A VALIDATOR FOR SASF (SPACECRAFT ACTIVITY SEQUENCE FILE) NASA LANGUAGE INTERFACE IN DAWN/VIR EXPERIMENT

S. Fonte[1], F. Carraro[1], E. Ammannito[1],A. Coradini[1], M.C. De Sanctis[1], G. Magni[1], [1] INAF-IAPS, Via Fosso del Cavaliere 100, Rome - Italy; sergio.fonte@ifsi-roma.inaf.it.

Introduction: Dawn is a NASA mission for Solar System exploration; there are three instruments involved, one is a Visible and Infrared spectrometer (an Italian industrial development). The commands of each instruments come using the SASF (Spacecraft Activity Sequence File) interface: an ASCII timing script containing the instruments setting. In order to avoid problems during the management of SASF language, a syntax analyzer for this script has been developed: VIRV (VIR Validator). The VIRV purpose is to help in checking the SASF syntax and to produce a simulation of the instrument operative sequence that will be executed in the VIR flight machine.

The development guidelines of VIRV are the idea of usual compiler, as C/C++ or fortran compilers, where a language (as close as possible to human language) is translated into machine language (where each byte can be understood by the microchip which will execute it. It is the so called machine byte code) during the compilation/linking phase. VIRV was developed to behave as a C/C++ compiler: there's no byte code but a table with a simulation of VIR instrument operation during a SASF procedure.

VIRV internal structure: The VIRV include two macro system: the Lexer and the Parser (see figure 1). The Lexer checks the syntax of input file (”file.sasf”) and provides the tokens to the Parser, that performs a checks of the grammar: using internal rules, it checks if the tokens match the programmed grammar.

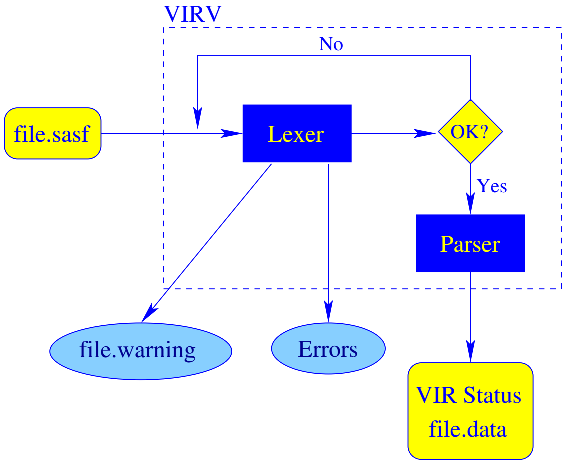


Figure 1: The internal organization of the VIRV.

The VIRV Parser is designed to check the presence of a violation of the VIR flight rules within the analyzed sequence. It makes a test of the instruments constraints on the data volume and checks if VIR command parameters values are contained within their boundaries.

The results of the checking are reported into two file (”file.warning” and ”Errors”), while the instruments status are save in extra file (”file.data”)

To perform a VIR simulation, an apposite data structure has been created within the Parser (see figure **??**): a SASF Process Unit (SPU), the Memories Unit and the Settable parameters. The goal of these data structures is to perform a simulation of the VIR instrument flight status diagram. The internal register of the VIR instrument are simulated with the Settable Parameters, it includes the real values of the instrument parameters: like the default science mode acquisitions, the time need to stabilize the infrared detector before use it, and so on.

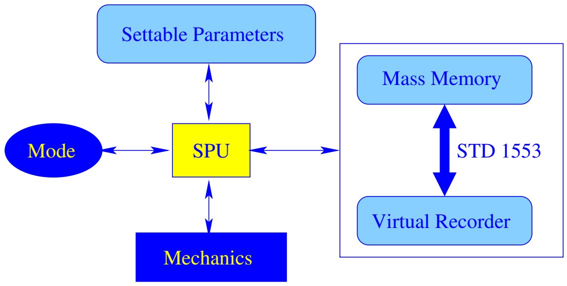


Figure 2: The VIRV internal control structures.

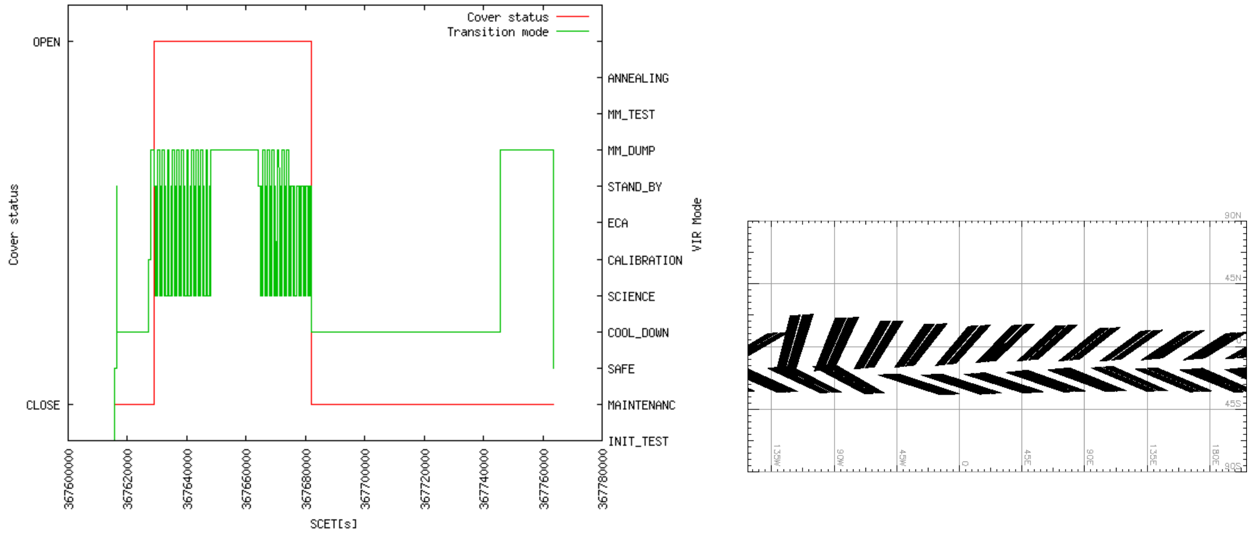
The Mode unit includes the transitions status for VIR instrument, it is used to check if VIR performs the operations as planned.

The Mass Memory and the Virtual Recorder units are used to store informations about the data volume occupation during the VIR operations; it is simulated also the data flux between the two units.

The Mechanics unit performs a simulation about the VIR instrument mechanics (as the cover actuation); it is used to checks if the Cover is opened during the VIR science acquisitions.

The core of the simulation is the SPU, it organizes the other units to perform the virtual operations that come from the input file (”file.sasf”).

SASF language analysis, the Backus Naur Form: A SASF is an ASCII text file containing three parts: an header, an optional cyclic definition section and a body part. The header defines the general context in which the file was produced. The cyclic definition section allows a user to define one or more ad hoc repetitive sequencing constructs composed of fundamental building blocks. The body gives the detailed parameter keyword pairs for the developing sequence.



(a) The transition mode.

(b) The footprint on Vesta of VIR field of view.

Figure 3: Some plots of the simulations during the Dawn/Survey on Vesta of VIR operations.

The BNF (Backus Naur Form) describes the grammar of a language using a specific structures. These structures include the tokens of the language (the brick of the frame language); the ”::=” and ”|” elements concern the syntactic relationships between the tokens.

The core of the SASF is the Body parts: this is composed by micro cells (”step section”) that includes the instrument commands. The ”step section” is represented by the following BNF code:

```
<step section> ::= [<list_of_steps>]

     <list_of_steps> ::= <step> [,<list_of_steps>]

     <step> ::= <step name>(<common fields>,
                <step specific fields> )

     <step name> ::= activity
                   | note
                   | command
                   | spawn

     <common fields> ::= <step_label>,
                         <scheduled time field>

     <scheduled time field> ::=
           SCHEDULED_TIME, \<time offset>\
                          [,<timing relation>],

     <timing relation> := FROM_REQUEST_START
                        | FROM_PREVIOUS_START
                        | FROM_PREVIOUS_END
                        | DEFAULT
```

The SASF is timing script sequence, so the command must wait for the end of the command before; this characterize is represented by the ”schedule time field”: it includes the time offset parameters and the timing relation.

The VIR commands are managed by the directive ”command” while the Spacecraft operations are managed by the directive ”activity”. The directive ”note” represent a comment in the SASF language. The directive ”spawn” loads a command block into the Spacecraft’s memory.

The VIR and the Spacecraft commands and them parameters are included into the ”step specific fields”.

Results of the simulations: In the figure 3 are reported some graphical reports about the VIR operations performed during the Dawn/Survey campaign. An usual plot includes the Cover mode status and the VIR transition status, this plot is use to checks if the Cover is open during the VIR acquisition. As you can see, there are eleven status for VIR while for the Cover mode there are only two modes; the green line indicates the transitions while the red one is the Cover actuations.

Another type of plot produced by the validator is the foot print of FOV (Filed Of View) on the asteroid; in the figure 3 the asteroids is Vesta. This plot is used to check the section of asteroid observed.

References: [1] C.T. Russell, A. Coradini, *Planetary and Space Science*, 52, 5-6, 465-489, 2004.

[2] M.C. De Sanctis, et al., *Space Sci. Rev.* DOI 10.1007/s11214-010-9668-5, 2010.

[3] John R. Levine, Flex & Bison, *O’Reilly*, ISBN: 978-0-596-15597-1

[4] J. Stipanuk and Adans Ko, *Deep Space Mission System (DSMS) Subsystem Software Interface Specification* D-10,2004.