SMALL SCALE SEQUENCE AU"J'OMA"1'1ON PAYS BIG DIVIDENDS

Bill Nelson
Jet Propulsion Laboratory
California institute of Technology

AllS'J'ItAC"l'

Galileo sequence design and integration are supported by a suite of formal software tools. Sequence review, however, is largely a manual process with reviewers scanning hundreds of pages of cryptic computer printouts to verify sequence correctness. Beginning in 1990, a series of small, PC-based sequence review tools evolved. Each tool performs a specific task but all have a common 'look and fee]." The narrow focus of each tool means simpler operation, and easier creation, testing and maintenance. Benefits from these tools are (1) decreased review time by factors of S to 20 or more with a concomitant reduction in staffing, (2) increased review accuracy, and (3) excellent returns on time invested.

Key Words: Sequence review, sequence automation

THE GALILEO SEQUENCING PROCESS

The Galileo sequencing process is a "top down" process that consists of two overlapping functions: the design and integration function and the review function. Both are iterative processes with a considerable amount of manual interaction. "Top down" means that development proceeds from the general to the specific. The major steps along the way are:

- A Planning phase which specifies the timing of mission phases and major activities. It covers one or more years and is the general guide for later, more detailed sequencing.
- A Design and Integration phase where the timing and placement of the major activities is finalized and where minor and supporting activities are added, all subject to timing and other resource constraints
- A Specification phase, where details arc added, parameters are specified and commands "expanded" from predefined routines, The end product of the specification stage is the final command level sequence.

The design and integration stage in particular benefits from prepackaged and pretested activities called Profile Activities or PAs. A Profile Activity is a sort of sequencing subroutine that encapsulates the commands making up its activity. Each PA has a name, a unique ID, a starting time, and a duration, Most also have further parameters that will later control the composition and timing of the encapsulated commands. PAs are an abstraction tool that frees the sequence designer from concern for the details of an activity. In the earlier stages, the designer need only consider the PA function, its start time and duration in integrating the activities into a composite whole. Unique activities are specified by a general purpose PA called the UTILITY PA. It has a start time and duration but no parameters. Its commands are added manually later in the expansion process,

After a sequence is integrated for the first time, it goes through an iterative development cycle of integration, review, correction and addition, and re-integration. As the cycle progresses, the sequence becomes more detailed and specific. General activities have more parameters specified. Supporting activities are added and made more specific, and resource predictions arc updated. This 'fleshing out" takes a big leap forward with the expansion step which results in a listing of all the specific spacecraft commands.

Once the development cycle begins, each iteration is reviewed by anywhere from half a dozen to nearly two dozen people. Reviewers represent various science instruments or engineering subsyst ems, ground station operations, and general spacecraft and sequencing perspectives. At earlier stages, the checks are fewer and more general while at later stages they, like the sequence itself, are more detailed and specific. Each reviewer uses checklists specific to these various development stages.

Because it is an obviously difficult job to integrate hundreds of PAs into a limited time span under numerous constraints, sequence design and integration tools have received considerable attention. The process is far from automatic but at least there are support tools to manipulate activities, to design experiments, to manage resources and to present activities graphically. Furt her, software development continues to stress sequence design and integration tools

SEQUENCE REVIEW SUPPORT

The review part of the cycle has received considerably less support. Most reviewers still go through hundreds of pages of cryptic computer printouts, manually highlighting items, checking for problems and marking their checklists. Only two mainframe based tools, the CHECKER module of SEQGEN and the STRIPPER program provided any sequence review support.

CHECKER is a hard coded constraint checker. While it can compare actual states against predicted or required states, and can check timing, those abilities are hard coded and limited to (usually) the simpler flight rules. CHECKER is also often out of date. With limited programming resources, it is simply not important enough to keep current. Spurious warnings are common and each must be checked and resolved by hand,

STRIPPER is a data extractor driven by a fixed, change controlled database. It was designed specifically to extract commands and it depends on the rigid sequencing format for proper operation, It cannot extract arbitrary text or scan arbitrary locations on a line. Because by policy, there is only one strip per subsystem, multiple or custom strips arc impossible. Generally, STRIPPER is used to create a subset of the main sequence product containing only the commands specific to a given instrument or subsyst em. Most science instruments and some engineering subsystems benefit from STRIPPER but those requiring a more global view such as Fault Protection, Power or Telecom do not. STRIPPER may reduce the product from several hundred pages to less than one hundred but those pages must still be reviewed by hand.

Beginning in 1990, a series of small, PC-based sequence review tools evolved. These were created by reviewers in their spare time and in response to their own needs. They were without official support and were unburdened with the paperwork and change control of more formal tools.

SKIMX, A DATA EXTRACTOR

One of the first of these tools was SKIMX, a data extractor so named because it could "skim" any arbitrary text, 'k," from a file. SKIMX accepted "match strings" from user prompts or from a file and extracted all lines containing any "match string" text. This gave sequence reviewers a means of creating custom strips. If a check required comparing two commands, for example, SKIMX would find all occurrences of the two commands - and only those commands. Comparison was then straightforward. In effect, the sequence could be separately interrogated for each of the different checklist checks. This simple tool alone cut review times by factors of 2-4. It also represented an excellent return on time invested.

SKIMX has several features that adapted it particularly well to sequence review. It could save the matched lines to a file for later use or for pasting into the reviewer's comments. It accepted frequently used sets of "match strings" from pt-c-defined datafiles. It counted the number of matches or reported "No match found" which simplified checking for forbidden commands. This feature was sometimes used simply to quickly count the number of occurrences of events. SKIMX could repot-t matches in either physical or logical lines. PAs are built as a single, comma delimited logical line with the end of the logical line denoted by a semicolon. A long logical line may take several physical lines, each intermediate physical line ending in a comma. Sometimes matching only the physical line is sufficient, sometimes the full PA, the logical line, is required.

The original SKIMX was created in a single day and when printed took all of four pages. Code for the actual "skim" occupied only half a page with the rest being help screen text, user prompting, and commenting. Within six months, SKIMX was regularly used by about a half a dozen people who reported anywhere from two to eight hours saved per review.

```
SKIMX V1/1.66 9/30/93

SKIMX finds all lines containing any specified string or strings.  SKIMX
    ignores upper/lower case.  Matches may be saved to an Output File.
USAGE: SKIMX [/x] . . . [/x] [Input FileSpec [, Output FileSpec] ]     where
    /x represents any of these options:
    /B for BLACK AND WHITE (monochrome) monitors.
    /C to force upper/lower CASE SENSITIVE matching.
    /FMatchFILESpec to read MatchStrings from plain ASCII MatchFILE.
    /H for HELP screen (this screen).
    /Kwordto enter a single KEYWORD MatchString from the command line.
        NO blanks, slashes, commas or '<,>,|' characters allowed.
    /M[m] [n] for MULTIPLE lines per item (like ORPRO files) . Omit 'm'
        for special handling of $ and . header lines, use 'n' decimal
        ASCII value to change terminator.
    /Q for QUICK output - no output to screen while working.
    /R to REVERSE the sense of the match.  This option OMITS matching
        lines and only lines WITHOUT any matching strings are saved.
    Input File Spec is the file of data to skim,
    Output File Spec is the file where skimmed output is put.  If omitted,
        output is to screen only.  NOTE: comma must separate FileSpecs.
                    Hit any key to continue
```

Figure 1- SKIMX help screen

SAFPRINT reformats the Station Allocation File,

```
PA2, STALOC,362A,PRI, 94-192/21:52:03.010,07 :15: 00, +07: 15:00,OMT,GLL GOE,
CFG 6081,T/P DMSCOND,94-192/22:25:00   .000,14,1733,94-92/23:   40:00.000, ,
S,3,,N,,94-193/05:25:00.000, 94-193/05:40:00.000;
PA2,sTVUPD,360A,PRI,94-192/22  :32:28.772,09:06:21,+09  :06:21,OMT,GLL GOE,
DSN VIEW,EJ7-5_ME_36.4 1994-193T03 :38:37,14,1733 ,94-192/23:05:26.000,
94-192/23:05:26.000,94=192/23  :29:15.000,94-193/07:47 :58.000,
94-193/08:11:47.000,94-193/08: 11:47.000;
PA2,sTHAND,366A,PRI,94-193/00:  22:57.546,00:15:00,+00 :15:00,OMT,GLL GOE,
DSN U/L,ACQUIRE UPLINK,94-192/23:50:00 .000,,
14, ,,+00:05:00, ,100 .0,HIGH,4.0,,,,,S;
PA2,sTHAND,366B,PRI,94-193/05  :57:59.529,00:15:00, +00:15:00,OMT,GLL GOE,
DSN U/L,TXR 0FF,94-193/05:25: 00.000,14,,,-00:05:00,,,,,,,,,,s;
```

into a more readable format:

```
 14    RISE: 94-192/23:05:26  SET :   94-193/08:11:47  MaxEl :  36.4 at 03:38:37
1733  BOT:   94-192/23:40:00  EOT:   94-193/05:25:00  DESC:    T/P DMSCOND
 HI    XON :  94-192/23:50:00  XOFF:  94-193/05:25:00  CFG :    6081, DUR: 05:45;
```

Figure 2 - SAFPRINT input and output

Now, some four year later, over two dozen people use SKIMX and the time saved to date is well over 1000 hours. (Since copies of SKIMX are kept on several Galileo servers, total usage is unknown). SKIMX itself has grown to seven pages but still represents a return on time invested of well over 1200 per cent.

DATA REFORMATTERS

Another early tool was SAFPRINT. This utility cast the Station Allocation File into a more readable format and in the process made some simple constraint checks. During its creation, SAFPRINT found errors in five consecutive Station Allocation Files. In response to this, SAFPRINT's constraint checking was expanded and a companion program, SAFCHECK, was created to checked for timing errors. SAFPRINT and S SAFCHECK were so successful that the Mission Control Team, the group responsible for creating and maintaining the Station Allocation File, adopted them as part of their standard internal checking procedures, 'l"here have been no timing or logical errors in any Station Allocation File pre-checked with the SAFPRINT suite of tools.

SAFPRINT is also used in sequence development. Here, however, its ability to convert allocations from their ground timeframe to spacecraft time is as valuable as the better format. Furthermore, SKIMX can used to interrogate the reformatted file to locate allocations by day or by scheduled activity. Success with SAFPRINT demonstrated that just casting data into a more convenient format is sometimes sufficient to gain significant savings of time and effort.

OPEVENT followed in the reformatting tradition by reformatting the unexpanded products, It gave the user the ability to select which PAs to reformat and which to ignore. Of the PAs being reformatted, he user could select which parameters to display, the display order and the titles to assign. One other unique feature was he ability to do time arithmetic on parameter fields. This made it possible to turn a start time and duration into an end time, or to make some limited ground to spacecraft time

conversions. The result was a sequence summary that, like SAFPRINT, could be further interrogated with SKIMX. Custom reformats with C) PEVENT provide one of the fcw tools for assisting reviewers in checking the unexpanded products. Since the PA description fields are not passed through into the expanded products, the summary is also the only easy way to spot significant activities -- the other means, the timeline, is primarily used as an early planning tool and is not kept updated.

The reformatting capabilities of OPEVENT have also been used to provide management with summaries of sequence activities and to provide alternative reformats of the Station Allocations File.

TELECOM SUBSYSTEM CO STRAINT CHECKERS

Finding and organizing or reformatting data simply did not address some review problems. Constraints with complex rules, those depending on current spacecraft state, those requiring time calculations and those without an easily identified trigger generally exceeded the abilities of SKIMX and OPEVENT.

One such difficult constraint was the Telecom check that no spacecraft events occurred during a data outage. Data outages were triggered by three types of events: (1) data rate changes, (2) switching between coherent and non-coherent mode, a function of both a commanded spacecraft state and

An OPEVENT reformat of the Station Allocations File

```
*CREATION    94-222/18:37:05.000
*BEGIN       95-268/19:10:08.439
'CUTOF F     96-014/17:13:26.530
'TITLE       STATION ALLOCATIONS FILE FOR JAJOE-5
95-268/22:28:50 STALOC,362A 0SS 14 BOT: 95-268/23:13:50 EOT: 95-269/03:13:50 CFG 008S
)5-269/18 :53:43 STALOC,362B DSS 14 BOT: 95-269/19:53:43 EOT: 95-270/02:58:43 CFG 6081
95-271/18:53:28 STALOC,362C 0SS 14 BOT: 95-271/19:53:28 EOT: 95-272/02:58:28 CFG 6081
95-272/18:23:21 STALOC,3620 DSS 14 BOT: 95-272/19:23:21 EOT: 95-273/02:58:21 CFG 6081
95-274/18:23:06 STALOC,362E DSS 14 BOT: 95-274/19:23:06 EOT: 95-275/02:58:06 CFG 6081
                          •      •      •
```

An OPEVENT reformat of the Comet Shoemaker-Levy observation sequence

```
                          •      •      •
94-198/02:56:16 DLKCAP,364 J      S-Band Sup Bit Rate: 10
94-198/03:46:00  CMDMRO,480LC     Dur: 08:36:00        Rate: 10          Desc: EVENT B BUFFER MRO PT 1
94- 198/05:29:27 CMDRS,157JB      Dur: +CDS 02:00:0    Desc:NIMS FRAGMENT C OBSERVATION
94-198/05:31:16  UTILITY,20JB     Dur: CDS 96:00:0     Desc:NIMS RECORD FRAG C
94-198/05:31:28  SCITLM,176JB     ELSMPW   CHG: NO     S_ HI_ LO: NONE       Desc: FRAGMENT C OBSERVAT ION
94- 198/05:31:28 TARGET, 165JB    Dur: +00:04:04       Body: JUPITER         Desc: FRAGMENT C OBSERVATION
94-198/05:31:28  CSMOS,117JB     Dur: +CDS 96:00:0    Desc:NIMS FRAGMENT C OBSERVAT ION
94-198/07:08:32  CMDRS, 157JZ    Dur: +CDS 02:00:0    Desc:NIMS FRAGMENT C OBSERVAT ION
94-198/07:24:40  UTI LITY,20VM    Dur: :05:            Desc: SAFE S/P FOR SAS MAINT
94-198/08:11:14  DLKCAP,364K     S-Band Sup Bit Rate: 40
94-198/09:41:13  Sc IT LM,611D176KD ELSLRS   CHG: NO   S_HI_LO: NONE       Desc:UVS F-F ON JUPITER
94-198/09:49:18  SCITLM,611D1761D NCGIM4   CHG: NO     S_H I_LO: NONE       Desc: SSI_SL-9_ IMPACT_D
94-198/09:41:02 UT IL ITY,20EA    Dur: 2:15:00         Desc:SSI/UVS RECORD FRAG D
94-198/09:41:13  TARGET,1651D     Dur: +02:02:22       Body: JUPITER         Desc: SSI_SL-9_ IMPACT_D
94-198/09:41:13  SMOS,1181D       Dur: +CDS 116:78:0   Desc: SSI_SL-9_ IMPACT_D
94-198/09:41:13  INITRS,1281D     Dur: +CDS 01:01:0    Desc: SSI_SL-9_IMPACT_D
94-198/09:56:14 0LKcAP,364L       S-Band Sup Bit Rate: 10
                          •      •      •
```

Figure 3- OPEVENT output examples

the presence of an uplink to the spacecraft, and (3) station Begin-Of-Track. Outage duration depended on the data rate and was expressed as a probability of successful lockup. The faster and less restrictive lockup time applied only to certain events and only during certain mission phases. Station Begin-Of-Track did not have a separate and unique line in the review product.

OUTCHK for "Outage Check" was the program written to perform this task. It had to do all the following:

- track the spacecraft data rate and coherency mode,
- determine when station Begin-Of-Track occurred and "trigger" an outage for it,
- resolve overlapping station coverage,
- resolve overlapping data outages,
- identify all data outages and compute their durations, and
- identify any spacecraft activity in any outages discovered.

As a test, the time to hand check a particular sequence for data outages was recorded. It took the analyst 14 hours to complete. OUTCHK was then run on the same sequence. Its elapsed time, including the time to print its report, was 12 minutes. A comparison of the two checks showed that OUTCHK had correctly identified all data outages found by the analyst, had correctly timed all data outages including several the analyst had not, and had found three more outages that had been missed in the hand check. This represents a seventyfold decrease in checking time with increased accuracy as well.

OUTCHK was written part time in about three weeks with fewer than 80 hours invested. Even with updates, it still has fewer than 120 hours invested while the estimated time savings run well over 1000 hours. This represents over an 800% return on time invested.

Two other related tools arc also used for difficult telecom constraint checking, one to verify events have ground station coverage and the other to verify the data rate is supportable, Combined with OUTCHK and SKIMX, these tools have cut average Telecom review time by a factor of about twelve: what once took a week is now done in an afternoon.

By launching the checking programs from a batch file, still more of the user's time can be saved. Typical sequences take from five to fifteen minutes to process through the Telecom sequence checking batch file. During this time, the user is free for other duties.

UTILITY PROGRAMS

The sequence review effort has also been aided by several small utility programs. The first of these, DAYS, converted calendar dates to and from day-of-year and computed the day-of-the-week. DAYS covers the years 1583 (the beginning of the Gregorian calendar) through 9999. Two digit years are assumed to lie between 1980 and 2079. Typing 'DAYS TODAY" returns the current date in both calendar and day-of-year formats (or an error message if the computer's clock isn't current).

TIMECALC adds and subtracts times in hours: minutes: seconds format. It has a memory store and recall function that is ideal for adding or subtracting a one-way light time from a series of number.

PA RENUM was originally written to change the PA identification suffixes after a file had been created or edited by cutting and

pasting PAs. At the request of several users, it was expanded to also renumber sub-PAs and commands. PA_ RENUM isn't often needed but when PAs must be renumbered, the only alternative is change each suffix manually.

COMMON CHARACTERISTICS

Shortly after the creation of SKIMX, it was apparent that there was no easy means for users to verify they had the latest version. This lead to the definition of a common user interface, the general format being shown in Figure 1, the SKIMX help screen. All programs show date and version, all accept options before filenames, all use the forward slash as an option switch character, and all respond to "/l I" with a standard help screen.

To facilitate batch file operation, all programs accept command line input. If required information is missing, the user will be prompted to supply it. Programs verify that the specified files exist and will re-prompt if necessary.

Programs benefit from a 'toolkit" of utility and support routines, about half written in assembler, that provide services such as time addition and subtraction, parsing the command line, tokenizing a logical line, verifying file existence, setting up help screens and screen colors, and modeling various spacecraft and ground resources. The toolkit both enables and enforces much of the commonality among the programs.

Most of the programs also have accompanying "DOC" files that expand on what each program does, how it does it, what its options arc, and often includes review tips or other usage information.

OBSERVATIONS AND CONCLUSIONS

This suite of programs shows worthwhile savings of time and effort can be achieved with a relatively small programming effort. The problems and programs may be relatively small but that doesn't mean insignificant: for example, the Telecom unit will use these programs instead of hiring two additional analysts during the intensive Orbital Operations phase of the mission.

By finding and organizing data, by presenting it in more easily understood ways, and by performing rote logical tests and checks, these small scale sequencing review tools have dramatically reduced the time and effort required of this formerly all manual process.