mission Sequencing (cont)

- Mission Sequence integration
  - For stored sequences, execute the following steps:
    - merge science observations and engineering events into a complete sequence
    - translate the sequence into spacecraft-readable command packets
    - perform constraint checks and memory management
    - produce sequence review products
  - For real-time commanding, generate real-time commands and real-time command mini-sequences for transmission to the spacecraft according to a multi-mission real-time command process
- A key strategy to make Mission Sequence integration multi-mission, employ adaptable core software tools

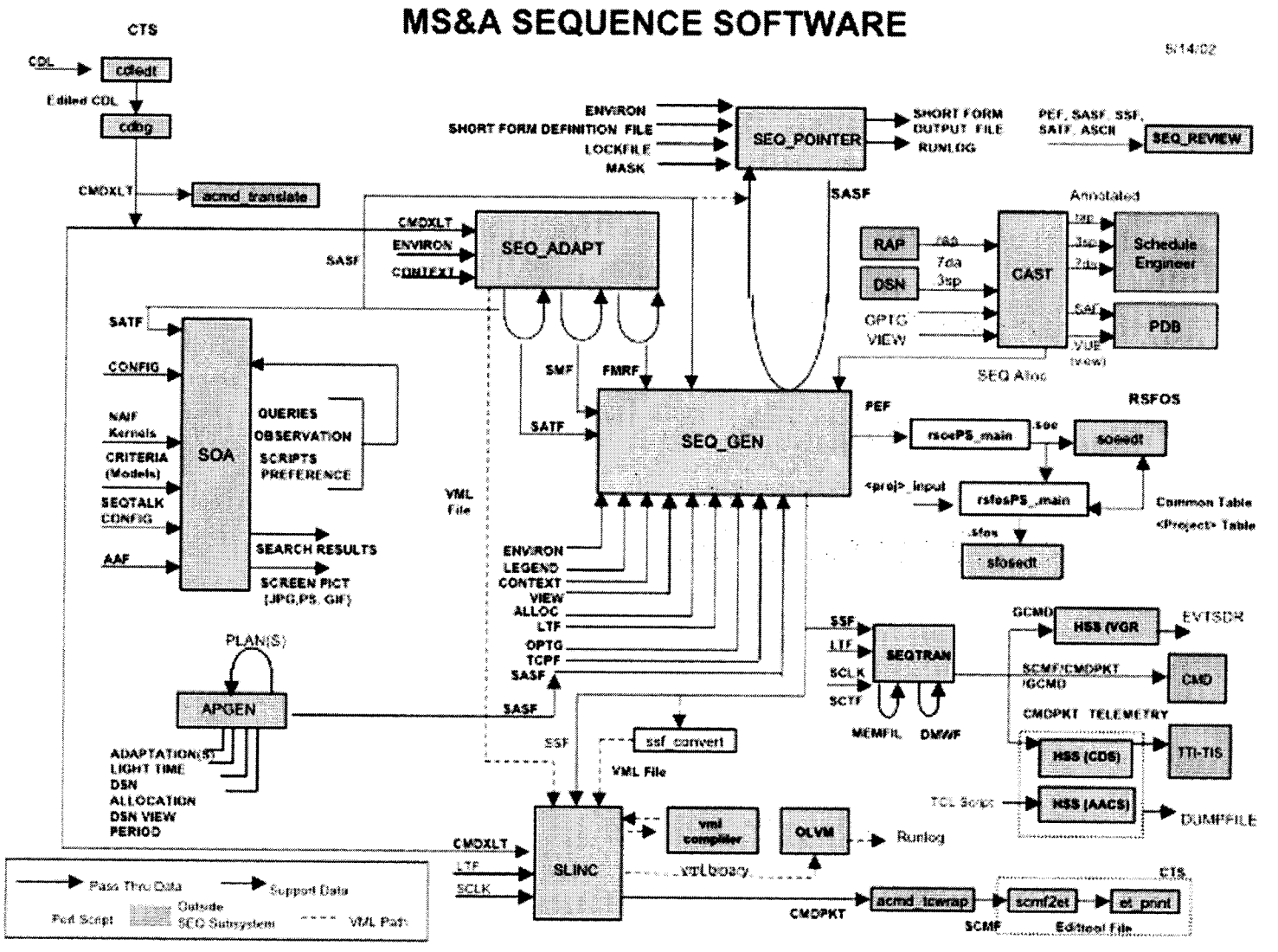
Part 1: Multi-mission Sequencing (cont)

- Core Software Tools used
  - SEQGEN (Merges inputs, expands input blocks into commands, constraint checks)
  - SEQTRAN (on Surveyor Bus based spacecraft, translates the SSF into S/C readable files and tracks memory usage and insures correct memory management)
  - SLINC (converts SSF or UNIX binary file to Command Packet File format)
  - CMD_TCWRAP (converts CPF into Spacecraft Message File)
  - AUTOGEN (for highly repetitive sequencing, SASF inputs)
  - Automated Command Tracker (tracks command and sequence status, review comments, provides required action notifications)
  - Electronic Command Request Form (form ACT uses to record, review, and approve results)

Part 1: Multi-mission Sequencing (cont)

- To achieve extremely efficient multi-mission sequence operations, core software tools are built to handle mission specific adaptations via mission specific "adaptation" files
  - Adaptation files define the mission-specific commands, models, and constraint checks. "
  - Adaptation file complexity is driven by mission complexity.
- Figure 1 shows the overall Mission Services and Applications (MS&A) software with both core and adaptation file inputs.

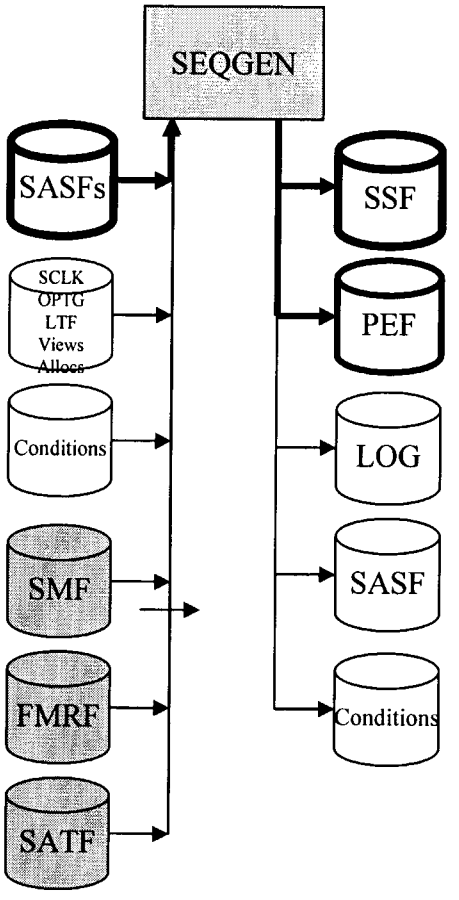
Figure 1


MS&A SEQUENCE SOFTWARE
5/14/02
CTS
CDL
cdledt
Edited CDL
cdbg
CMDXLT
acmd_translate
ENVIRON
SHORT FORM DEFINITION FILE
LOCKFILE
MASK
SEQ_POINTER
SHORT FORM
OUTPUT FILE
RUNLOG
PEF, SASF, SSF,
SATF, ASCII
SEQ_REVIEW
SASF
CMDXLT
ENVIRON
CONTEXT
SEQ_ADAPT
SASF
Annotated
RAF
DSN
7da
3sp
CAST
Schedule Engineer
PDB
OPTG
VIEW
SEQ Alloc
SATF
CONFIG
NAIF Kernels
CRITERIA (Models)
SEQTALK CONFIG
AAF
SOA
QUERIES
OBSERVATION
SCRIPTS
PREFERENCE
SEARCH RESULTS
SCREEN PICT (JPG,PS, GIF)
SMF
FMRF
SATF
SEQ_GEN
PEF
RSFOS
rsoePS_main
soeedt
<proj>_input
rsfosPS_main
Common Table
<Project> Table
sfosedt
VML File
ENVIRON
LEGEND
CONTEXT
VIEW
ALLOC
LTF
OPTG
TCPF
SASF
PLAN(S)
APGEN
ADAPTATION(S)
LIGHT TIME
DSN
ALLOCATION
DSN VIEW
PERIOD
SASF
SSF
ssf_convert
VML File
SSF
LTF
SCLK
SCTF
SEQTRAN
MEMFIL
DMWF
GCMD
HSS (VGR
EVTSDR
SCMF/CMDPKT
/GCMD
CMD
CMDPKT TELEMETRY
HSS (CDS)
TTI-TIS
TCL Scripts
HSS (AACS)
DUMPFILE
CMDXLT
LTF
SCLK
SLINC
vml compiler
OLVM
Runlog
CMDPKT
acmd_tcwrap
scmf2et
et_print
CTS
SCMF
Edited File
Pass Thru Data
Support Data
Perl Script
Outside SEQ Subsystem
VML Path

Part 1: Multi-mission Sequencing (cont)

- Consider SEQGEN as an example. SEQGEN consists of a multi-mission core program and a mission specific adaptation. SEQGEN allows a user to perform the following functions:
  - generate and modify requests
  - expand a series of requests into their resultant S/C commands
  - model these S/C commands
  - flag conflicts in the modeling of commands
  - flag violations of flight/mission rules
  - show the time extent of each request graphically, and
  - graphically display model attributes.

Part 1: Multi-mission Sequencing(cont)



- For SEQGEN, the mission specific adaptation employs the following set of "adaptation" files (shaded in green) that define the mission specific commands, models, and constraint checks:
  - Spacecraft Model File (SMF). Contains the definition of spacecraft and ground subsystem models, and spacecraft command/parameter definitions
  - Flight/Mission Rules File (FMRF). Contains flight and mission rule checking algorithms
  - Spacecraft Activity Type File (SATF). Contains names and definitions of the activity types, including on-board blocks, ground expanded blocks, SEQGEN directives and SLINC directives

Part 1: Multi-mission Sequencing(cont)

- Additional adaptation files include the following:
  - Context Variable (Definition) File (CVF). Contains parameters defined during the adaptation process that are used in the definition of activity types or models
  - Legend File. Contains data to define display definitions and layout
- Overall, SEQGEN requires the following core input files to perform sequence expansion and constraint checking:
  - Spacecraft Clock Coefficient File (SCLK)
  - Orbit Propagation Timing and Geometry File (OPTG)
  - Lighttime File (LTF)
  - DSN Viewperiod File (VP)
  - Viewperiod Format Description File (VIEW_FD)
  - DSN Station Allocation File (SAF)
  - Initial Conditions File (INCON)
  - Spacecraft Activity Sequence File (SASF)
  - Spacecraft Activity Type File (SATF)

Part 1: Multi-mission Sequencing(cont)

- SEQGEN generates the following outputs:
  - Spacecraft Activity Sequence File (SASF)
  - Spacecraft Sequence File (SSF)
  - Predicted Events File (PEF)
  - Final Conditions File (FINCON)
  - Run Log

Part 1: Multi-mission Sequencing(cont)

- These adaptable, core tools are "wrapped" so that their use enables a consistent, multi-mission process
- "Wrappers" are scripts which use tables that define states for different spacecraft
  - For example, during sequence integration, the sequence integration engineer identifies the spacecraft number.
    - "Wrapper" script then references the spacecraft data tables corresponding to the spacecraft number. These data tables tell the script what to do for the identified spacecraft.
    - So if running SLINC, the script will use a specified spacecraft number to consult a table to determine whether the spacecraft number refers to a VML spacecraft. If so, the script runs a VML compiler. If not, it executes another routine which involves memory management.
- By cross-training personnel to use these wrapped, adaptable, core tools, six engineers are successfully providing full sequence planning, integration and command generation services to four launched missions of mostly medium complexity (MGS, Genesis, Odyssey, and Stardust).
  - These numbers are much lower than what was required in the past (eg, at peak operations, Magellan employed six sequence integration and three planning engineers)

Part 1: Multi-mission Sequencing(cont)

- Extension to Skeleton Timeline Development
  - Skeletal timeline development produces the backbone upon which science observations and engineering events are placed
    - Process steps include sequence boundary identification, identification of key navigation and engineering event windows, high level identification of science campaign or observation boundaries
  - Core software tools which can be used include APGEN and SOA
    - APGEN automatically schedules DSN passes, critical engineering, and navigation events into a timeline based upon a mission-specific set of rules
    - SOA (Science Opportunity Analyzer) can accept mission-specific pointing and trajectory files for opportunity identification
- For missions of lesser complexity, the skeleton timeline development step can be eliminated

Part 1: Multi-mission Sequencing(cont)

- Extension to Science Sequencing
  - Science sequencing produces the detailed observation designs which are submitted to mission sequencing for integration.
    - Process steps include observation opportunity identification, observation design, observation command implementation, observation integration, science instrument and observation-internal flight rule checks.
  - Core software tools used include
    - SEQGEN (enables generation, modification, expansion, modeling, constraint checking of spacecraft commands),
    - PDT or POINTER (for remote sensing observations requiring spacecraft pointing and/or target motion compensation),
    - SOA (Science Opportunity Analyzer) can accept mission-specific pointing and trajectory files for opportunity identification and preliminary design for science observations
    - in-house tools that meet SEQGEN SIS requirements
- For missions of lesser complexity, science sequencing can be combined with mission sequence integration.

Part 2: Towards a Multi-Mission MOS

- Multi-Mission MOS seems very possible at Level 3
  - Level 3 capabilities include office-level general requirements, general uplink requirements, and general downlink requirements.
  - Level 3 capabilities also include flight rules and flight rules checking assignments, operational interface agreements, software interface specifications
  - Level 3 capabilities also include science and mission planning processes, sequencing processes, simulation process, navigation process, flight system analysis process, science data products process, and archive process
- Mission complexity contributes to mission operations costs

Part 2: Towards a Multi-Mission MOS (cont)

- Mission complexity contributes to mission operations costs
- Mission complexity can be classified as standard, medium, and complex
  - Standard missions
    - Do not have strict and tight pointing requirements,
    - Do not employ new technology on critical subsystems.
    - Operations can be repetitive (such as a mapping mission).
    - Teams (such as science integration and sequencing), can be combined and the number of required MOS components reduced.
    - Remaining MOS components can be employed with relatively small adaptations.

Part 2: Towards a Multi-Mission MOS (cont)

- Missions of medium complexity
  - Can have precise pointing requirements but not motion compensation.
  - Can have new technology on one critical subsystem which is well-tested and well-modeled.
  - Operations can be non-repetitive and can involve distant target contact such as firing at a target to analyze ejecta.
  - Teams (such as science sequencing and mission sequencing), can be combined and the number of required MOS components reduced.
  - MOS components, operations processes, and multi-mission tools can be employed with moderate adaptations.
  - Initial costing estimates would be performed based upon the number of MOS components used and then adjusted according to the number of moderate adaptations.

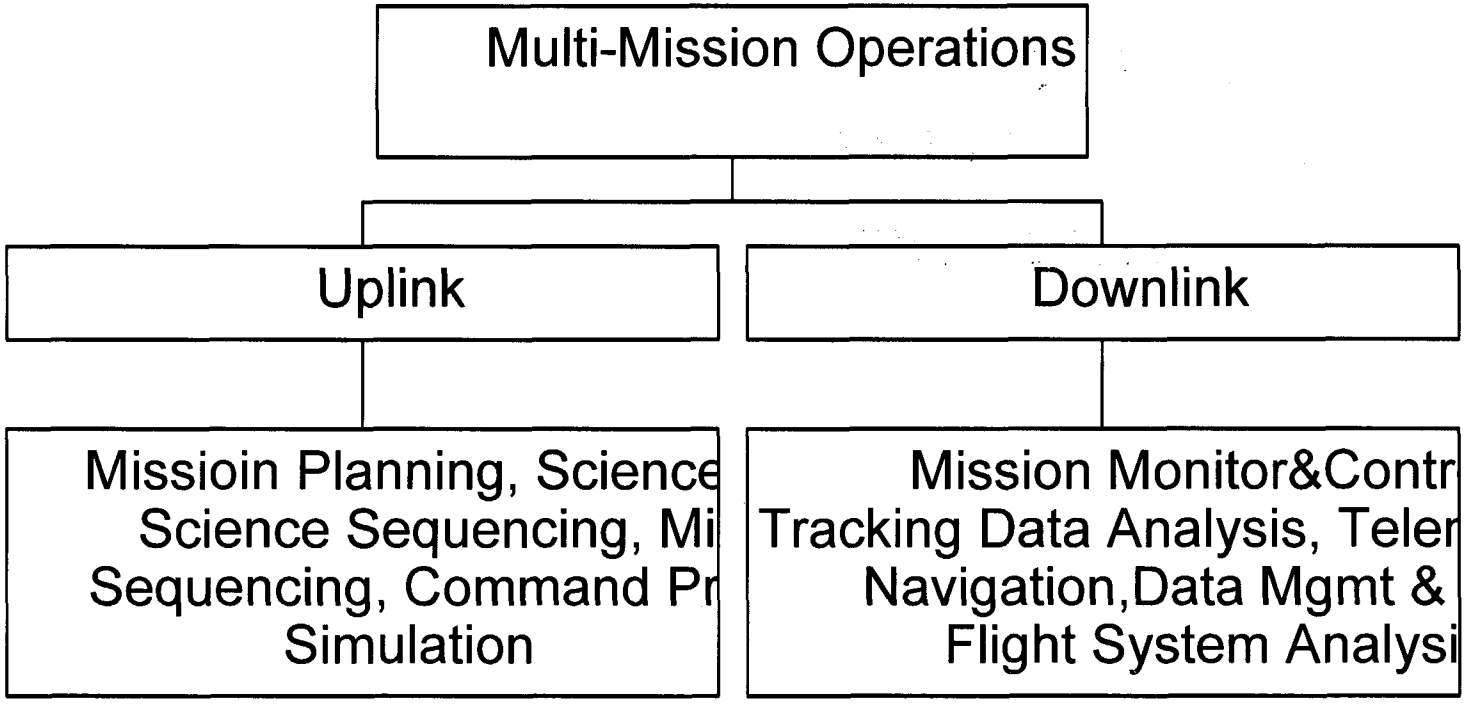
Part 2: Towards a Multi-Mission MOS (cont)

- Complex missions
  - Can have precise pointing requirements with motion compensation.
  - Can have new technology on one or more critical subsystems.
  - can have multiple, unique observation designs.
  - Main spacecraft body may be a lander.
  - May require all MOS components.
  - MOS components can be employed with extensive adaptations. Operations processes and multi-mission tools can be employed with extensive adaptations and tests.
  - Initial costing estimates would be performed based upon the complete set of MOS components used and then adjusted according to the number of extensive adaptations.

Part 2: Towards a Multi-Mission MOS (cont)

- By using the multi-mission sequencing strategy of developing a core capability which is adaptable to all missions of varying complexity, we can develop a multi-mission operations system which will greatly reduce development costs.

Part 2: Towards a Multi-Mission MOS (cont)



Part 2: Towards a Multi-Mission MOS (cont)

- Some parts of the MOS are already multi-mission
  - Deep Space Network (DSN) and the Deep Space Mission System (DSMS) services for handling telemetry, command, and radio metric data
  - Mission Sequencing

- Multi-Mission MOS seems very possible at Level 3
  - Level 3 capabilities include
    - General office-level, uplink, and downlink requirements.
    - Flight rules and flight rules checking assignments, operational interface agreements, software interface specifications
    - Science and mission planning processes, sequencing processes, simulation process, navigation process, flight system analysis process, science data products process, and archive process

Part 2: Towards a Multi-Mission MOS (cont)

- We propose an MOS baseline of level 3 core components could be made multi-mission
  - High level of commonality of level 3 components across missions
  - Differences should be largely accomodated by adjustments to core components
    - Analagous to adaptation of core mission sequence software
- Level 3 MOS baseline components can be quickly implemented
  - MOS Engineer could withdraw from the MOS Baseline database the MOS components characteristic of the project's complexity
  - Initial Costs can be extracted from recent project costs and existing cost algorithms
  - MOS could be constructed so that it could work, with adaptation, from the multi-mission core GDS

Part 2: Towards a Multi-Mission MOS (cont)

- Level 3 MOS baseline components can be quickly implemented (cont)
  - MOS adaptation scheduling, an automated multi-mission scheduling program (perhaps EXCEL) could lay out a schedule for the MOS component development based upon dates for Launch and key mission reviews
    - for complex missions, more time would have to be allowed for the adaptation and finalization of MOS components than for missions of medium complexity
    - The program would also allow some development margin (10% to 15%)
    - The program could then plot an entire MOS adaptation and finalization development timeline
      - Included in this schedule would be the date when the adapted multi-mission GDS would be available with the software and tools to support MOS functions

Part 2: Towards a Multi-Mission MOS (cont)

- Planning, costing, implementing, and validating a project mission operations system has sometimes required years
  - An MOS Engineer trained in the use of the multi-mission MOS Baseline database and scheduling tool could cut this time considerably
    - For a mission of moderate complexity requiring moderate adaptation, the MOS development timeline could be cut to weeks