Software Engineering for Satellites

Kathryn Anne Weiss

Software Engineering Research Laboratory
Department of Aeronautics and Astronautics
Massachusetts Institute of Technology
October 22, 2003

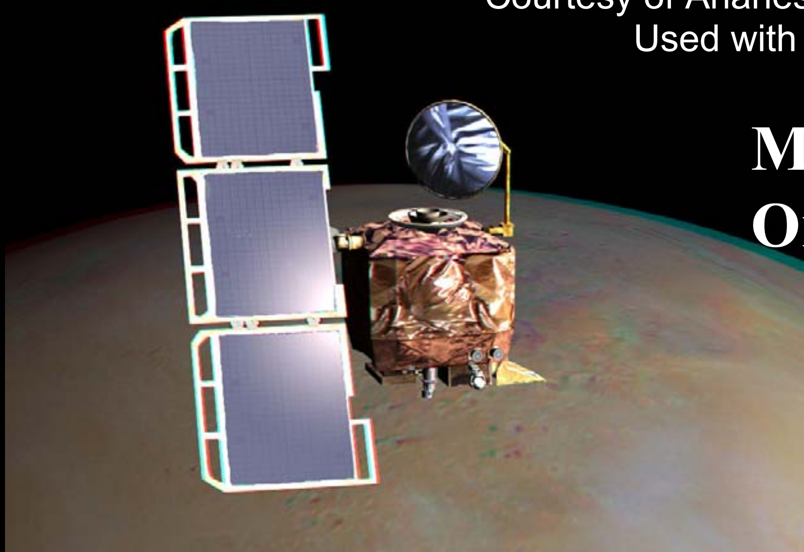
Topics of Discussion

- Background
  - Why is Software Engineering Hard?
  - Lifecycle
    - Cost
    - Requirements Specification
    - Approaches to Design
    - Implementation
    - Testing
    - Maintenance
- Why is Software Engineering Hard for Spacecraft?
- SERL Approach
- Component-Based Systems Engineering
  - SPHERES
- Conclusions

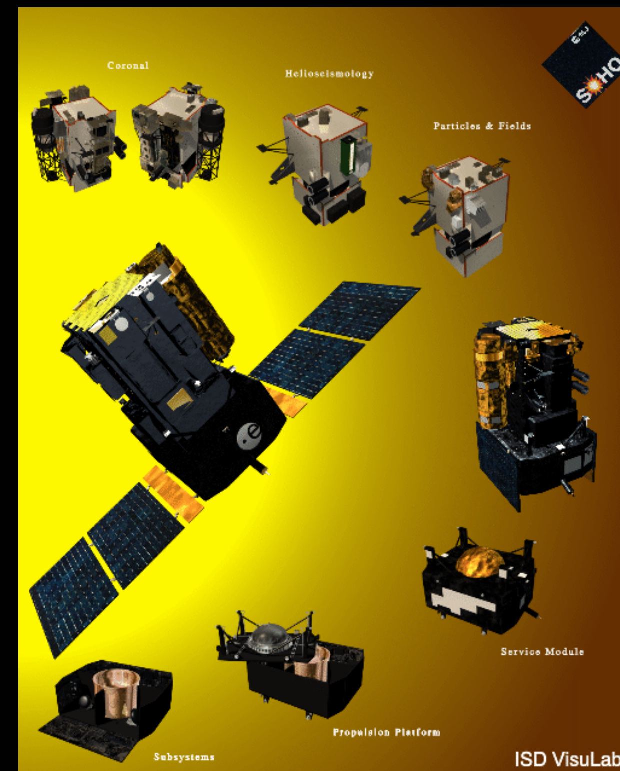
Background

Ariane 5

Courtesy of Arianespace / ESA / CSG.
Used with permission.



Mars Climate Orbiter



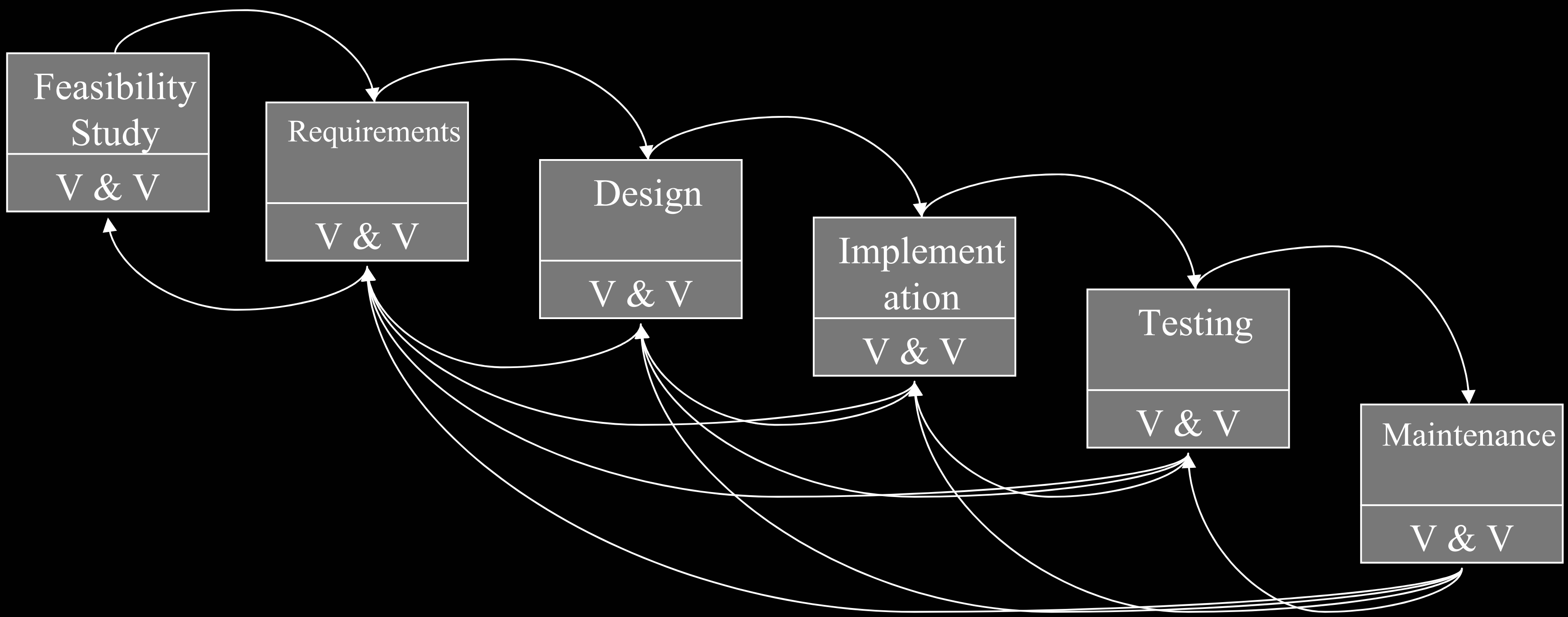
SOlar Heliospheric Observatory

Background

- Why is Software Engineering Hard?
  - "Curse of flexibility"
    - *"And they looked upon the software and saw that it was good. But they just had to add one other feature ..."*
    - No physical constraints
  - Intangibility
  - Lack of historical usage information
  - Organized complexity
    - Too complex for complete analysis
    - Too organized for statistics
  - Large discrete state spaces

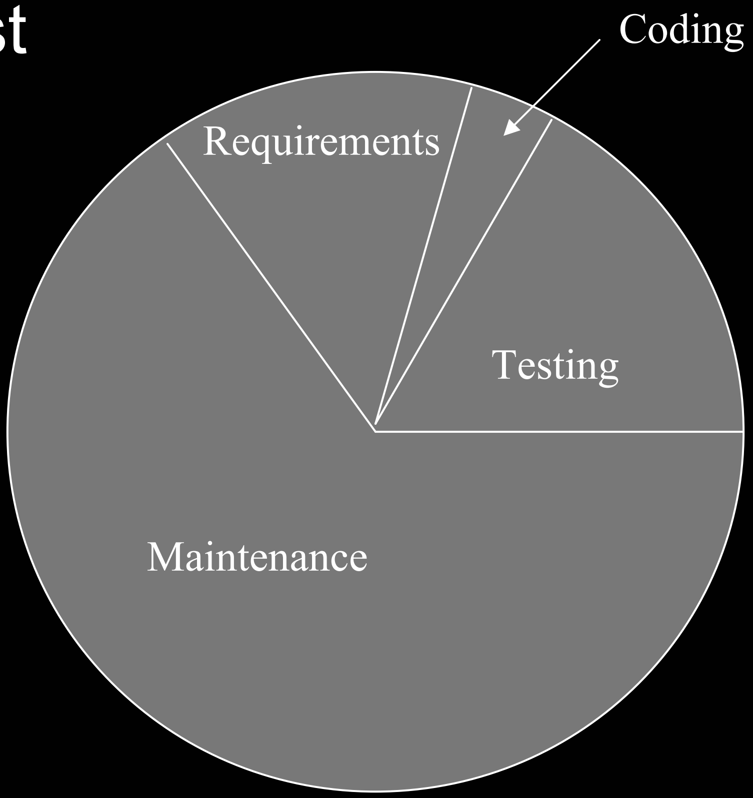
Background

- Software Lifecycle



Background

- Software Cost



Requirements Specification

- Most critical portion of the software lifecycle
- Majority of errors in software can be traced back to flaws in the requirements
- Many methods and types of requirements including:
  - Informal
    - English
    - UML
  - Formal
    - Zed
    - State Machines
    - Intent Specifications

Approaches to Design

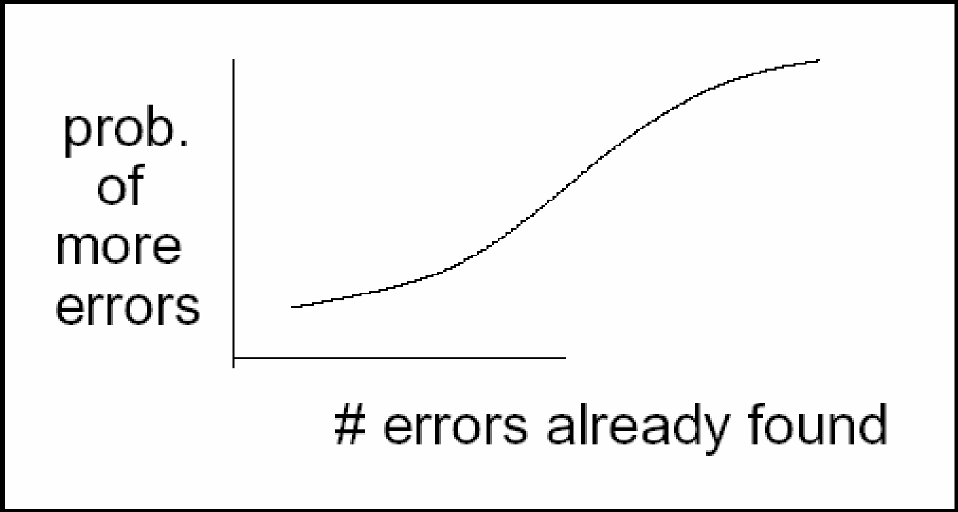
- Software design grew out of the structured programming movement beginning in the 1960s
- Many approaches to design including:
  - Functional Decomposition
  - Object-Orientation (OO)
  - Event-based CBSE
  - Agent Architectures
- What approach to Software Design is appropriate for Satellite Engineering?

Implementation

- Only 10% of the software development effort!!!
  - Other 90% made up of planning and testing
- Issues include:
  - Programming Languages
  - COTS and Reuse
  - Interfaces

Testing

- Examining a program to see if it does not do what it is supposed to do is only half the battle – the other half is seeing whether the program does what it is not supposed to do!



Maintenance

- Comprises approximately 70% of the software lifecycle cost and time
- Issues include:
  - Deployment and Training
  - Code Changes
    - Additional functionality
    - Fixing bugs
  - Diagnosis and Troubleshooting
  - Job Turnover – understanding someone else's code

Why is Software Engineering Hard for Spacecraft?

- Spacecraft Software Structure and a Lack of Autonomy
- Loss of Domain Knowledge
- Miscommunication Among Multi-disciplinary Engineering Teams

- Proposed Solution:
  - Component-Based Systems Engineering

SERL Approach

- Intent Specifications
  - Why? instead of merely What? and How?
  - Design Rationale
- SpecTRM
  - Specification Toolkit and Requirements Methodology
- SpecTRM-RL
  - SpecTRM-Requirements Language

SERL Approach (Cont.)

- Level 3 – SpecTRM-RL
  - Easily Readable and Reviewable
  - Unambiguous and uses simple semantics
- Complete
  - Can specify everything need to specify
- Analyzable
  - Executable
  - Formal (mathematical) foundation
  - Assists in finding incompleteness

Component-Based System Engineering

- Functional Decomposition
  - Spacecraft Level
    - Command and Data Handling Computer
  - Subsystem Level
    - Attitude Determination and Control
    - Power
    - Thermal
    - Communications
    - Guidance and Navigation
    - Propulsion

Component-Based System Engineering (Cont.)

- Top-Down Decomposition
  - Component Level
    - Ex) NEAR's Attitude Determination and Control Subsystem
      - Sun Sensors
      - Star Trackers
      - Inertial Measurement Units
      - Reaction Control Systems
      - Reaction Wheels

Component-Based System Engineering (Cont.)

- Construct software and hardware intent specifications from the component level to the system level
- Specification Toolkit and Requirements Methodology – Generic Spacecraft Component (SpecTRM-GSC)
  - Fully Encapsulated
  - Well-defined Interfaces
  - Generic
  - Component-level Fault Protection

Component-Based Systems Engineering (Cont.)

- Instead of performing CBSE, engineers can perform Component-Based *Systems* Engineering, in which the entire process of development (from the component-level to the system-level) is reused
- Benefits:
  - Provides the benefits of Component-Based Software Engineering without the detrimental effects of improper implementation of reuse
  - Supports the principles of systems engineering:
    - Common means of communication
    - Placing the component in context within the larger system

Component-Based System Engineering (Cont.)

- The development is performed in a systems engineering development environment (SpecTRM)
- Benefits:
  - Helps capture domain knowledge through recording rationale
  - Abstracts away the details of design
  - Provides various analyses
    - Simulate design alternatives
    - Nothing has been implemented at this point
  - Easy to incorporate changes to the software
  - Visualizations provide different perspectives on the same system

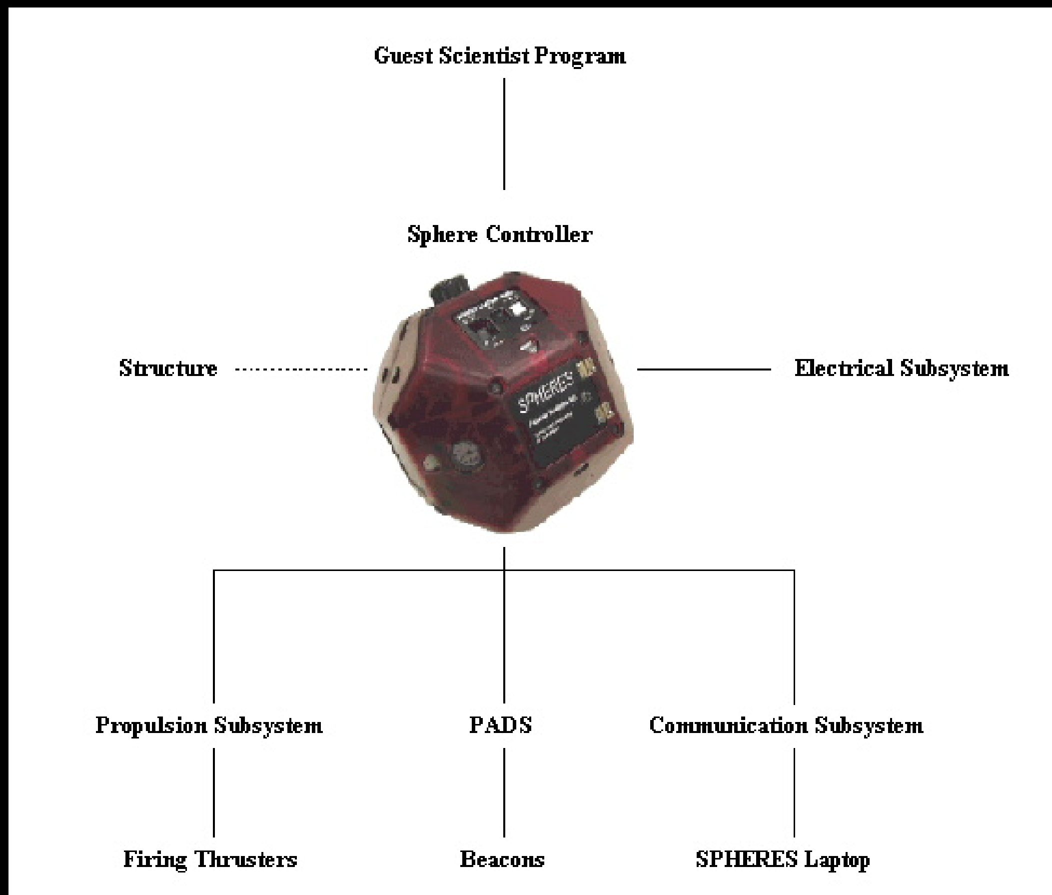
SPHERES

- Synchronized Position Hold Engage Reorient Experimental Satellites

Why SPHERES?

1. Autonomous
2. Highly modular
3. Test technique on a real system

SPHERES (Cont.)

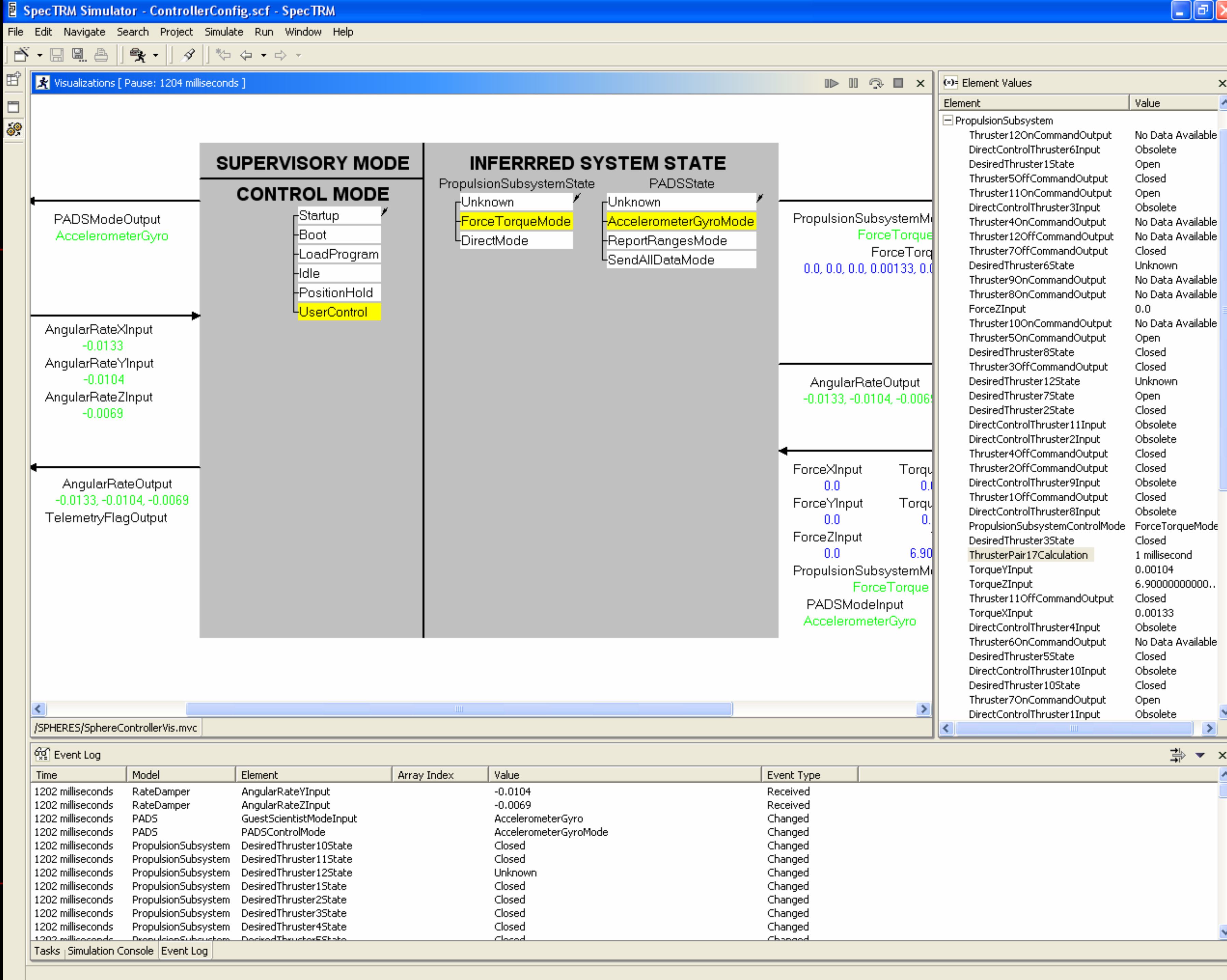


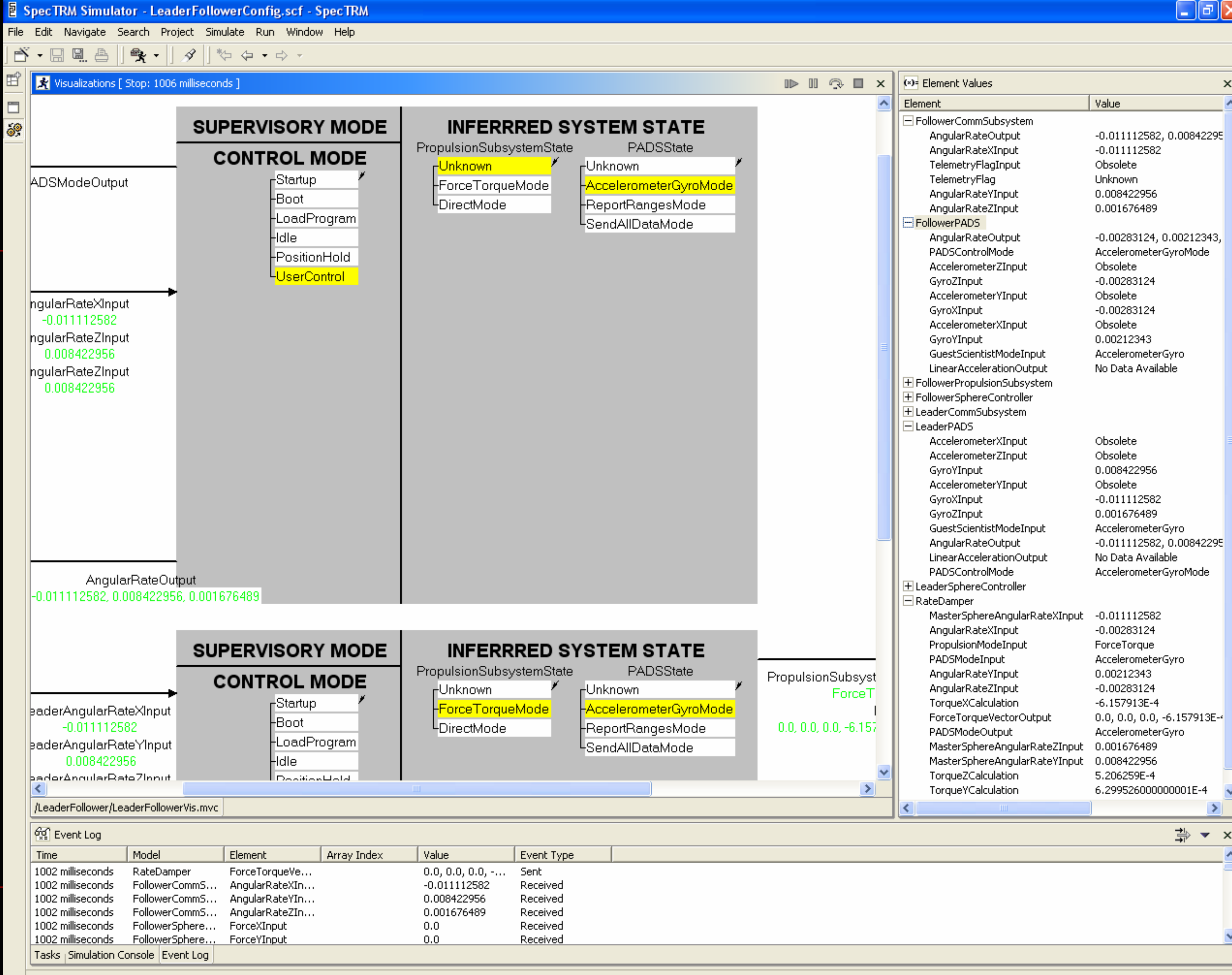
SPHERES (Cont.)

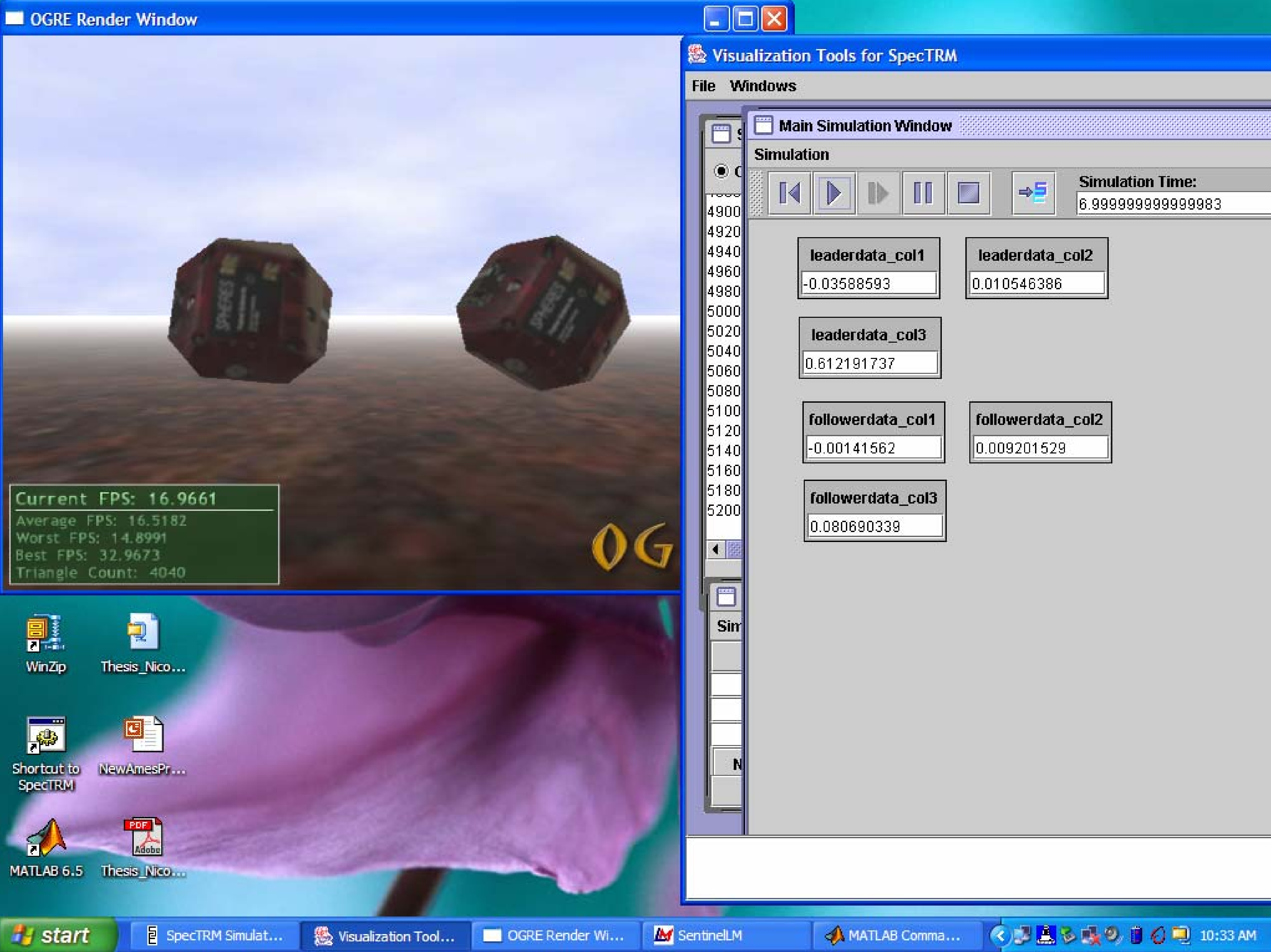
- Two Guest Scientist Programs were modeled to illustrate:
  - The feasibility/scalability of the technique
  - The ease with which the components can be reused
  - The process of building a new spacecraft configuration from already existing components

SPHERES (Cont.)

- Rate Damper
  - One Sphere Configuration
  - Nullifies any angular rate experienced by the Sphere
- Leader/Follower (Rate Matcher)
  - Two Sphere Configuration
  - Follower Sphere matches the angular rate experienced by the Leader Sphere
- Demonstration


SpecTRM Simulator - ControllerConfig.scf - SpecTRM
File Edit Navigate Search Project Simulate Run Window Help
Visualizations [Pause: 1204 milliseconds]
SUPERVISORY MODE
CONTROL MODE
Startup
Boot
LoadProgram
Idle
PositionHold
UserControl
INFERRRED SYSTEM STATE
PropulsionSubsystemState
Unknown
ForceTorqueMode
DirectMode
PADSState
Unknown
AccelerometerGyroMode
ReportRangesMode
SendAllDataMode
PADSModeOutput
AccelerometerGyro
AngularRateXInput
-0.0133
AngularRateYInput
-0.0104
AngularRateZInput
-0.0069
AngularRateOutput
-0.0133, -0.0104, -0.0069
TelemetryFlagOutput
ForceTorque
0.0, 0.0, 0.0, 0.00133, 0.0
AngularRateOutput
-0.0133, -0.0104, -0.006
ForceXInput
0.0
ForceYInput
0.0
ForceZInput
0.0
6.90
ForceTorque
PADSModeInput
AccelerometerGyro
/SPHERES/SphereControllerVis.mvc
Element Values
Element Value
PropulsionSubsystem
Thruster12OnCommandOutput No Data Available
DirectControlThruster6Input Obsolete
DesiredThruster1State Open
Thruster5OffCommandOutput Closed
Thruster11OnCommandOutput Open
DirectControlThruster3Input Obsolete
Thruster4OnCommandOutput No Data Available
Thruster12OffCommandOutput No Data Available
Thruster7OffCommandOutput Closed
DesiredThruster6State Unknown
Thruster9OnCommandOutput No Data Available
Thruster8OnCommandOutput No Data Available
ForceZInput 0.0
Thruster10OnCommandOutput No Data Available
Thruster5OnCommandOutput Open
DesiredThruster8State Closed
Thruster3OffCommandOutput Closed
DesiredThruster12State Unknown
DesiredThruster7State Open
DesiredThruster2State Closed
DirectControlThruster11Input Obsolete
DirectControlThruster2Input Obsolete
Thruster4OffCommandOutput Closed
Thruster2OffCommandOutput Closed
DirectControlThruster9Input Obsolete
Thruster1OffCommandOutput Closed
DirectControlThruster8Input Obsolete
PropulsionSubsystemControlMode ForceTorqueMode
DesiredThruster3State Closed
ThrusterPair17Calculation 1 millisecond
TorqueYInput 0.00104
TorqueZInput 6.90000000000..
Thruster11OffCommandOutput Closed
TorqueXInput 0.00133
DirectControlThruster4Input Obsolete
Thruster6OnCommandOutput No Data Available
DesiredThruster5State Closed
DirectControlThruster10Input Obsolete
DesiredThruster10State Closed
Thruster7OnCommandOutput Open
DirectControlThruster1Input Obsolete
Event Log
Time Model Element Array Index Value Event Type
1202 milliseconds RateDamper AngularRateYInput -0.0104 Received
1202 milliseconds RateDamper AngularRateZInput -0.0069 Received
1202 milliseconds PADS GuestScientistModeInput AccelerometerGyro Changed
1202 milliseconds PADS PADSControlMode AccelerometerGyroMode Changed
1202 milliseconds PropulsionSubsystem DesiredThruster10State Closed Changed
1202 milliseconds PropulsionSubsystem DesiredThruster11State Closed Changed
1202 milliseconds PropulsionSubsystem DesiredThruster12State Unknown Changed
1202 milliseconds PropulsionSubsystem DesiredThruster1State Closed Changed
1202 milliseconds PropulsionSubsystem DesiredThruster2State Closed Changed
1202 milliseconds PropulsionSubsystem DesiredThruster3State Closed Changed
1202 milliseconds PropulsionSubsystem DesiredThruster4State Closed Changed
Tasks Simulation Console Event Log


SpecTRM Simulator - LeaderFollowerConfig.scf - SpecTRM
File Edit Navigate Search Project Simulate Run Window Help
Visualizations [Stop: 1006 milliseconds]
SUPERVISORY MODE
CONTROL MODE
Startup
Boot
LoadProgram
Idle
PositionHold
UserControl
INFERRRED SYSTEM STATE
PropulsionSubsystemState
Unknown
ForceTorqueMode
DirectMode
PADSState
Unknown
AccelerometerGyroMode
ReportRangesMode
SendAllDataMode
ADSModeOutput
ngularRateXInput
-0.011112582
ngularRateZInput
0.008422956
ngularRateZInput
0.008422956
AngularRateOutput
-0.011112582, 0.008422956, 0.001676489
SUPERVISORY MODE
CONTROL MODE
Startup
Boot
LoadProgram
Idle
INFERRRED SYSTEM STATE
PropulsionSubsystemState
Unknown
ForceTorqueMode
DirectMode
PADSState
Unknown
AccelerometerGyroMode
ReportRangesMode
SendAllDataMode
eaderAngularRateXInput
-0.011112582
eaderAngularRateYInput
0.008422956
PropulsionSubsyst
ForceT
0.0, 0.0, 0.0, -6.157
/LeaderFollower/LeaderFollowerVis.mvc
Element Values
Element Value
FollowerCommSubsystem
AngularRateOutput -0.011112582, 0.0084229
AngularRateXInput -0.011112582
TelemetryFlagInput Obsolete
TelemetryFlag Unknown
AngularRateYInput 0.008422956
AngularRateZInput 0.001676489
FollowerPADS
AngularRateOutput -0.00283124, 0.00212343,
PADSControlMode AccelerometerGyroMode
AccelerometerZInput Obsolete
GyroZInput -0.00283124
AccelerometerYInput Obsolete
GyroXInput -0.00283124
AccelerometerXInput Obsolete
GyroYInput 0.00212343
GuestScientistModeInput AccelerometerGyro
LinearAccelerationOutput No Data Available
FollowerPropulsionSubsystem
FollowerSphereController
LeaderCommSubsystem
LeaderPADS
AccelerometerXInput Obsolete
AccelerometerZInput Obsolete
GyroYInput 0.008422956
AccelerometerYInput Obsolete
GyroXInput -0.011112582
GyroZInput 0.001676489
GuestScientistModeInput AccelerometerGyro
AngularRateOutput -0.011112582, 0.0084229
LinearAccelerationOutput No Data Available
PADSControlMode AccelerometerGyroMode
LeaderSphereController
RateDamper
MasterSphereAngularRateXInput -0.011112582
AngularRateXInput -0.00283124
PropulsionModeInput ForceTorque
PADSModeInput AccelerometerGyro
AngularRateYInput 0.00212343
AngularRateZInput -0.00283124
TorqueXCalculation -6.157913E-4
ForceTorqueVectorOutput 0.0, 0.0, 0.0, -6.157913E-
PADSModeOutput AccelerometerGyro
MasterSphereAngularRateZInput 0.001676489
MasterSphereAngularRateYInput 0.008422956
TorqueZCalculation 5.206259E-4
TorqueYCalculation 6.299526000000001E-4
Event Log
Time Model Element Array Index Value Event Type
1002 milliseconds RateDamper ForceTorqueVe... 0.0, 0.0, 0.0, -... Sent
1002 milliseconds FollowerCommS... AngularRateXIn... -0.011112582 Received
1002 milliseconds FollowerCommS... AngularRateYIn... 0.008422956 Received
1002 milliseconds FollowerCommS... AngularRateZIn... 0.001676489 Received
1002 milliseconds FollowerSphere... ForceXInput 0.0 Received
1002 milliseconds FollowerSphere... ForceYInput 0.0 Received
Tasks Simulation Console Event Log


OGRE Render Window
Current FPS: 16.9661
Average FPS: 16.5182
Worst FPS: 14.8991
Best FPS: 32.9673
Triangle Count: 4040
Visualization Tools for SpecTRM
File Windows
Main Simulation Window
Simulation
Simulation Time:
6.999999999999983
leaderdata_col1
-0.03588593
leaderdata_col2
0.010546386
leaderdata_col3
0.612191737
followerdata_col1
-0.00141562
followerdata_col2
0.009201529
followerdata_col3
0.080690339
4900
4920
4940
4960
4980
5000
5020
5040
5060
5080
5100
5120
5140
5160
5180
5200
WinZip
Thesis_Nico...
Shortcut to SpecTRM
NewAmesPr...
MATLAB 6.5
Thesis_Nico...
start
SpecTRM Simulat...
Visualization Tool...
OGRE Render Wi...
SentinelLM
MATLAB Comma...
10:33 AM

Conclusions

- The research on and the test case application of Component-Based Systems Engineering show its potential for use in developing the next generation of spacecraft
- The benefits of using the technique span not only the engineering issues faced by today's spacecraft development teams but also the difficulties inherent in the aerospace industry

Questions and Comments