3rd NASA Langley Formal Methods Workshop

Formal Methods Demonstration Project for Space Applications

May 11, 1995

John C. Kelly, Ph.D. and Ben DiVito, Ph.D.

- Joint NASA Code Q RTOP -

Jet Propulsion Laboratory*

Johnson Space Center

Langley Research Center

* The work described in this presentation was partially carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not consitute or imply its endorsement by the United States Government or the Jet Propulsion Laboratory, California Institute of Technology.

Contents

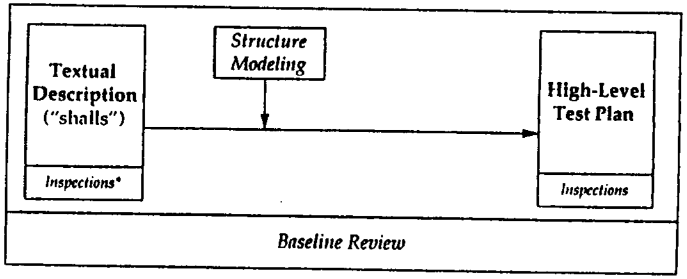
Overview of the Demonstration Project and the Guidebook - John C. Kelly
- The Software Requirements Problem
- Background Project Information
- Phases of the Project
- The NASA Formal Methods Specification and Verification Guidebook

Detailed look at applying Formal Methods to the Shuttle's GPS Upgrade Task - Ben DiVito
- Description of GPS CR Subset
- PVS Modeling of Principal Functions
- Receiver State Processing
- Results and Feedback
- Summary

The Requirements Problem in Engineering Software Subsystems

Requirements and design specifications are a high priority candidate for better software engineering techniques
- Most hazardous software safety errors found during system integration and test of two NASA spacecraft were the result of requirements discrepancies or interface specifications [Lutz93].
- The highest density of major defects found through the use of software inspections was during the requirements phase. This was 7 times higher than the density of major defects found in code inspections [Kelly92].
- Requirements errors are between 10 and 100 times more costly to fix at later phases of the software lifecycle than at the requirements phase itself [Basili84], [Boehm84], [Kelly92].

Current Requirements Process without Formal Methods



Textual Description ("shalls")

*Inspections**

Structure Modeling

High-Level Test Plan

Inspections

Baseline Review

* Note: The word "inspection" is used differently in software than in manufacturing or hardware context. In software the term 'inspection" refers to a structured peer review that uses a checklist and includes the collection of defect statistics. In software development, inspections are an upstream quality enhancement technique which support the principles of TQM (refer to NASA Standard 2202-93 or the two JPL Professional Development courses on this topic).

History of Formal Methods POP

Spring 1992	- JSC, JPL and LaRC submit independent Formal Methods proposals to NASA Code Q
Summer and Fall '92	- Code Q funds a Feasibility Study to derive a plan for how the 3 Centers could jointly do a Formal Methods RTOP
Winter '93	- Formal Methods Demonstration Project officially starts with a kick-off meeting at JSC
October '93	- Deliverables for Phase I completed (Formal Methods Specifications, Proofs, etc.)
December '93	- Phase I Report completed for demonstration of Formal Methods on an existing spacecraft
FY94	- Phase II Case Studies - Formal Methods Guidebook (Vol 1)
FY95	- Phase III Case Studies - Formal Methods Guidebook (Vol 2)

Purpose

- The Goal of this study is to demonstrate the applicability of Formal Methods techniques on critical NASA software subsystems
- Phase I Task: Demonstrate the Applicability of Formal Methods to Shuttle's On-Board Jet Select Software Subsystem (A highly critical, yet relatively stable set of requirements)
- Phase II Tasks: Demonstrate Formal Methods on several smaller projects which are currently developing critical software and provide guidance at the managerial level
- Phase III Tasks: Demonstrate Formal Methods on a large critical project in the early development stages, demonstrate at the design level, and provide guidance at the technical level

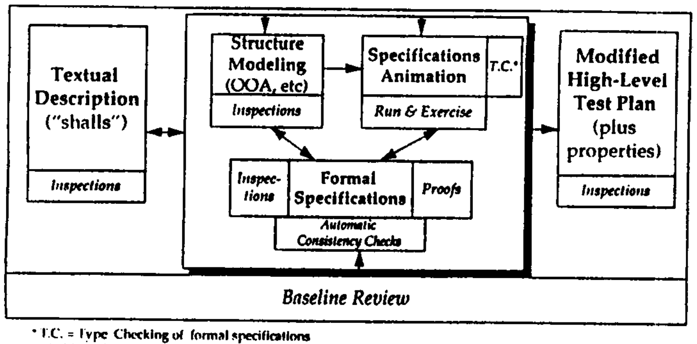
Introduction: Team Members

- Jet Propulsion Laboratory
  - John Kelly, Ph.D., Rick Covington, Ph.D., Robyn Lutz, Ph.D., Ken Abernethy, Ph.D. (Furman U.)
- Johnson Space Center
  - Ernie Fridge, Dan Bowman (LORAL), Mike Beims (LORAL-Shuttle RA), Chris Hickey (LORAL-Shuttle RA)
- Langley Research Center
  - Ben DiVito, Ph.D. (VIGYAN), Judith Crow, Ph.D. (SRI), Rick Butler
- NASA HQ Sponsor: Alice Robinson
- Alumni
  - Betty Cheng, Ph.D. (MSU), Mori Khorrami (JPL), Doc Shankar, Ph.D. (IBM), Scott French (LORAL), Sally Johnson (LaRC), John Rushby, Ph.D. (SRI), Sam Owre (SRI), Al Nikora(JPL), Brent Auernheimer, Ph.D. (CSUF), Yoko Ampo (NEC), David Hamilton (HP)

Generic approach for using Formal Methods to analyze requirements for Space Applications

- Select a portion of the requirements which is mission critical
- Model the selected requirement subset's structure using an informal approach (Object Modeling Techniques was used on some of the pilots)
- Develop Formal Specification for required functionality and state behavior (PVS was used for this on most of the pilots)
- Translate desired properties of the subsystem into a formal language then attempt to verify them against the Formal Specification with a theorem prover (PVS was used in most cases)
- Feedback lists of issues to Requirements Analysts and Development Engineers throughout this process

Integrated Formal Methods Process



Textual Description ("shalls")
Inspections
Structure Modeling (OOA, etc)
Inspections
Specifications Animation
T.C.*
Run & Exercise
Inspec-tions
Formal Specifications
Proofs
Automatic Consistency Checks
Modified High-Level Test Plan (plus properties)
Inspections
Baseline Review

* T.C. = Type Checking of formal specifications
** OOA = Object Oriented Analysis

Project Tasks and Deliverables

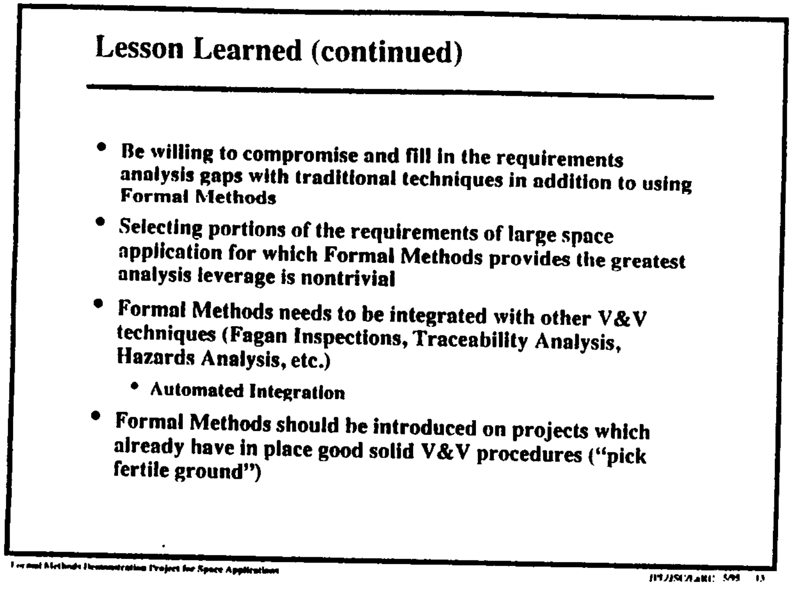
- Pre-project
  - Formal Methods Feasibility Study - October 1992
- Phase I: FY 1993
  - Shuttle's Jet Select Subsystem
    - 3 Levels of Formal Specifications
    - Ada emulator
  - Formal Methods Case Study Report
- Phase II: FY 1994
  - Shuttle's
    - MIR Docking Change Request
    - Orbit Digital Auto Pilot Model
    - Three Engine Out CR
    - Global Positioning System Change Request (ongoing)

Project Tasks and Deliverables (continued)

- Phase II: FY 1994 (continued)
  - Cassini Space Craft- Fault Protection (Requirements)
  - International Space Station Alpha FDIR (ongoing)
  - Formal Methods Guidebook - (development of Volume I)
- Phase III: FY 1995 (current)
  - Cassini Space Craft- Fault Protection (Design)
  - International Space Station Alpha FDIR
  - AP101S Assembler for Shuttle
  - Global Positioning System Change Request for Shuttle
  - Formal Methods Guidebook
    - Review Vol. I
    - Develop and Review Vol. 2

Overview of Results and Lessons Learned

- Formal Methods uncovered significant issues in mature requirements
- Even in an unstable requirements environment, formal specifications were found to be beneficial
- Requirements Analysts and Developers were generally impressed by the thoroughness and insight of the *issues lists* produced by the pilot study team
- Object Modeling Technique provides a good road map before developing Formal Specification
- PVS is robust enough to be used for the practical application of Formal Methods to requirements

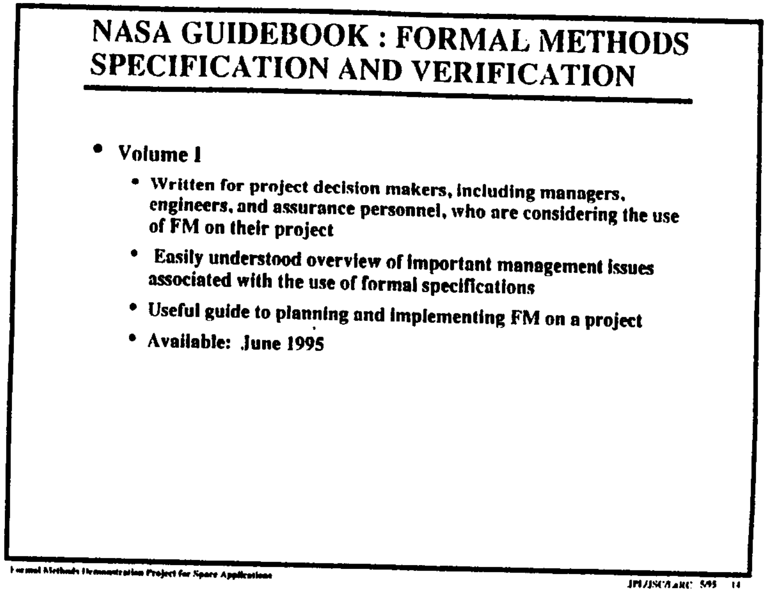


Lesson Learned (continued)

- Be willing to compromise and fill in the requirements analysis gaps with traditional techniques in addition to using Formal Methods
- Selecting portions of the requirements of large space application for which Formal Methods provides the greatest analysis leverage is nontrivial
- Formal Methods needs to be integrated with other V&V techniques (Fagan Inspections, Traceability Analysis, Hazards Analysis, etc.)
  - Automated Integration
- Formal Methods should be introduced on projects which already have in place good solid V&V procedures ("pick fertile ground")

Formal Methods Demonstration Project for Space Applications

JPL/JSC/LaRC 5/95 13

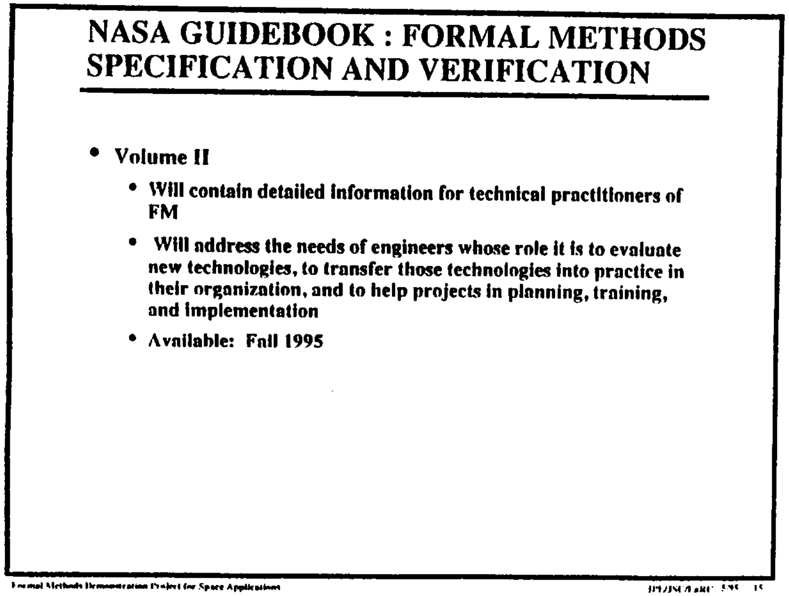


NASA GUIDEBOOK : FORMAL METHODS SPECIFICATION AND VERIFICATION

- Volume I
  - Written for project decision makers, including managers, engineers, and assurance personnel, who are considering the use of FM on their project
  - Easily understood overview of important management issues associated with the use of formal specifications
  - Useful guide to planning and implementing FM on a project
  - Available: June 1995

Formal Methods Demonstration Project for Space Applications

JPL/JSC/LaRC 5/95 14

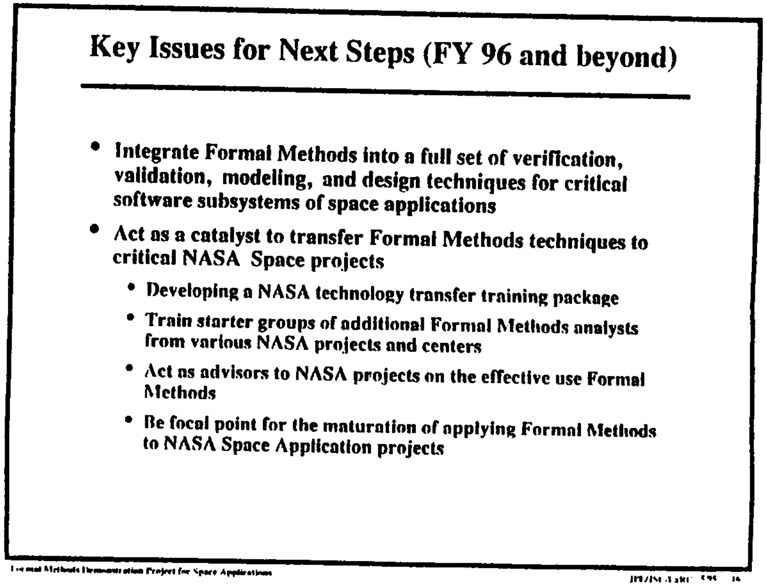


NASA GUIDEBOOK : FORMAL METHODS SPECIFICATION AND VERIFICATION

- Volume II
  - Will contain detailed information for technical practitioners of FM
  - Will address the needs of engineers whose role it is to evaluate new technologies, to transfer those technologies into practice in their organization, and to help projects in planning, training, and implementation
  - Available: Fall 1995

Formal Methods Demonstration Project for Space Applications

JPL/JSC/LaRC 5/95 15



Key Issues for Next Steps (FY 96 and beyond)

- Integrate Formal Methods into a full set of verification, validation, modeling, and design techniques for critical software subsystems of space applications
- Act as a catalyst to transfer Formal Methods techniques to critical NASA Space projects
  - Developing a NASA technology transfer training package
  - Train starter groups of additional Formal Methods analysts from various NASA projects and centers
  - Act as advisors to NASA projects on the effective use Formal Methods
  - Be focal point for the maturation of applying Formal Methods to NASA Space Application projects

Formal Methods Demonstration Project for Space Applications

JPL/JSC/LaRC 5/95 16

C-2.

S2-61

N96-10028

2

FORMAL METHODS DEMONSTRATION PROJECT FOR SPACE APPLICATIONS

Ben L. DiVito

VÍGYAN, Inc

p. 7

The Space Shuttle program is cooperating in a pilot project to apply formal methods to live requirements analysis activities. As one of the larger ongoing Shuttle Change Requests (CRs), the Global Positioning System (GPS) CR involves a significant upgrade to the Shuttle's navigation capability. Shuttles are to be outfitted with GPS receivers and the primary avionics software will be enhanced to accept GPS-provided positions and integrate them into navigation calculations. Prior to implementing the CR, requirements analysts at Loral Space Information Systems, the Shuttle software contractor, must scrutinize the CR to identify and resolve any requirements issues.

We describe an ongoing task of the Formal Methods Demonstration Project for Space Applications whose goal is to find an effective way to use formal methods in the GPS CR requirements analysis phase. This phase is currently under way and a small team from NASA Langley, ViGYAN Inc. and Loral is now engaged in this task. Background on the GPS CR is provided and an overview of the hardware/software architecture is presented. We outline the approach being taken to formalize the requirements, only a subset of which is being attempted. The approach features the use of the PVS specification language to model "principal functions," which are major units of Shuttle software. Conventional state machine techniques form the basis of our approach.

Given this background, we present interim results based on a snapshot of work in progress. Samples of requirements specifications rendered in PVS are offered for illustration. We walk through a specification sketch for the principal function known as GPS Receiver State Processing. Results to date are summarized and feedback from Loral requirements analysts is highlighted. Preliminary data is shown comparing issues detected by the formal methods team versus those detected using existing requirements analysis methods. We conclude by discussing our plan to complete the remaining activities of this task.

Using Formal Methods to Analyze the Space Shuttle's GPS Change Request

Ben L. Di Vito

ViGYAN, Inc.
30 Research Drive
Hampton, VA 23666

3rd NASA Langley Formal Methods Workshop
11 May 1995

Shuttle Program Background

- Contractor organization
  - Rockwell International – Shuttle prime contractor
  - Loral Space Info. Sys. (was IBM) – Software contractor
  - Draper Lab – Experts in Guidance, Navigation and Control
- Software modifications are packaged as Change Requests (CRs)
  - Usually modest in scope, localized in function
  - Provide capabilities to meet specific mission needs
- Software releases are called Operational Increments (OIs)
  - Include one or more CRs – issued around once per year
- Requirements Analysis
  - Conducted by Loral Requirements Analysts (RAs) prior to turning CR over to development team
  - Once in development, problems are more costly to fix

Global Positioning System (GPS)

GPS is a satellite-based navigation system operated by DoD

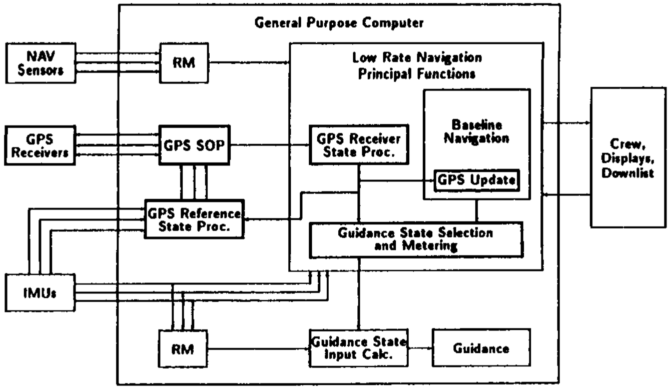
- Constellation of 24 satellites in high orbits
- Receive-only system requires dedicated hardware
  - Need to track 4 or more satellites simultaneously
  - Separate signals need to be recovered after undergoing code division multiplexing
- Receivers solve for position and velocity (and time)
  - Standard Positioning Service gives 100m accuracy
  - Precise Positioning Service gives 10m accuracy
- DoD phasing out TACAN navigation system by 2000

GPS Change Request

Shuttle Program is undertaking a large and complex CR to add GPS navigation capabilities to the Shuttle fleet

- Motivation: loss of TACAN navigation system
  - Need equivalent navigation aid during entry and landing
- Two-phase integration plan
  - Single-string implementation (one receiver)
  - Full-up implementation (three receivers)
- Integrated architecture
  - GPS receivers provide navigation data to General Purpose Computers (GPCs)
  - Several new Principal Functions added to software
  - Many smaller changes made to existing navigation software
- GPS CR more complex than typical CR

GPS Integrated Architecture



Approach to Applying Formal Methods

We are pursuing a selective application of formal methods

- Focus on core subset involving new principal functions
  - GPS Receiver State Processing
  - GPS Reference State Processing
- Use PVS to model principal function requirements
  - Derive stylized specification approach tailored to Shuttle software
  - Maintain traceability to existing requirements
  - Emphasize readability by nonexperts
- Formulate properties and attempt proving later, if warranted
  - Possible candidate is feedback loop from receivers to Receiver State Processing and back

Description of GPS CR Subset

Principal functions are decomposed into subfunctions

- Receiver State Processing
  - GPS IMU Assign
  - GPS Navigation State Propagation
  - GPS State Vector Quality Assessment
  - GPS State Vector Selection
  - GPS Reference State Announced Reset
- Reference State Processing
  - GPS External Data Snap
  - IMU GPS Selection
  - GPS Reference State Initialization and Reset
  - GPS Reference State Propagation

PVS Modeling of Principal Functions

Principal functions are regularly scheduled software entities

- Execution environment provides a large variable space that is read from and written into
- Interface is defined by explicit inputs and outputs enumerated in tables
- Principal functions are composed of several subfunctions invoked sequentially
  - Inputs are "passed down" to subfunctions
  - Outputs are "passed up" from subfunctions
- Local variables may be either transient or persistent
- We use a simple state machine model encoded in PVS
  - Principal function is represented as a single state-transition function

$$M : I \times S \rightarrow [O \times S]$$

PVS Modeling of Principal Functions (Cont'd)

Abstract structure in PVS notation:

```
pf_result: TYPE = [# output: pf_outputs, state: pf_state #]

principal_function (pf_inputs, pf_state,
                    pf_I_loads, pf_K_loads,
                    pf_constants) : pf_result =

      (# output := <output expression>,
         state  := <next-state expression>
      #)
```

- Principal functions use two kinds of variable data (input values, previous-state values) and three kinds of constant data (I-loads, K-loads, constants)
- Executing a principal function produces output values and next-state values
- All side effects are to be captured by this model

Requirements for Receiver State Processing

2.0 Step 2.1 is performed for each receiver up to the software designed maximum, whether or not the receiver has been actually installed on the vehicle:
DO FOR I = 1 to GPS_SW_CAP

2.1 For each GPS receiver that has a valid state (GPS_DG$_I$ = ON) and has not been deselected by the crew (CREW_DESELECT_RCVR$_I$ = OFF), perform the following:

IF (GPS_DG$_I$= ON and CREW_DESELECT_RCVR$_I$= OFF) THEN

If all receivers have been forced to be candidates for selection (GPS_AIF_RCVD = 'FORCE'), independent of their quality assessment status, or if the quality assessment status for that receiver has been overridden (CREW_QA_OVERRIDE$_I$ = ON) or the receiver's state has passed all quality assessment tests (GPS_FAIL_QA$_I$ = OFF), specify that the state is a candidate for selection by setting the selection command for that receiver and increment the counter of candidate states:

```
IF ( GPS_AIF_RCVD = 'FORCE' or
   CREW_QA_OVERRIDE_I = ON. or
   GPS_FAIL_QA_I = OFF THEN
      SEL_CMD_I = 1
      NUM_GPS_SEL = NUM_GPS_SEL + 1
```

3.0 If the number of GPS states identified as candidates is greater than 0, (NUM_GPS_SEL >0), then the selected GPS states are formed from the eligible candidate as described.

```
      R̄_GPS_SEL  = R̄_GPS_2
      V̄_GPS_SEL  = V̄_GPS_2
```

Requirements for Receiver State Processing (Cont'd)

Table 4.3.3.3-1 GPS Navigation State Propagation Input Parameters (cont'd)

Description	Symbol	Input Source	Type	Prec	Range	Units	Sample Rate
GPS data good flag	GPS_DG$_I$**	*	D	—	ON/ OFF	—	Filter rate
Previous GPS data good flags	GPS_DG_PREV$_I$**	GPS SV quality assess	D	—	ON/ OFF	—	Filter rate
Flag indicating (ON) a GPS selected state vector is available	GPS_SV_SEL_AVAIL	GPS SV select	D	—	ON/ OFF	—	Filter rate
Maximum number of GPS receivers for which the GPS specific software is designed	GPS_SW_CAP	K-load	I	—	3	—	Filter rate
Snapped major mode flag	NAV_MM_CODE	*	I	—		—	Filter rate
Latest GPS receiver orbiter position estimate in WGS84 coordinates	R̄_GPS_ECEF$_I$**	*	V	DP		ft	Filter rate

* See principal function input list in Table 4.3.3.7-1.

** I = 1, 2, 3 (GPS Receiver ID)

Modeling of Receiver State Processing

Selected data types rendered in PVS

```
major_mode_code:     TYPE = nat
mission_time:        TYPE = real
GPS_id:              TYPE = {n: nat | 1 <= n & n <= 3}

receiver_mode:       TYPE = {init, test, nav}
AIF_flag:            TYPE = {auto, inhibit, force}

M50_axis:            TYPE = {Xm, Ym, Zm}
position_vector:     TYPE = [M50_axis -> real]
velocity_vector:     TYPE = [M50_axis -> real]
GPS_positions:       TYPE = [GPS_id -> position_vector]
GPS_velocities:      TYPE = [GPS_id -> velocity_vector]

GPS_predicate:       TYPE = [GPS_id -> bool]
GPS_times:           TYPE = [GPS_id -> mission_time]
GPS_FOM_vector:      TYPE = [GPS_id -> GPS_figure_of_merit]
tolerance_vector:    TYPE = [GPS_id -> real]
WGS84_to_EF_matrix:  TYPE = [earth_fixed_axis -> [WGS84_axis -> real]]
```

Sample Subfunction of Receiver State Processing

```
ref_state_anncd_reset_out: TYPE = [#
      GPS_anncd_reset_avail:   GPS_predicate,
      GPS_anncd_reset:         GPS_predicate,
      R_ref_anncd_reset:       GPS_positions,
      T_anncd_reset:           mission_time,
      T_ref_anncd_reset:       mission_time,
      V_IMU_ref_anncd_reset:   velocity_vector,
      V_ref_anncd_reset:       GPS_velocities
      #]

ref_state_announced_reset(DT_anncd_reset,
                          GPS_DG,
                          GPS_SW_cap,
                          R_GPS,
                          T_anncd_reset,
                          T_current_filt,
                          T_GPS,
                          V_current_GPS,
                          V_GPS) : ref_state_anncd_reset_out
```

Principal Function Interface Types

```
rec_sp_inputs: TYPE = [#
      crew_deselect_rcvr:      GPS_predicate,
       . . .                    . . .
      V_GPS_ECEF:              GPS_velocities_WGS84,
      V_IMU2_save:             GPS_velocities
      #]

rec_sp_state: TYPE = [#
      GPS_DG_prev:             GPS_predicate,
       . . .                    . . .
      V_last_GPS_sel:          velocity_vector
      #]

rec_sp_I_loads: TYPE = [#
      acc_prop_min:            real,
       . . .                    . . .
      M_WGS84_to_EF:           WGS84_to_EF_matrix,
      sig_diag_GPS_nom:        cov_diagonal_vector
      #]
```

Principal Function Interface Types (Cont'd)

```
rec_sp_K_loads: TYPE = [#
      GPS_SW_cap:              num_GPS
      #]

rec_sp_constants: TYPE = [#
      deg_to_rad:              real,
      earth_rate:              real,
      G0:                      real,
      nautmi_per_ft:           real
      #]

rec_sp_outputs: TYPE = [#
      corr_coeff_GPS:          real,
      DELR_ratio_QA2_display:  GPS_ratios,
       . . .                    . . .
      V_ref_anncd_reset:       GPS_velocities
      #]

rec_sp_result: TYPE = [# output: rec_sp_outputs, state: rec_sp_state #]
```

Principal Function Specification

```
GPS_receiver_state_processing((rec_sp_inputs:    rec_sp_inputs),
                              (rec_sp_state:     rec_sp_state),
                              (rec_sp_I_loads:   rec_sp_I_loads),
                              (rec_sp_K_loads:   rec_sp_K_loads),
                              (rec_sp_constants: rec_sp_constants) )
                              : rec_sp_result =
  LET
    IMU_assign_out =
      IMU_assign(
        V_current_filt      (rec_sp_inputs),
        V_IMU2_save         (rec_sp_inputs)),

    nav_state_prop_out =
      nav_state_propagation(
        acc_prop_min        (rec_sp_I_loads),
         . . .               . . .
        V_GPS_ECEF          (rec_sp_inputs),
        V_last_GPS          (IMU_assign_out),
        V_last_GPS_sel      (rec_sp_state) ),
```

Principal Function Specification (Cont'd)

```
SV_qual_assess_out =
    state_vector_quality_assessment(
        GPS_DG              (rec_sp_inputs),
         . . .               . . .
        V_GPS               (nav_state_prop_out) ),

state_vect_sel_out =
    state_vector_selection(
        corr_coeff_GPS_nom  (rec_sp_I_loads),
         . . .               . . .
        V_GPS_sel           (nav_state_prop_out) ),

ref_st_ann_reset_out =
    ref_state_announced_reset(
        DT_anncd_reset      (rec_sp_I_loads),
         . . .               . . .
        V_GPS               (nav_state_prop_out) )
```

Principal Function Specification (Cont'd)

```
IN
  (# output := (#
      corr_coeff_GPS :=     corr_coeff_GPS     (state_vect_sel_out),
       . . .                 . . .              . . .
      GPS_fail_QA4 :=       GPS_fail_QA4       (SV_qual_assess_out),
      GPS_lat :=            GPS_lat            (state_vect_sel_out),
      GPS_lon :=            GPS_lon            (state_vect_sel_out),
       . . .                 . . .              . . .
      V_ref_anncd_reset :=  V_ref_anncd_reset  (ref_st_ann_reset_out)
      #),

    state := (#
      GPS_DG_prev :=        GPS_DG_prev        (SV_qual_assess_out),
      GPS_SV_sel_avail :=   GPS_SV_sel_avail   (state_vect_sel_out),
       . . .                 . . .              . . .
      V_GPS_sel :=          V_GPS_sel          (state_vect_sel_out),
      V_last_GPS_sel :=     V_last_GPS_sel     (nav_state_prop_out)
      #)
  #)
```

Results (In Progress)

Experience with effort so far:

- Outlook is promising, but it's still early
- CR requirements are still converging
  - Another revision cycle is likely
  - After next cycle, FM results should be more meaningful
- FM-based review is helping requirements analysis
  - Many interface errors being detected
- PVS can be used effectively to formalize this application
  - Custom specification approach should be easy to duplicate
  - Good prospects for continuation by nonexperts
  - Specification activity assisted by tools, but doing manual specification is also feasible here

Feedback from Requirements Analysts

Some RAs are optimistic about potential impact of FM

- Approach used is helpful in detecting two classes of errors:
  - "Requirements meet the CR author's intent; CR will work"
  - "Interfaces Documented and Consistent"
- Preliminary comparison with conventional process
  - Errors detected in Reference State Processing versus those also found by current process

Error Severity	With FM	Existing
High Major	2	0
Low Major	5	1
High Minor	17	3
Low Minor	6	0
Totals	30	4

Continuation Plans

Plan for near term CR analysis

- Continue elaborating formal specifications using current style
  - Complete two principal functions under way
  - Consider adding other functions as resources allow
  - Collect feedback and data from RAs
- Go down to moderate level of detail
  - Incorporate enough detail to capture branching conditions
- Evaluate tradeoffs of formalizing subsystem properties
  - Properties based on sequences of state vectors
  - Limited proving may be worthwhile

21

Summary

- Formal methods aiding requirements analysis on a significant Shuttle CR
- Specification techniques customized for Shuttle software
- So far, keeping up with requirements analysis process
  - Enables meaningful comparisons
- Early results are encouraging
- Remains to be seen whether techniques can be transferred to and adopted by RAs

22

omit

Session 5: Software Systems (2)

C. Michael Holloway, Chair

- **Ada 9X Language Precision Team**, by *David Guaspari*, Odyssey Research Associates
- **Introduction to Penelope and Its Applications**, by *David Guaspari*, Odyssey Research Associates