SSC18-WKV-04

Model-Based Assurance for Satellites with Commercial Parts in Radiation Environments

A. Witulski, B. Sierawski, R. Austin, G. Karsai, N. Mahadevan, R. Reed, R. Schrimpf,
Electrical Engineering and Computer Science Department, Vanderbilt University
PMB 351824, 2301 Vanderbilt Place, Nashville, TN 37235-182, USA
Arthur Witulski: Tele: 615 343 8833, arthur.f.witulski@vanderbilt.edu

Ken LaBel
NASA Electronic Parts and Packaging Program
NASA Goddard Space Flight Center, Greenbelt, MD 20771, USA
kenneth.a.label@nasa.gov

John Evans
NASA Office of Safety and Mission Assurance, NASA Headquarters
NASA Headquarters, Washington, D.C., 20024, USA
john.w.evans@nasa.gov

Philippe Adell
Jet Propulsion Laboratory, California Institute of Technology
4800 Oak Grove Drive, Pasadena, CA, 91109
Philippe.C.Adell@jpl.nasa.gov

ABSTRACT

Small satellite projects often do not have the budget or schedule to incorporate radiation-hardened parts or extensive radiation test campaigns into their schedule. Yet a case must be made that the spacecraft will function as intended in orbit, with radiation, temperature and vacuum affecting part performance. The Vanderbilt Institute for Space and Defense Electronics, with support from NASA HQ, NASA NEPP, and NASA JPL, has developed a platform for making a safety case for systems with commercial (non-hardened) parts, called the Systems Engineering Assurance and Modeling (SEAM) platform. The platform has three elements: goal structuring notation (GSN), systems engineering models (SysML and our extensions), and Bayesian networks (BN). The GSN is a visual argument structure that presents an argument that the system meets specifications based on goals, strategies, and evidence. The systems engineering model is a high-level descriptive language that captures the spacecraft design and system architecture through various diagrams. We extend the SysML diagram set to include fault propagation diagrams, which map the environment, failure manifestations, anomalies, failure effects and responses (mitigation measures) of components and systems. The SEAM platform provides a low-cost alternative to conventional radiation hardening assurance paradigms.

1.0 INTRODUCTION

CubeSats and other small satellites present a great opportunity for getting experiments into space quickly at low cost. However, that short schedule and low budget can lead to a high on-orbit failure rate because conventional radiation hardening and reliability procedures are typically not feasible during their development. CubeSats typically use mostly commercial-off-the-shelf (COTS) parts, not radiation-hardened parts. Many first-time CubeSat designers and developers are not aware of the impact that radiation and temperature can play in the performance of electronics in space. The low budget precludes extensive ground-based radiation testing and the use of rad-hard, space-qualified parts. The short schedule precludes the extensive documentation, lot-acceptance testing and reliability analysis typically used on NASA Class A missions. Consequently, CubeSats have a much higher failure rate in space than conventional missions (Fig.1). In this paper, we present an alternative to conventional radiation-hardening and mission-assurance paradigms, based on model-based system engineering and mission assurance, that can be performed much more quickly and with lower resource expenditure than conventional radiation hardening paradigms. We begin with a short review of radiation effects and describe the CubeSate experiment that we apply the assurance method to. We then describe the three fundamental aspects of the System Engineering and Assurance Modeling (SEAM) platform: modeling of the system architecture,

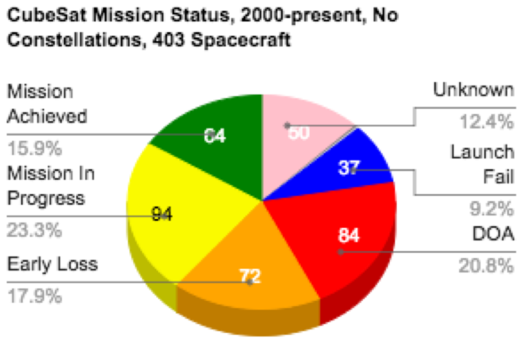


Figure 1. CubeSat Success/Failure Rate [1]

construction of a radiation safety case through a graphical argument construct, and construction of Bayesian nets to estimate relative probabilities of fault impacts on system performance. We then discuss the integrated assurance platform and its embodiment in a web-based application on a public website.

2.0 RADIATION EFFECTS

2.1 Total Ionizing Dose

Radiation exposure produces relatively stable, long-term changes in device and circuit characteristics that may result in parametric degradation or functional failure. The total ionizing dose primarily impacts insulating layers. In state-of-the-art MOS integrated circuits (ICs), effects of radiation-induced charge on gate oxides are small, but field oxides and isolation structures are usually less radiation-tolerant than the active device regions. Total ionizing dose is specified in units of rads, which are units of absorbed energy per unit mass for a given material, for example, rad(SiO_2)

The behavior of these thick field oxides and shallow trench isolation regions dominates the radiation response of most unhardened CMOS integrated circuits. As positive charge is trapped in these oxides, negative charge is induced in the nearby Si. For p-type substrates or wells, an inversion layer will form when the positive charge density in the oxide becomes sufficiently high. The inversion layer can short the source and drain of a transistor together at the edge of the active area. It also is possible to invert the field region between adjacent n-channel devices, leading to leakage currents between the drain of one transistor and the source of another. These mechanisms are described in greater detail, with examples, in [3-5, 15].

The transistor-level leakage current shows up as a large increase in power supply current if significant portions of the die are affected. The net charge created by ionizing radiation is almost always positive, so field inversion is only a problem for n-channel MOSFETs. This effect is illustrated in Fig. 2, which shows the supply current vs. total ionizing dose for a D-Flipflop.

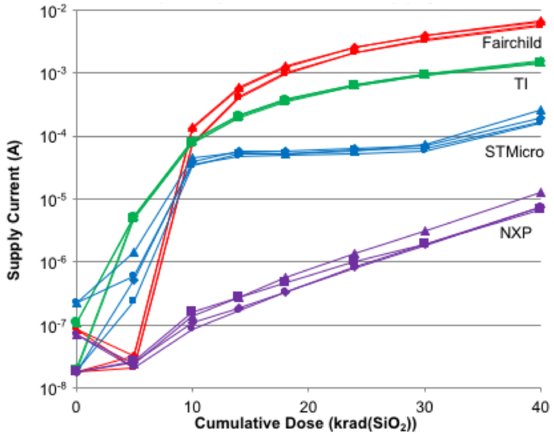


Figure 2. Impact of TID on supply current of D-Flipflop, same part number, notice variation with instance and manufacturer [2].

2.2 Single Event Effects and Latch-up

"Single Event Effects" (SEE) is an umbrella term for various effect that occur when an energetic ion deposits energy in a semiconductor device or integrated circuit. An ion from any element with atomic mass greater than that of helium is considered a "heavy ion." The sources of ions are the sun and galactic cosmic rays. In addition, the Van Allen belts around the earth consist of electrons or protons trapped in the earth's magnetic field. The actual "radiation environment," or concentrations and fluxes of ions of various species, of charged particles is a function of orbit or position in space, e.g., the lunar radiation enrionment is quite different from low earth orbit around earth. A detailed introduction to single events is given in [6].

When an ion passes through a semiconductor in an integrated circuit, it deposits energy in the semiconductor lattice, which ionizes the atoms in the lattice and liberates holes and electrons, creating mobile charge in the device. This mobile charge can move to the terminals of the transistor or diode and cause current and voltage transients. The transients cause a wide range of effects including single event upset of bits in memories or latches (SEU), single event transient (SET) pulses, and single event functional interrupts (SEFI) in FPGAs and microcontrollers, which cause the digital device control logic to hang up in an undesired state. These events are usually non-destructive.

Destructive single event effects include single event burnout (SEB) in power transistors, and single event

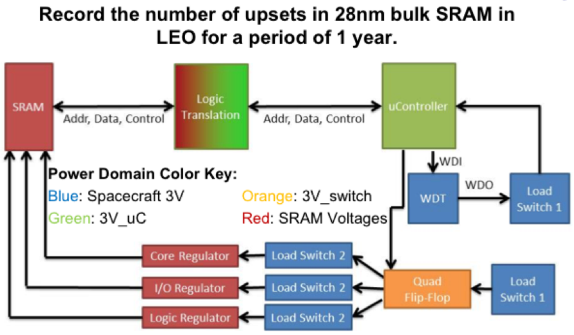


Figure 3: Top-level schematic of the SRAM CubeSat radiation experiment REM board [7].

latchup (SEL)in bulk CMOS devices. Latchup is common in COTS parts, since many of them are manufactured with bulk (not SOI) CMOS processes. In a latchup event, a regenerative condition occurs in the p-n-p-n structure of the PMOS, the p-well, the n-well, and the NMOS that causes the devices to be permanently on, thus shunting current directly from the power supply Vdd to ground. A sustained latch condition can result in thermal damage to the integrated circuit and can only be interrupted by turning off the power for a short time.

3.0 CUBESAT EXAMPLE: SINGLE EVENT UPSETS IN SRAM

CubeSats make extensive use of commercial-off-the-shelf (COTS) components that were never intended to operate in space. As a consequence, these components are subject to single event effects and total ionizing dose failures. The electronic component that is often the focus of research on single event upsets is a Static Random Access Memory (SRAM). These memories are ubiquitous and are extensively used by microcontrollers and microprocessors to operate of program data. The memory provides volatile storage of data, which can be individually addressed and written or read. Each SRAM cell stores a single bit of data in a bi-stable circuit. These memory arrays tend to be designed with the smallest available transistors, are highly integrated, and have large storage capacity. These three factors make them suitable for easily and quickly identifying single event effects.

In the last 10 years a number of investigators have discovered that very advanced memories are susceptible to upset from proton ionization [16]. The possibility that on-orbit bit error rates might increase because of ionization from protons is disruptive to our reliability assurance techniques. The pressing question is how much of the overall error rate will this account for and do we need to change the way that we assess the vulnerability of parts?

The CubeSat program at Vanderbilt was conceived to generate data for research on the effects of radiation on modern microelectronics. The RadFxSat platform is a system architecture designed to monitor single event effects and total ionizing dose in a component under test. The system conducts the experiments and interrogates the device, monitors and regulates power to the device, and reports the status of the device. On November 18, 2017, the AO-91 satellite was launched into orbit carrying the second such RadFxSat payload named Phoenix. The Phoenix payload contained three instances of the Radiation Effects Modeling (REM) experiment.

3.1 Radiation Effects Modeling (REM)

The REM experiment was developed at Vanderbilt University to report on the occurrence of single event upsets in a commercial 28 nm SRAM. Although the objective is to observe radiation effects in the device under test, peripheral circuitry required to operate the device should be should be immune to or capable of recovering from single event effects. Each board includes the device under test, a microcontroller and non-volatile memory to conduct the experiment, and a system of regulators and load switches to provide power and mitigate against potential permanent failures due to single event latchup. All components are required to meet the total ionizing dose screening tests self-imposed by the RadFxSat platform (> 30 krad(Si)).

During flight, the microcontroller handles reading and writing to the SRAM, counting the number of upsets, and communicating science data and health of the board through an I2C bus. The microcontroller writes a blanket checker-board pattern to the memories and checks the data after a 5-minute exposure. The total number of bits in error for the exposure is added to the mission total upsets and the experiment live-time is incremented. Both values are updated in the telemetry and another exposure begins. The two values together are sufficient to determine the upsets per bit per day.

In Figure 3, a block diagram of the REM experiment board is presented. The memory required multiple voltage levels and complicated the design of the experiment. The input power from the spacecraft is a regulated 3V rail (blue boxes). This 3V is divided to the different power domains by load switches to create a 3V_uC rail (green boxes) and 3V_switch rail (orange box). There are three voltage regulators on the board to provide the three voltage domains for the SRAM: 1.8V, 0.9V, and a variable core voltage (red boxes). The load switches provide current limiting which protects against SELs on the board. These load switches also prevent

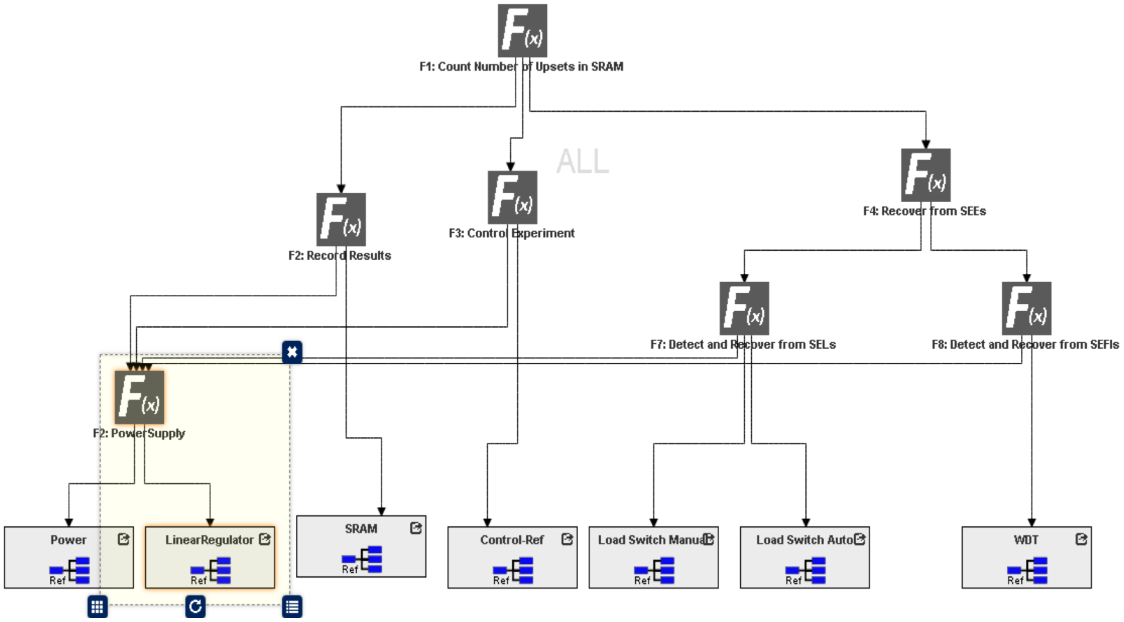


Figure 4. A functional decomposition model for "Count the number of upsets in the SRAM" [17].

high current conditions from propagating to the rest of the satellite. The load switches result in 5 different power domains on the experiment board which power all of the integrated circuits (ICs) on the board.

The watchdog timer (WDT) assists the microcontroller in recovering from SEFIs and SEU. The assumption being that misoperation of the controller will result in an unhandled exception, out of bounds memory access, or program jump. Failure to make progress in these scenarios will trigger a power reset by the hardware watchdog timer. Configuration parameters are stored in an external, non-volatile memory to both provide a greater level of protection against (single-event) data corruption and to continue experiment operation after unforeseen power loss.

4.0 MODEL-BASED ENGINEERING AND ASSURANCE

Fundamentally, model-based engineering is a top-down design flow as opposed to the bottom-up design flow that takes place on many first-time CubeSat designs. The idea is to define the system performance goals and requirements first then identify the high level functions and their decomposition into more concrete implementable functions, and finally build an architectural model for the system and sub-systems that provide the functions. - Setting up the problem (through the models) in a systematic way, helps identify the functional and safety issues that need to be addressed up-front while dealing with space missions, rather than as an add-ons to typicall terrestrial design flow.. This could help avoid potential re-design from unexpected problems and improve the chances of mission success.. Model-based engineering and assurance are in fact being adopted even for large space organizations like NASA [2,8]. In this section, we introduce some of the main aspects and advantages of model-based design, in the next we present a paradigm and platform that addresses the specific issues of small-sat mission development.

A model can be defined as "A physical, mathematical, or otherwise logical representation of a system, entity, phenomenon, or process," [9]. Models can be constructed for any physical domain, such as electrical, optical, or mechanical, and they can be classified as computational or descriptive [10]. A computational model is an equation-based model suitable for simulation in computer application, which can be deterministic or statistical. A common example would be the equations for circuit elements in a Spice simulator. A descriptive model is a human-interpretable model that captures elements of system according to particular format, such as a block-diagram or flow chart. The System Modeling Language, or SysML, is a widely-accepted standard for a descriptive model for systems. It is a graphical modeling language that supports specification, analysis, design, verification, and validation of systems, based on

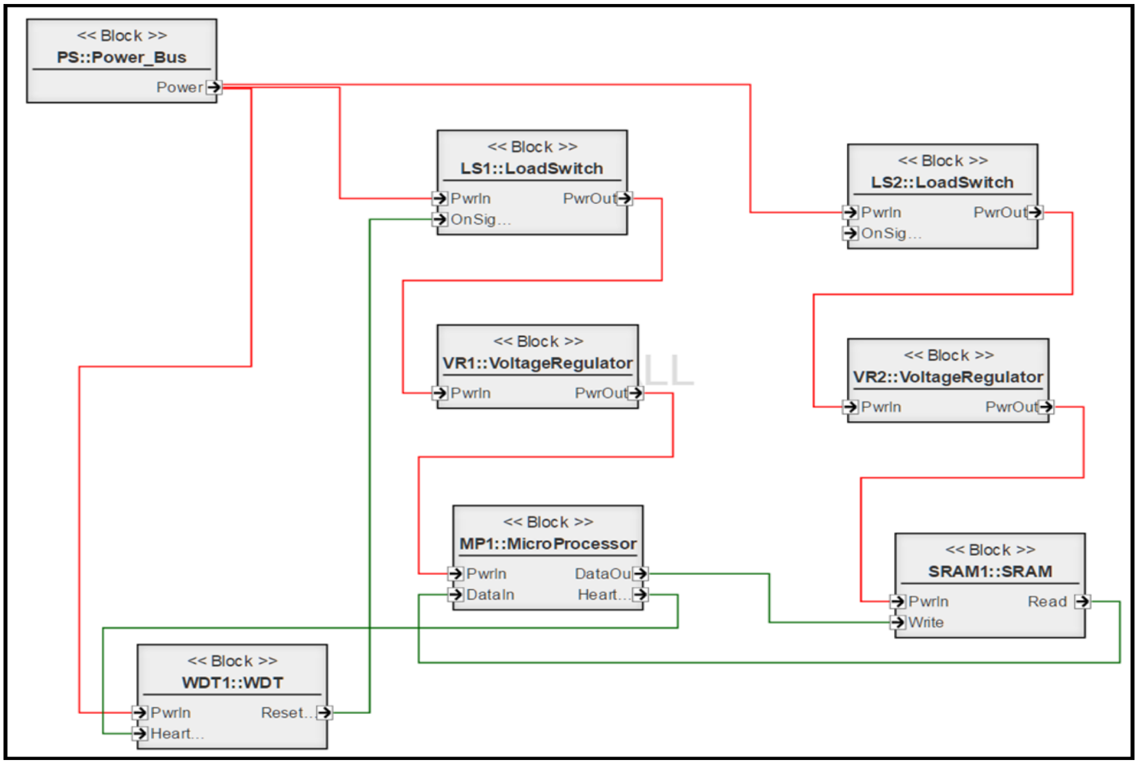


Figure 5. SysML Block diagram of REM board SRAM Experiment.

Red lines are power flows, green lines are signal flows.

the four "pillars" of SysML: structure, behavior, requirements, and parametrics [11]. SysML defines different kinds of diagrams that represent the system including: behavior (functional), structural (block-diagram), requirements diagrams, and lower-level diagrams such as state machine or use-case diagrams. Creating a SysML description of the system is the first step in the proposed approach. A functional block diagram showin the relationships between individual functiona and the components that execute them is shown in Fig. 4. A block-diagram representation of the SRAM experiment board is shown in Fig. 5, showing signal and power flows.

5.0 GOAL STRUCTURING NOTATION

Goal Structuring Notation (GSN) is a graphical notation standard used to explicitly document an assurance case and defined in the standard in [13]. An assurance case is a reasoned and compelling argument, supported by evidence, that a system will operate as intended for a given, defined environment. This argument is made through a series of connected claims that support an overall claim. Acceptance of the assurance case requires that the argument by reviewed. GSN provides a way of documenting the assurance case that allows others to discuss, challenge, and review the assurance case.

The GSN model provides structure to show how the overall claim is supported by sub-claims. These claim in GSN are represented as goals and blue boxes in Fig. 6. The assertion of evidence to support the truth of a goal is represented by a solution and the orange boxes in Fig. 6. When documenting the reasoning between the goals and sub-goals, strategy elements, the green boxes in Fig. 6, are used. These boxes are connected and show the flow of the argument from top to bottom by solid arrows, "Supported By" in Fig. 6, that show evidential or inferential relationships. These three types of boxes make up the main argument structure.

To show how the system and it's environment affect the argument, three different types of context boxes are used. The first are the generic context boxes, the yellow boxes in Fig. 6, which provide general information about how a goal or strategy should be interpreted. The second are

Figure 6. Explanation of Goal Structuring notation [2]

the grey boxes in Fig. 6, assumption boxes. Assumptions are premises that need to be true in order for the goal or strategy to be valid. The last type are justification boxes, teal boxes in Fig. 6. These boxes explain why a goal or strategy is acceptable. These three types of context boxes are connected the main argument structure by dotted "In Context Of" arrows. All of these boxes and arrows are show in Fig. 6 and a more detailed explanation of the example argument can be found in [7].

6.0 BAYESIAN NETS

A Bayesian Net (BN) is a way of computing joint probalities by calculating conditional probabilities, for example, the joint probability of X and Y and Z, often written $P(X,Y,Z)$, can be written as

$$P(X,Y,Z)= P(X|Y,Z)\ P(Y|Z)P(Z) \qquad (1)$$

[14]. An example is shown in Fig. 7, which shows a BN for computing the probability that the data read from an SRAM is good given assumptions about the radiation environment. Each box in the graph consists of a conditional probability table, a matrix of probabilities that describe the outcomes given the value of the condictional variable represented by the incoming arrows to the box. It is possible to generate Bayesian nets semi-automatically from the fault propagation diagrams described in Section 7.1. The utility of a Bayesian net is that it is a way to quantitatively evaluate the probability of faults in a given system.

7.0 SEAM: SYSTEMS ENGINEERING ANALYSIS AND MODELING (SEAM)

SEAM is modeling language, supported by a web-based, collaborative modeling platform called WebGME, for supporting a mission assurance process. SEAM borrows a number modeling concepts and techniques from the standard SysML but it extends those with new features and adds completely new ones. It is compatible with some SysML features, including: requirement models and internal block diagram models, and those can be imported into the modeling tool.

Having studied and used SysML on real systems, we found that it is not sufficient for supporting an assurance process – it needs several extensions. First, it lacks support for functional decomposition, which is a core

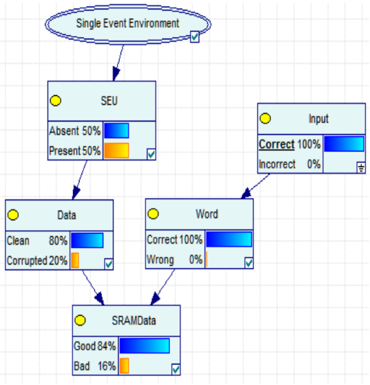


Fig.7 Example of a Bayesian Net showing the probability of good SRAM data.

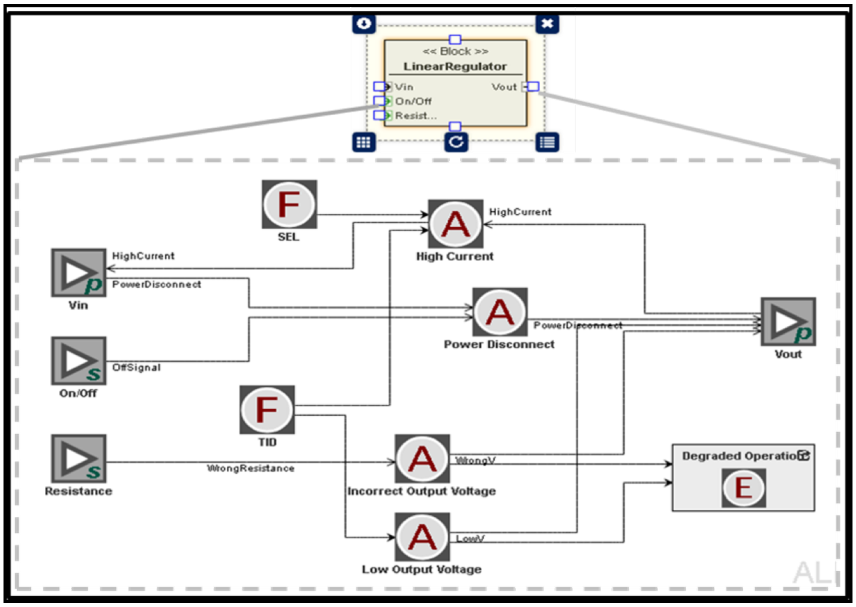


Fig. 8. Fault propagation diagram for linear regulator operation [12]

systems engineering activity. Missions need various functions from the spacecraft, hence the identification of functions and their mapping to various (hardware and software) components of the system is highly relevant. Second, it has no concepts for modeling component faults and degradations (to the environment effects like radiation), their effects and the propagation of those, and the impact of those on the various functions of the system. Third, it does not directly support the modeling of assurance cases: a formal or informal logical argument that captures why the system is 'mission safe', i.e. it does no harm that would put the mission at risk, in case of components failing.

While recognizing that SysML's extension mechanism based on stereotypes *can* be used to address these concerns, we made our solution through adding the new concepts as first class modeling elements to SEAM. Below, we briefly describe (1) the method used for fault modeling, and (2) how the various modeling aspects are integrated in the language.

7.1 The Fault Propagation Model

In order to understand the vulnerabilities of the system to be deployed in a CubeSat, it is important to understand the faults that could originate in each component and the local effect of these faults within the component and global effect across the system i.e. in other components. This information can be captured as a 'Fault Propagation Model' in the SEAM platform, which improves upon the SysML internal block diagram model.

Fig. 8 shows the internal block diagram model (fault propagation) of a 'Linear Regulator' block (component). The radiation-effects such as 'TID' and 'SEL' are captured as fault ('F' nodes) which lead to anomalies ('A' nodes) such as 'HighCurrent' and 'LowOutputVoltage' which further lead to functional degradation effects ('E' node) representing 'Degraded Operations' in the regulator. Additionally, the SEAM fault propagation model allows users to capture responses ('R' nodes) of a component to failures. This helps describe the mitigation functionality in the context of fault propagation.

Further, the propagation of failures into and out of the component are captured through labeled edges ('HighCurrent', 'WrongV', 'LowV' etc.) connected to the input/ output ports. The interconnections between the component ports (in the system design) helps complete the fault propagation model for the overall system.

7.2 The SEAM Platform: Combination of SysML and GSN and BN

For mission assurance one is to build a set of assurance cases that formalize why the system is 'safe' – i.e. why it does 'not do harm' to the mission, even if components exhibit various failure modes. We argue that such argument can be built properly only in the context of the entire system, including its requirements, functions, design, anticipated faults, and its environment. Note that this task necessitates not just the collection of these

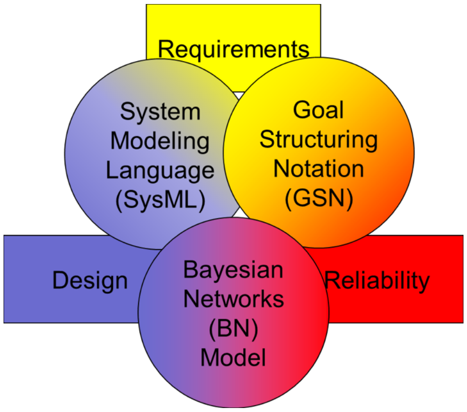


Figure 9. Relationship of the 3 paradigms in the SEAM platform. [2]

models, but also the linkage among them. This is goal of SEAM: to provide a language for integrated modeling.

In SEAM there various 'links' among the various types of models available in the language. Below we list a few of such links:

- Requirements are linked to system functions. The meaning of the link is that the function implements the requirement.
- Functions are linked to components. The link here means that the component/s is/are required to provide the functions. In other words, if the component fails, the corresponding function may be lost.
- Fault Propagation models are linked to appropriate functions through the functional degradation Effect ('E' nodes) and mitigation Response ('R' nodes). Here the links help understand how a function could be affected due to faults (anywhere in the system).
- The 'solution' (or 'evidence') nodes in the GSN assurance case model are linked to specific components. The link means that the evidence is related to the specific component.

Note that links are bi-directional and traceable. This allows various automated checks on the models for completeness, etc.

The integration of the SysML block diagrams (with failure propagation graphs) and GSN provides a powerful modeling paradigm, where a target system (e.g. a spacecraft) can be modeled together with the assurance that demonstrates its mission safety. The interlinked models allow structural and informal checks on the integrated models for completeness, but, in themselves are not sufficient for quantitative analysis.

Quantitative analysis of assurance cases considers the uncertainty (or 'degree of belief') in the argument and its evidence, hence it involves probabilistic reasoning. To follow this paradigm, we associate SEAM models with Bayesian Belief Networks (BNs). Currently this is done manually, but we are working on a (partially) automated solution.

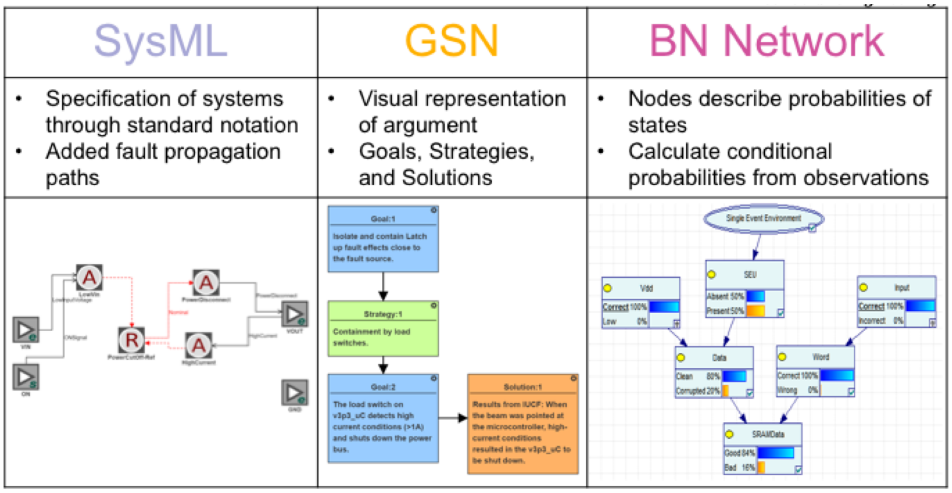
The reasoning for BN-s is as follows: The assurance expresses a logical inference chain structured as tree or, more generally, a Directed Acyclic Graph (DAG), with evidence at the leaves and the 'top goal' – the mission assurance statement at the top. The intermediate nodes combine evidence emanating from the evidence nodes and propagating upwards, towards the top goal. Given the calculation rules of the BNs and conditional probabilities at the intermediate nodes, one can propagate the uncertainty in the tree, resulting in a probabilistic assessment of the mission safety, in terms of the evidence available. Such inference is straightforward to perform and several tools are available. The key challenge he is how to obtain the intermediate nodes' conditional probabilities. However, it appears the these can be obtained from tests and/or operational data.

7.3 SEAM Website

SEAM platform is supported by a web-based, collaborative modeling framework called WebGME The SEAM platform can be accessed online at https://modelbasedassurance.org where users can browse through demo models as well as register for accounts to create their models.

SEAM platform allows users to create requirement models (SysML style), functional decomposition model, component library and system architecture model (SysML Block diagram), fault propagation model (extension to SysML Internal block diagram model) and assurance case model (Goal Structuring Notation). SEAM allows these models to be inter-linked so that the different modeling aspects can be contextually related to one another.

Fig 11. shows a screenshot of the SEAM modeling platform. The users can edit the model by dragging and dropping model elements from the "Model Parts Panel" to the "Model Editor Canvas". Models can be traversed via the "Tree Browser" or by double-clicking on a model element to move down the hierarchy. Model attributes can be edited through the "Attribute Panel". More tutorial slides, papers and demo models can be found on the website.



SysML	GSN	BN Network
Specification of systems through standard notation Added fault propagation paths	Visual representation of argument Goals, Strategies, and Solutions	Nodes describe probabilities of states Calculate conditional probabilities from observations

Fig. 10. Representation of the three platforms in SEAM [12]

8.0 SUMMARY

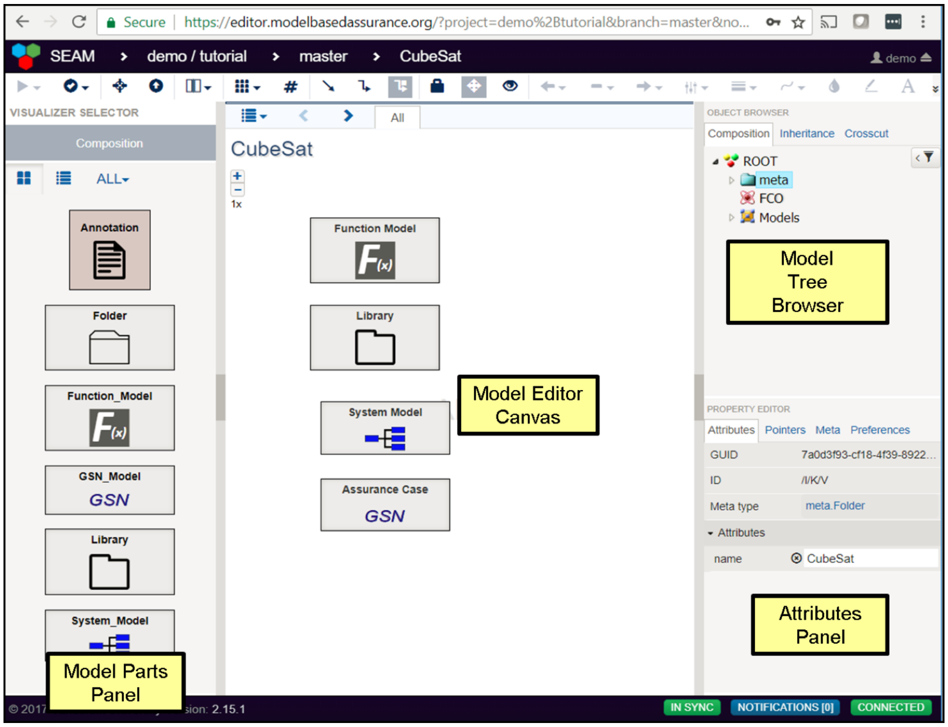
In this paper we describe a model-based assurance platform called SEAM intended to provide a systematic assurance case that a give satellite design will function as intended in the specified space environment. The approach is intended to provide systematic approach to assurance without requiring extensive ground-based radiation testing or the use of radiation hardened parts. The first element in the platform is a descriptive architectural model captured in SysML-like diagrams that articulate the functional decomposition of the design, the design structure, and the fault propagation flow between components. The second element in the SEAM platform is the use of Goal Structuring Notation which uses a graphical argument construct to build an assurance and safety case that the design operated properly given the characteristics of the particular space environment of the satellite. The third element in the assurance flow is the use of Bayesian nets to estimate the probability of observing degraded behavior or faults at particular points in the mission lifetime. The platform has been encoded in a browser-based application that is available to the public for free (though registration is required) at modelbasedassurance.com.

Acknowledgments

The authors wish to thank NASA OSMA, NEPP and JPL for their support of this work on NASA contracts NNX15AV48G, NNX16AR25G, 80NSSC18K0493, and JPL Subcontracts 1568812 and 1600101.

References

1. M. Swartout, "CubeSat Database," sites.google.com/a/slu.edu/swartwout/home/cubesat-database, ref. June 8, 2018.
2. B. Sierawski, R. Austin, A. Witulski, N. Mahadevan, G. Karsai, R. Schrimpf, R. Reed, "Model-Based Mission Assurance," 27th Annual Single Event Effects (SEE) Symposium, May 21-24, 2018, San Diego, CA.
3. D. M. Fleetwood, "Total Ionizing Dose Effects in MOS and Low-Dose-Rate-Sensitive Linear-Bipolar Devices," *IEEE Trans. Nucl. Sci.*, vol.60, no.3, pp.1706,1730, June 2013.
4. J. Schwank, "Total Dose Effects in MOS Devices," *Proc. NSREC Short Course*, 1993.
5. Z. J. Diggins, N. Mahadevan, D. Herbison, G. Karsai, E. J. Barth, R. A. Reed, R. D. Schrimpf, R. A. Weller, M. L. Alles, and A. F. Witulski, "Range-Finding Sensor Degradation in Gamma Radiation Environments," *IEEE Sens. J.*, vol. 15, no. 3, pp. 1864–1871, 2015.
6. P. E. Dodd and L. W. Massengill, "Basic mechanisms and modeling of single-event upset in digital microelectronics," in *IEEE Transactions on Nuclear Science*, vol. 50, no. 3, pp. 583-602, June 2003.
7. R. Austin, "A Radiation-Reliability Assurance Case Using Goal Structuring Notation for a CubeSat Experiment," M.S. Thesis, Vanderbilt University, 2016.
8. Evans, J. Cornford, S., Feather, M. (2016). "Model based mission assurance: NASA's assurance future," Reliability and Maintainability Symposium, p. 1-7. RAMS. 2016.
9. U.S. Undersecretary of Defense for Acquisition Technology, "DoD Modeling and Simulation



GUID 7a0d3f93-cf18-4f39-8922…
ID /I/K/V
Meta type meta.Folder
name CubeSat

Fig. 11. Screenshot of SEAM deployed on https://modelbasedassurance.org

(M&S) Glossary," DoD 5000.59-M 1998, Jan. 1998.

10. National Defense Industrial Association (NDIA), "Final Report of the Model Based Engineering (MBE) Subcommittee," Feb. 10, 2011.
11. Sanford Friedenthal, Alan Moore, Rick Steiner, "OMG SysML™ Tutorial," www.omgsysml.org/INCOSE-OMGSysML-Tutorial-Final-090901.pdf, INCOSE, 2009.
12. A. Witulski, R. Austin, G. Karsai, N. Mahadevan, B. Sierawski, R. Schrimpf, R. Reed, "Reliability Assurance of CubeSats using Bayesian Nets and Radiation-Induced Fault Propagation Models," NEPP Electronic Technology Workshop (ETW), 2017, nepp.nasa.gov/workshops/etw2017/talks.cfm.
13. GSN Community Standard Version 2, Assurance Case Working Group (ACWG), SCSC-141B, Jan. 2018.
14. Diggins, Z.J.; Mahadevan, N.; Pitt, E.B.; Herbison, D.; Karsai, G.; Sierawski, B.D.; Barth, E.J.; Reed, R.A.; Schrimpf, R.D.; Weller, R.A.; Alles, M.L.; Witulski, A.F., "System Health Awareness in Total-Ionizing Dose Environments," Nuclear Science, IEEE Transactions on, Year: 2015, Volume: 62, Issue: 4, DOI: 10.1109/TNS.2015.2440993, Pages: 1674 – 1681.
15. J. Pellish, "Radiation 101: Effects on Hardware and Robotic Systems," NASA Tutorial, accessed on swc.gsfc.nasa.gov, 06/12/18.
16. B. D. Sierawski, J. A. Pellish, R. A. Reed, R. D. Schrimpf, R. A. Weller, M. H. Mendenhall, J. D. Black, A. D. Tipton, M. A. Xapsos, R. C. Baumann, X. Deng, M. R. Friendlich, H. S. Kim, A. M. Phan, and C. M. Seidleck, "Impact of low-energy proton induced upsets on test methods and rate predictions," IEEE Trans. Nucl. Sci., vol.56, no.6, pp. 3085-3092, Dec. 2009.
17. J. W. Evans, F. Groen, L. Wang, R. Austin, **A. Witulski**, N. Mahadevan, S. L. Cornford, M. S. Feather and N. Lindsey, "Towards a Framework for Reliability and Safety Analysis of Complex Space Missions" Session 269-NDA-06, 2017 AIAA SciTech Conference, Grapevine, Texas, January 11, 2017.