Issues in Assessing Reliability Risk for Software-Intensive Space Systems

Allen P. Nikora
Jet Propulsion Laboratory,
California Institute of Technology
Allen.P.Nikora@jpl.nasa.gov

Henk Roelant
NASA Johnson Space Center
Henk.Roelant-1@nasa.gov
Nelson Thompson

JSC-NC, SAIC
Nelson.A.Thompson@nasa.gov

Mike Yau
ASCA, Inc.
mike.yau@ascainc.com

Copyright 2010 California Institute of Technology.
Government sponsorship acknowledged.

Agenda

National Aeronautics and Space Administration

- Overview
- Assessing System Reliability Risk
- Data Issues
- Tool Issues
- Discussion
- Backup Material
  - Mini AerCam Example

Overview

National Aeronautics and Space Administration

- As software performs critical functions for space mission systems (e.g, on-board fault detection, identification, and repair), accurate determination/prediction of how system reliability is affected by its software components is increasingly important to minimize the risk of inopportune system failure.
  - Numerous techniques for assessing software reliability exist and have been implemented in tools.
  - Several issues must be addressed to correctly incorporate software reliability assessments into an overall assessment of system reliability risk.
- The reliability of software components needs to be established in the context in which the system of which they are a part is expected to operate.
  - May require construction of more than one system model corresponding to different phases of a mission, as software components may operate differently during each phase (e.g., launch, cruise, encounter, and observation phases).

Overview (cont'd)

National Aeronautics and Space Administration

- Techniques and tools used to assess the reliability of software components must be clearly understood.
  - Software reliability growth models (SRGMs) used during later phases of testing usually have two or three parameters that are clearly defined and can be estimated from available failure data during test.
    - Some exceptions (e.g., Littlewood-Verrall model parameter quantifying "goodness of programmer").
  - Techniques used during earlier development phases may require several dozen inputs characterizing development process and the software itself. Each of these inputs must be unambiguously defined and measurable if there is to be confidence in the results.
- Techniques and tools for assessing software reliability risk should be used only for the purpose for which they are intended.
  - For example, models that estimate software defect content and failure intensity at the start of mission operations may be appropriate for software systems that operate over long durations in the same fashion.
  - Same models may not be appropriate for estimating the reliability of software that is operated at irregular intervals on a demand basis (e.g., fault identification and repair software).

Assessing System Reliability Risk

National Aeronautics and Space Administration

- Historically, software has often been ignored when assessing reliability risk for systems that include software components
  - Software reliability assumed to be 1.
  - Effects of software failure on other system components are not detailed. For example, fault trees may simply identify a generic "software fault" as one of the leaves of a subtree instead of going into the same level of detail as for hardware components.
  - …
- Many techniques and tools are available to quantitatively/qualitatively assess software failure risk:
  - SRGMs (implemented by tools such as SMERFS, CASRE).
  - "Predictive" models based on measurable development process/software artifact characteristics can yield predictions of defect content during early software development phases.
  - Architectural analysis via Markov/semi-Markov models developed from software call graph (tools such as SHARPE can perform this analysis).

Assessing System Reliability Risk (cont'd)

National Aeronautics and Space Administration

- **Many techniques and tools are available to quantitatively/qualitatively assess software failure risk (cont'd):**
  - **Software Fault Tree Analysis (SFTA) and Software Failure Modes and Effects Analysis (SFMEA)**
    - **If relevant data is available, probabilities of failure events can be estimated.**
  - **Analytical verification of specifications (e.g., model checking). Can indirectly provide quantitative analysis of software failure risk (e.g., out of "*N*" computation trees checked, "*n*" property violations were detected).**
- **Assessing system risk requires assessing hardware and software associated risks in system's anticipated operational context and combining them in an appropriate system model.**
  - **Risk model(s) must be based on anticipated mission profile**
  - **Hardware components must consider effects of incorrect signals/commands from software in addition to physics-based failure mechanisms (e.g., aging, thermal stress, radiation damage)**
    - **For example, can an instrument be damaged by a command to the instrument to move an optical component into the stops?**
  - **Software components need to consider potential hardware failure modes and operator error that may provide unexpected/illegal inputs in addition to defects inserted into the software.**

Assessing System Reliability Risk (cont'd)

National Aeronautics and Space Administration

- Current work investigates feasibility of integrating currently available software reliability risk assessment techniques with PRA or PRA-like modeling techniques.
  - Assess risk of software-involved failure for mission-critical CxP mission scenario.
  - CSRM chosen as system-level modeling technique.
    - Also used for lower-level subsystem modeling.
  - Model construction follows canonical principles of constructing PRA models:
    - Top-level models are constructed as event trees. Events are based on critical events defined in the high-level mission description.
    - Intermediate and lower level models can be constructed using fault trees or other techniques. Dynamic Flowgraph Methodology was chosen to develop these models:
      - Allows for multi-valued logic.
      - Control loops of the type found in on-board software systems (e.g., guidance and navigation) are easily modeled in DFM.
      - Analysis is similar to that for fault trees.
    - Failure probabilities in DFM sub-models are linked to pivot events in top-level model(s).
      - See example taken from "RISK-INFORMED SAFETY ASSURANCE AND PROBABILISTIC RISK ASSESSMENT OF MISSION-CRITICAL SOFTWARE-INTENSIVE SYSTEMS".

Data Issues

National Aeronautics and Space Administration

- Experience in modeling CxP scenarios is that obtaining accurate and relevant data is the most important factor in producing useful models and predictions.
- Three general types of data are available:
  - Product Characterization
    - Application type
    - Size
    - Language
    - …
  - Development process characterization
    - Configuration management
    - Problem reporting and tracking
    - Technical and milestone reviews
    - …
  - Operational environment characterization
    - Input space characterization (ranges, use frequencies, ordering)
    - System characteristics
      - Processor speed
      - Component interactions (e.g, how many tasks are running concurrently?)
      - …
    - Failure history
    - …
- SRGMs use primarily data in the third group.
- Assessment techniques that can be used prior to test and implementation may use data in all three groups.

Data Issues (cont'd)

National Aeronautics and Space Administration

- Experience indicates that operational data can be collected with good accuracy
  - AT&T
  - Microsoft
  - JPL (project by project basis)
  - …

  Relevant operational data is collected by most development organizations of which the authors are aware, but it's not available until late in the development life cycle (e.g., integration testing).

- Data in second group appears to be most difficult to acquire:
  - May not be quantitative.
    - May be alleviated by using level-valued data (e.g., "required reliability is low, nominal, high, or very high").
  - Development processes can be difficult to characterize.
  - Data may be subjective. Introduces uncertainty, nonrepeatability in measurements.
    - Interview multiple respondents on same project,
    - Reconcile differences between interviewees,
    - Repeat as necessary.

Tool and Technique Issues

National Aeronautics and Space Administration

- **SRGMs can assess software reliability risk based on failure history observed during test and operations**
  - **Advantages**
    - **Models are simple (often 2 or 3 parameters) and easily understandable.**
    - **For many models initial estimates of the parameters can be made by direct measurement.**
    - **Can be used to estimate reliability risk for both time-based and demand-based systems.**
    - **Several tools available (e.g., SMERFS, CASRE)**

Tool and Technique Issues (cont'd)

National Aeronautics and Space Administration

- SRGMs can assess software reliability risk based on failure history observed during test and operations (cont'd)
  - Potential Issues
    - Black box.
    - Usable only after software has been implemented and is running.
    - Operational profile must be defined if assessments made during test can be related to anticipated operational use
    - Lower limit to failure intensity estimates: $O(10^{-4})$ failures/hour.
      - Too much testing required to accurately estimate lower values.
      - Noise in data may prevent accurate estimates of lower values.
    - Some tools nearing end of useful life
      - CASRE does not run under newer versions of Windows, does not include some newer models and model recalibration techniques

Tool and Technique Issues (cont'd)

National Aeronautics and Space Administration

- Early Assessment Techniques/Tools
  - Many estimate defect content/failure intensity based on measurable development process/software artifact characteristics.
  - Those known to the authors have been developed using empirical data from past and present software development efforts.
    - Depending on tool, data used to developed empirical model(s) implemented in tool may not be known.
      - Increases difficulty of determining whether model/tool is relevant to current development effort.
    - Accuracy, uncertainty of data used to develop tool's models may also be unknown.
      - Applies to proprietary as well as non-proprietary tools/techniques.

Tool and Technique Issues (cont'd)

National Aeronautics and Space Administration

- Early Assessment Techniques/Tools (cont'd)
  - Those known to the authors have been developed using empirical data from past and present software development efforts (cont'd).
    - For many tools, the data used to develop the model(s) is primarily based on artifact and development process characteristics. Operational context is often not included.
  - Computations made by some tools may be invisible to users
    - Some tools make estimates by comparing measurements for a current project with the "closest" match in the empirical database used to implement the tool. What does "closest" mean?
  - Use of these techniques for demand-based systems may not be appropriate.

Discussion

- **Lessons learned to date:**
  - **Most significant: Relevant measurement mechanisms need to be set up at the institutional as well as development effort levels.**
    - **Ensure that quantitative goals are defined before setting up/implementing measurement mechanism.**
    - **Use GQM to guide measurement.**
    - **Measurements of the context in which the system will be operating need to be included.**
  - **Tools used to perform the assessment need to be:**
    - **Consistent with defined goal(s).**
      - **For example, don't use a tool that makes assessments only for time-based systems for assessing demand-based ones.**
    - **Consistent with defined measurements.**
  - **Users need to be able to understand the model(s) implemented in a tool if they are to have an informed opinion of whether or not to use it.**
  - **Users also need to understand the way in which different tools can interact to achieve the stated goal(s). Our experience indicates it is likely that a single tool will not be able to perform all of the analyses required to satisfy the goal(s).**

Backup Material

National Aeronautics and Space Administration

- CSRM Mini AerCam example

Mini AerCam Example

- **Overview**

The Miniature Autonomous Extravehicular Robotic Camera (Mini AERCam) was intended as a free flying satellite to provide flexible remote viewing capabilities in support of manned-space missions. In a nominal mission, it would have been released from the cargo bay of the Space Transportation System (STS). Its main function was to provide a color video orthogonal view to support the use of the International Space Station (ISS) robotic arm. Some features of the Mini AERCam nanosatellite included:

- Capability to transmit color NTSC video and high resolution still images,
- Six degrees of freedom motion,
- Manually controlled via joysticks.
- Autonomously perform point-to-point movement, absolute position hold, and relative station keeping.

Its predecessor, the AERCam Sprint, was successfully tested on a shuttle mission. The Mini AER-Cam was intended to provide a superset of the Sprint capabilities while using Commercial Off-the-Shelf (COTS) technology within a smaller, less expensive and more robust package.

Return to Assessing Reliability Risk

Mini AerCam Example

National Aeronautics and Space Administration

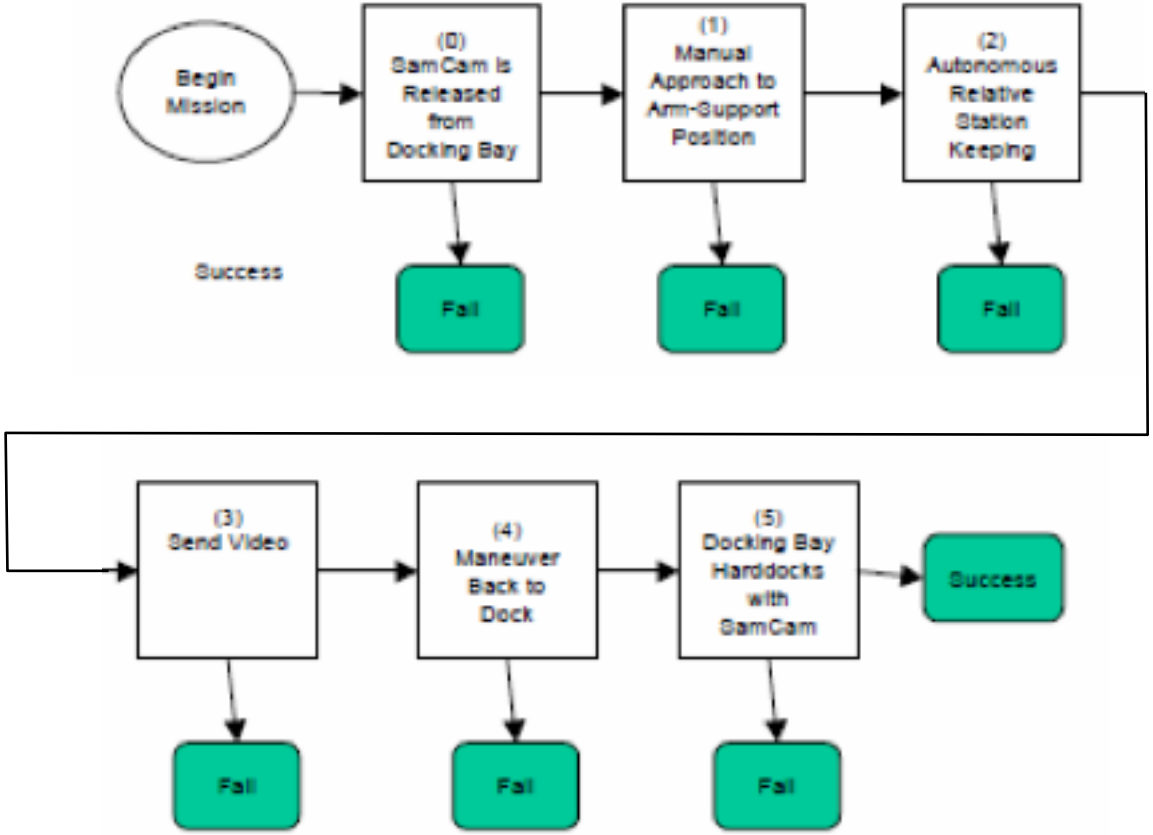
Typical mission consists of the following phases:

- **Release from the docking bay,**
- **Autonomous control of the Mini AERCam to reach the vicinity of the target position,**
- **Autonomous station keeping to maintain relative position with the target, so as to carry out the video capture and transmission functions,**
- **Autonomous control of the Mini AERCam to return to the docking bay, and**
- **Retrieval of the Mini AERCam into the docking bay.**

Return to Assessing Reliability Risk

Mini AerCam Example

National Aeronautics and Space Administration

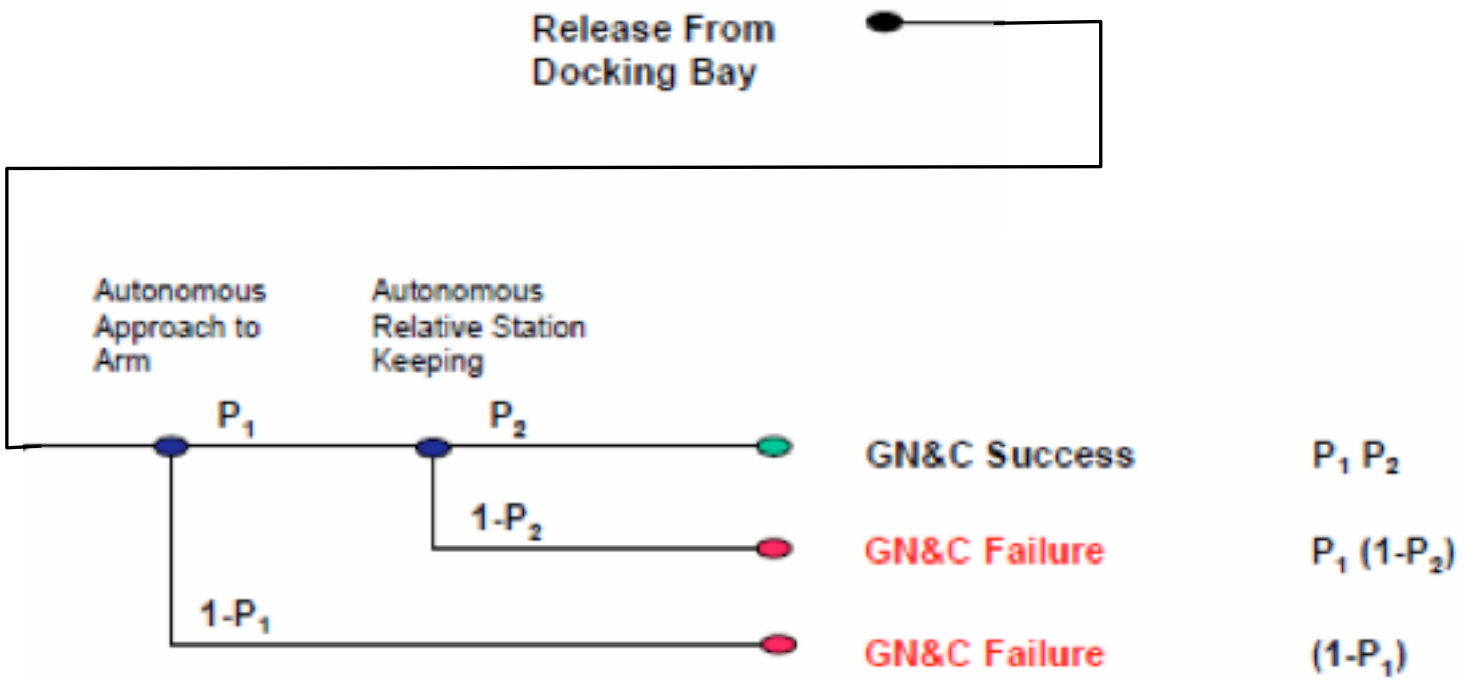


Mini AERCam Mission Tree

Return to Assessing Reliability Risk

Mini AerCam Example

National Aeronautics and Space Administration

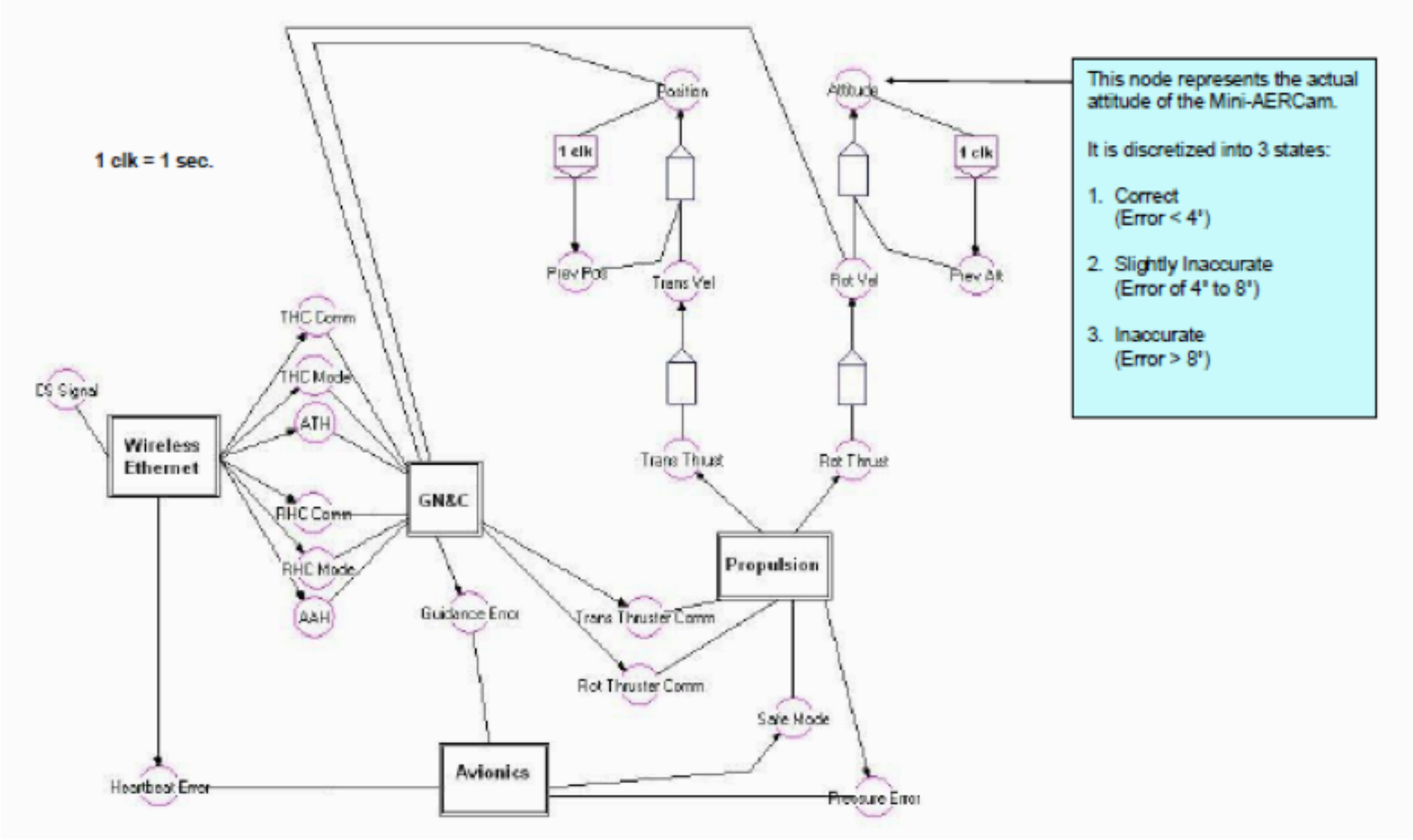


Mini AERCam Top-Level Event Tree

Return to Assessing Reliability Risk

Mini AerCam Example

National Aeronautics and Space Administration



Mini AERCam Top-Level DFM Model

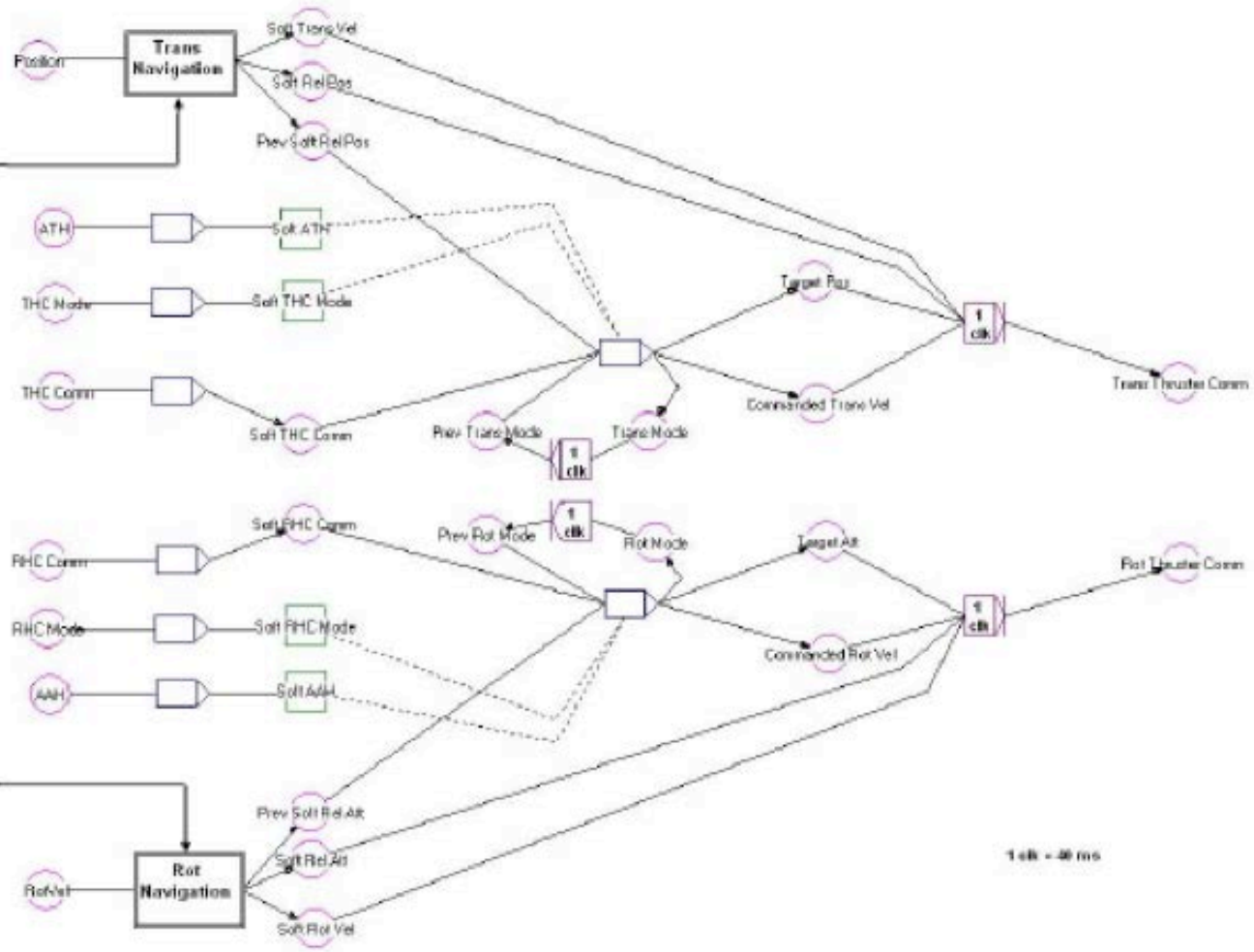
Return to Assessing Reliability Risk

Mini AerCam Example

National Aeronautics and Space Administration

This sub-model includes the GPS hardware and the translational navigation software.

This sub-model includes the angular rate gyro hardware and the rotational navigation software.



Intermediate-Level DFM Model of the GN&C Subsystem

Return to Assessing Reliability Risk