Risk Management of Student-Run Small Satellite Programs

by

Elizabeth Deems

Submitted to the Department of Aeronautics and Astronautics in partial fulfillment of the requirements for the degree of

Master of Science in Aeronautics and Astronautics at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2007

Author……………………………………………………………………………..
Department of Aeronautics and Astronautics
May 25, 2007

Certified by……………………………………………………………………..
Colonel John Keesee
Senior Lecturer in Aeronautics and Astronautics
Thesis Supervisor

Certified by……………………………………………………………………..
David Miller
Professor in Aeronautics and Astronautics
Thesis Supervisor

Accepted by…………………………………………………………………….
Jaime Peraire
Professor of Aeronautics and Astronautics
Chair, Committee on Graduate Students

Risk Management of Student-Run Small Satellite Programs

by

Elizabeth Deems

Submitted to the Department of Aeronautics and Astronautics on
May 25, 2007, in partial fulfillment of the requirements for the
Degree of Master of Science in Aeronautics and Astronautics

ABSTRACT

This paper proposes an approach for failure mode identification in university-affiliated, small satellite programs. These small programs have a unique set of risks due to many factors, including a typically inexperienced workforce, limited corporate knowledge, and a high student turnover rate. Only those risks unique to small, student-run satellite programs are presented. Technical risks and mitigation strategies of student and industry satellites are also discussed. Additionally, several risk management strategies are explored, and the advantages and disadvantages of these risk-related tools and techniques are examined.

To aid the process of risk identification in these particular programs, a master logic diagram (MLD) for small satellites was created to help identify potential initiating events that could lead to failures during the mission. To validate the MLD, a case study and multiple experiments are presented and analyzed. This master logic diagram approach is shown to provide an effective method of risk identification that can be easily adapted to small, student-run satellite programs.

Thesis Supervisor: Colonel John Keesee
Title: Senior Lecturer in Aeronautics and Astronautics

Thesis Supervisor: David Miller
Title: Professor in Aeronautics and Astronautics

Acknowledgements

I would like to thank Colonel John Keesee, for his continued support as a mentor and advisor. He has helped me through this entire experience and cannot be thanked enough. Thanks to my family for their never-ending support and love. To the Mars Gravity team, thank you for the laughs, the support, the chocolate, and all the wonderful meetings. And specifically to Dan, you are the best Bus and System co-lead one could ask for, and a good friend on top of it all. Thanks to the Z-Center for keeping me sane. And last but not least, Dave Miller, the other Space Systems Lab professors, and my friends there have been a great source of support.

In addition, much insight into general risk management programs at universities was provided by the following people: Tom Bleier, Cameron Boyd, Giovanna Brizio, Matt Carney, Carl Claussen, Terry Deshler, Ryan Ellis, Dr. Stephen Horan, Derek Huerta, George Hunyadi, Dr. Chris Kitts, Brian Larsen, Gerald Lehmacher, Kyle Leveque, Mark Maier, Michael McCarthy, Neil Melville, Dr. Dave Miller, Jason Mintz, Matt Nelson, Dr. Mason Peck, Hank Pernicka, Dr. Jordi Puig-Suari, Dr. Helen Reed, Dr. Michael Swartout, Sir Martin Sweeting, Prof. Bob Twiggs, Wouter Jan Ubbels, Earl Wells, Colonel Pete Young, and Dr. Robert Zee.

This research and my tenure with Mars Gravity have been supported in part from the following: the Harriet G. Jenkins Predoctoral Fellowship 2005-2006, the Aero/Astro Department Fellowship 2005-2006, the National Science Foundation Fellowship 2006-2007, and the Mars Gravity Biosatellite project 2006-2007.

Contents

List of Figures

List of Tables

Nomenclature

Abbreviations

ACS	Attitude Control Subsystem
AFRL	Air Force Research Laboratory
C&DH	Command and Data Handling
Cal Poly	California Polytechnic State University, San Luis Obispo
COTS	Commercial off the Shelf
DoD	Department of Defense
EDL	Entry, Descent, and Landing
EOL	End of Life
EPS	Electrical Power Subsystem
ESD	Event Sequence Diagram
FBC	Faster Better Cheaper
FMEA	Failure Mode and Effects Analysis
GN&C	Guidance, Navigation, and Control
LV	Launch Vehicle
MGB	Mars Gravity Biosatellite
MIL-STD	Military Standard
MIT	Massachusetts Institute of Technology
MLD	Master Logic Diagram
NASA	National Aeronautics and Space Administration

NOAA	National Oceanic and Atmospheric Administration
NMSU	New Mexico State University
PRA	Probabilistic Risk Assessment
PSI	Payload Systems Incorporated
QUT	Queensland University of Technology
S/C	Spacecraft
SPHERES	Synchronized Position Hold Engage and Reorient Experimental Satellites
SSL	Space Systems Lab
SSTL	Surrey Space Technology, Ltd.
TT&C	Telemetry, Tracking, and Command
UMR	University of Missouri-Rolla
UNP	University Nanosatellite Program
UTIAS/SFL	University of Toronto Institute for Aerospace Studies' Space Flight Laboratory

Chapter 1: Introduction

1.1 Motivation

Universities are becoming more and more involved in small satellite projects. This relatively new research field has students excited about space exploration they can be involved in right away, and schools are finding that satellite projects are a good way to teach students hands-on satellite engineering.

However, professors across the world have recognized that student satellites have many risks, and two of the most commonly mentioned are:

1) Lack of experience in designing and testing satellites and in using risk management
2) An organization and culture inherent to university programs to take on risks and to not manage those risks in the long term, especially from one group of students to the next

This paper will investigate all risks related to student satellites and provide strategies that can be used to reduce these risks.

1.2 Why Use Risk Management?

A risk management process helps to promote mission success and safety in any engineering project. While keeping the program's objectives in mind, this process can help identify what might affect the outcome of the project in a negative manner. It is important to identify critical end states of the satellite early so that design changes can be made to prevent these problems, or resources can be allocated to them.

Without keeping track of and mitigating the effects of risks that threaten the mission, there is little hope of maintaining the performance, schedule, or budget; a management plan helps allocate all three of these things in the risk reduction process. A risk management plan can even decrease costs if started at the beginning of a design by devoting money to developing high risk items early on. This will prevent schedule delays and hurried design fixes, both of which are costly and risky.

A risk plan brings together all levels of engineers with the project management team. Making all personnel aware of failure modes throughout the program (and not just the dangers within their own work) can help to ensure the success of the mission as a whole. In addition, continually monitoring risks and updating the team members on the status of the risks can help to keep the project on track.

Aside from the benefits to the project, risk management should be used to help teach the students how satellite projects are run in industry. Student satellites are mainly tools for teaching about the process of engineering design, of which risk management is an important aspect. The students can learn to resolve both technical and programmatic risks through their involvement in the risk management process.

1.3 Thesis Objectives and Outline

The objective of this thesis is to research and analyze the programmatic and technical risks that student satellite projects face. Then, a method for risk management, and specifically failure mode identification, will be developed, analyzed, and applied to a satellite project.

Chapter 2 gives an overview of risk management and the five steps in the risk management process. The scope and scale of student-run, small satellite programs are defined, and programmatic risks unique to student satellite projects are discussed. These risks include funding, experience, staffing, direction, schedule, documentation, and recruitment. Then, a study was done to collect information on the success rate of student satellites that have launched, and the technical failures are discussed. In addition, the technical risks associated with industry satellites are analyzed and compared to the types of failures for small satellites.

Chapter 3 first covers current risk management methods at universities. About ten case studies of schools are presented to show the range of risk management plans that schools utilize. Taking these management plans into consideration, and looking at the risks unique to schools, suggestions are made for the improvement of programmatic risks in university risk management. In this section, it is noted that a framework for risk identification would help schools identify and reduce risks. Next, techniques for the mitigation of technical risks are presented. Finally, advantages and disadvantages for alternative platforms versus satellite platforms for space and near-space missions are discussed.

In Chapter 4, failure mode analysis options are presented and compared to each other. The master logic diagram (MLD) is chosen as the best option to mitigate the programmatic and technical risks related to student projects. The development of the MLD for small satellites is then discussed. Uses of the master logic diagram are presented along with limitations of the MLD. The benefits of the MLD, relating back to the programmatic risks facing small satellites, are given at the end of this chapter.

The application of the MLD to the Mars Gravity Biosatellite (MGB), a project at the Massachusetts Institute of Technology (MIT), is presented in Chapter 5. The background of the Mars Gravity project and its previous risk management attempts are presented to show why the MLD was applied to the project. Next, the MLD was compared with other common risk identification methods to see whether the MLD is in fact helpful in identifying risks. The MLD was also compared to the percentages of failure modes per subsystem for on-orbit satellites in order to show that the MLD could identify not only the type, but also the number, of failure modes. Then, the full risk management process, with the MLD fulfilling many of the steps, is discussed. Finally, the results and benefits of applying the master logic diagram to the Mars Gravity project are presented.

Chapter 6 gives a summary and details the contributions made in this thesis. Recommendations are also made for future work.

Chapter 2: Programmatic and Technical Risks for Student-Run and Industry Satellites

This chapter will discuss risk management and the steps usually included in the risk management process in industry. The definition for student-run, small satellites is given in greater detail, and both programmatic and technical risks facing student satellites will be discussed. For comparison purposes, technical failures of industry satellites will also be presented.

2.1 Risk Management Overview

A risk is “a factor, thing, element, or course involving uncertain danger; a hazard.”[1] identifying something as a risk means that the combination of the likelihood of the event and the severity of its consequences threatens the mission. Risk management is a broad term used to describe is a multi-step process to reduce both programmatic and technical risks. Programmatic risks are those that threaten the program as a whole, including high turnover of staff, a tight schedule, etc. Technical risks relate to components and subsystems. Failure modes, such as equipment failure, short circuits, etc., of the satellite are some examples of technical risks.

Risk management is the process of identifying issues that may be potential pitfalls to the success of a program and then creating and implementing a plan to mitigate those risks, assuring that the available resources are enough to facilitate mission success. A risk management plan first requires understanding and identifying risks. Then, it is necessary to analyze the probability, impact, severity, and urgency of the failure modes. Next, a mitigation strategy is developed to reduce risks, and decision points and testing strategies are identified to see whether failures have been eliminated. All risks, including ones that have been mitigated, are then monitored and updated throughout a project's lifecycle.[2]

2.1.1 Steps of Risk Management

The first step of a risk management process is to understand the types of risks that a program faces. Due to the variation in project size and type, each project requires a different level of sophistication in their risk management programs, and the team must decide what level of management is necessary for the success of their project. When trying to understand the risks, engineers must consider what types of problems the program will face as well as the framework of the whole project. By understanding such aspects as the mission objectives and scope, customer needs, acceptable risk levels, etc., each person will have a better view of what represents a risk for their project.

Once it is understood what constitutes a risk and the context in which they must be mitigated, risks to the project must be identified. The risk manager should specify what level of detail is necessary so that there is consistency within the program. There are many strategies to identify and assess risk, which will be discussed in Sections 3.1 and 4.1, but a consistent technique must be chosen for the program.

In traditional risk management programs, the risk team should assess the risks, analyzing the probability and severity of the risk occurring as well as the timeframe in which the risk must be handled. Calculating the probability is often a difficult task given the lack of failure rate data and the early phase of the design. However, given some preliminary data on failure rates of components or subsystems, this process can be included in the risk management plan from the beginning. This data can be useful for allocating resources to the most probable failures.

Teams should also analyze the severity, or impact on the project, of a risk occurring. The impact on the mission's goals and other parts of the project is an important factor in determining the priority for risk mitigation. If a risk impinges on another subsystem and the relationship is not adequately understood, serious problems could occur when the risk propagates through the program. Understanding the severity of the risks will help to plan the monetary resources and the schedule of the mitigation plan according to the failure modes that will affect mission goals the most. The timeframe of the risks should be noted and monitored to ensure that all risks are dealt with before its threat becomes more serious. Note that all of these tasks take time and money to implement, and they might not be necessary for all projects. However, these are the traditional steps taken in risk management.

After analyzing the risks, a mitigation approach is needed. Depending on the level of risk of each item, the team can accept the risk, monitor the risk without immediate intervention, transfer the risk to another team, or implement a plan to lessen the impact or probability of the risk. An action plan should include potential mitigation methods, the chosen approach, decision points, and tests to see whether the threat has been eliminated or reduced to an acceptable level. To verify that a risk has been mitigated may involve testing, analysis, demonstration, or observation. Having a good grasp of the context of the risks (as mentioned in the first step) also helps when deciding what mitigation and testing strategies to use.

The mitigation plan is not a static document; the risks must be tracked and updated on a predetermined schedule in regular reviews. Continuous monitoring of risk allows for better control over the resources that are being used in the mitigation process. Figure 1 shows the flow of risk management information.

Throughout the risk management process, communication is a key element of ensuring a successful program. All members of the team must be informed about the processes being used, and decisions must be thoroughly documented. Strong support from the leaders of the project will help in all stages of risk management by providing clear guidelines and opening communication. These leaders also need to direct the effort to formalize a risk management program, and the management should carry through by implementing the generated strategies.

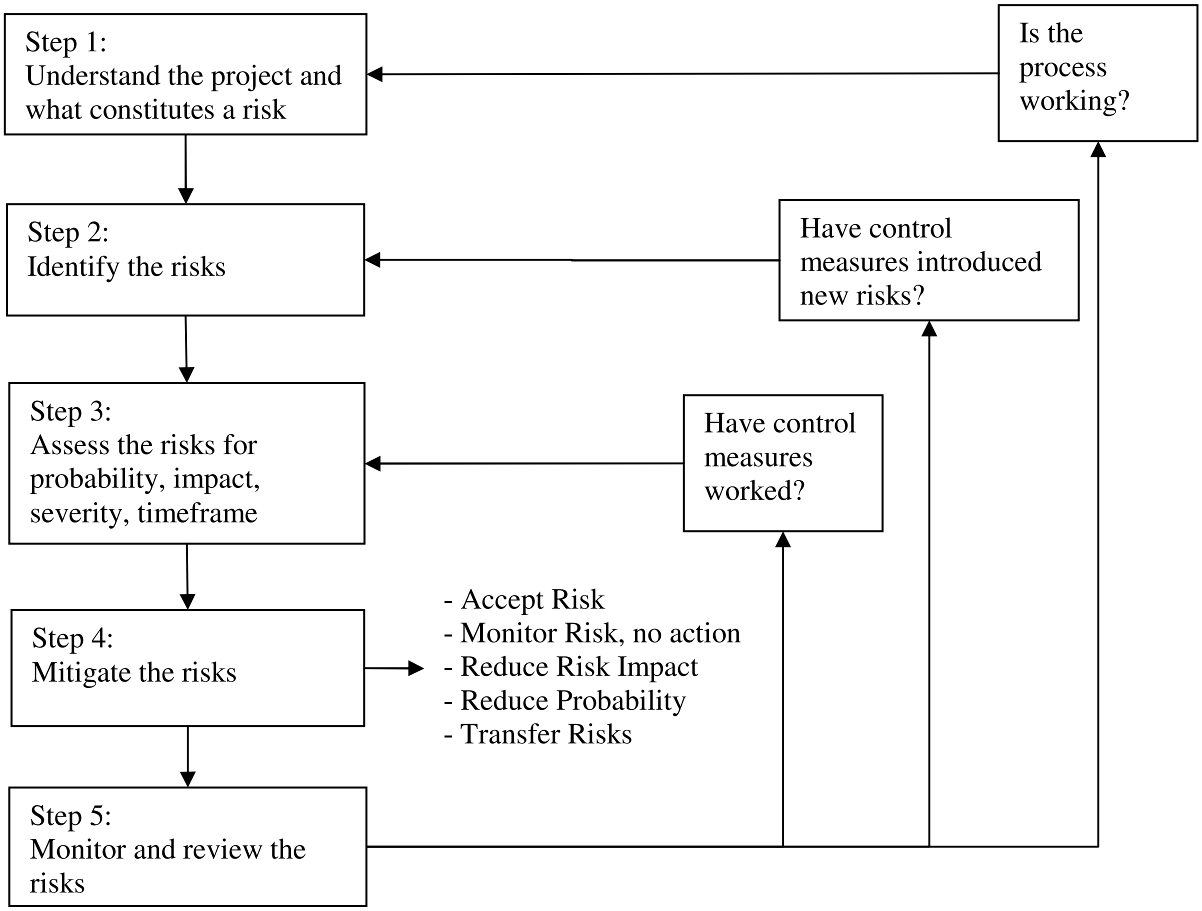

Step 1:
Understand the project and what constitutes a risk
Step 2:
Identify the risks
Step 3:
Assess the risks for probability, impact, severity, timeframe
Step 4:
Mitigate the risks
Step 5:
Monitor and review the risks
- Accept Risk
- Monitor Risk, no action
- Reduce Risk Impact
- Reduce Probability
- Transfer Risks
Have control measures worked?
Have control measures introduced new risks?
Is the process working?

Figure 1. Risk Management Process Information Flow

2.2 Definition of Student-run, Small Satellites

To understand the context of this paper, it is necessary to define what a student-run small satellite program is. Michael Swartout, from Washington University in St. Louis, gives an explanation of "university-class" satellites.[3] The definition of a university-class (or student-run) satellite is one that is functional as a self-contained satellite, with inexperienced personnel doing a majority of the work. In addition, the learning process and student training is an integral part of student-run satellite projects. Throughout this paper, only programs that meet these criteria will be discussed as student-run satellites.

The term "small" is used loosely in the satellite community and does not yet have a widely accepted definition. Surrey Satellite Technology, Ltd. (SSTL)[4], a for-profit small satellite supplier, claims that satellites less than 500 kilograms are small, but today's small satellites are also distinct from small satellites from the days of early space exploration because of their complexity. SSTL further breaks the satellites into classifications of minisatellite, nanosatellite, etc., but all of these have a mass less than 500 kilograms.

Michael Swartout claims that mass is the wrong discriminator for satellites less than 60 kilograms, and that schools should use volume because it is a better estimate of the true capabilities of the spacecraft.[3] While it is a good point that classifying satellites into discrete groups such as the minisatellites and nanosatellites can be misleading, these distinctions are not of concern in this paper. Here, it is more important to understand that the programs are student-run and are generally smaller than programs in industry. Most typical student satellites fall into the one kilogram CubeSat or 30 kilogram Nanosatellite programs, but satellites with masses up to a few hundred kilograms will be discussed.

In summary, risk management is affected by the management as well as the size of the program. Therefore, both of these factors are important considerations. All satellites that are student-run will have risk management programs different from those done in the industry, and university-specific risk management plans can utilize the fact that their missions are smaller. Further sections in this paper will address these topics.

2.3 Programmatic Risk Discussion

While all satellite programs have threatening risks, small, student-run satellite programs have a unique set of risks associated with them. Student and industry projects are fundamentally different, from the program to the component level, and it is important to understand their distinctions. Because of their programmatic differences, unique university satellite risks occur in the areas of funding and competition, experience, staff, direction, schedule, documentation, and recruitment. Professors and students across the world have mentioned these problem areas, and MIT suffers from the same issues in its student-run programs.

2.3.1 Programmatic Differences between Industry and Student Satellites

In general, the fundamental elements of commercial businesses or government programs are similar to those found at universities. The biggest differences are related to the fact that universities normally have “less” of all major resources, which leads to many risks for small satellite programs.

In student-run programs, the scope and budget for the spacecraft are usually much smaller than in industry or government spacecraft. As a result, the monetary consequences of failure are lower; therefore, risk is perceived differently. While the student engineers working on a project are motivated by a desire to see the program succeed, the loss of the satellite is not as large of a financial burden and does not have the implications it would have for a satellite produced in industry. However, losing a satellite could be detrimental to the long-term success of a school’s program, and risk should be taken seriously. Risk poses different threats to university-based programs than to industry projects, and, therefore, university programs have unique requirements, varying acceptable levels of risk, and different mitigation strategies.

In university-based satellite programs, the total number of people working on the program is less, so in turn, the subsystem teams consist of fewer people. Smaller teams lead to shorter lines of communication, meaning risk mitigation could be implemented more quickly. However, many universities tend to have a lax risk policy, which means that risk management is conducted in an ad hoc and informal basis.

Since university teams are made up of mostly students, one major difference between university and industry programs is that a student’s primary focus is academic. The students must split their time between class, homework, extracurricular activities, research, and possibly work. With all of these activities (of which classes are usually the most important), research programs often do not receive the attention they need from students. In addition, with a limited number of working hours that students devote to satellite projects, their time is usually spent on designing and building the hardware and software, and not on risk management.

2.3.2 Funding Risks

Student projects are run with less money, which limits design options, time, and available resources, including staff (both professors and students), components, etc. Competing for funding as a university project can also be difficult because there are still limited resources for small satellites since the perceived value of these projects is low.

Obtaining financial support in the first place can be difficult for universities. This could mean that at the beginning of a project, there is little to no money to pay either students or professors. For staffing purposes, funding is especially critical to attract graduate students, who often do the bulk of the work. The programmatic risks discussed throughout this paper will become more prevalent without proper funding for an adequate staff.

Funding also effects design decisions, testing procedures, schedule, and nearly every other aspect of designing and building a satellite. Of course money is necessary for building and launching any satellite, but the budget of the program also drives the project and its schedule. In addition, without adequate funding, student programs will have to rely on cheaper parts and methods, most likely resulting in very high risk.

In many cases, student-run satellites compete for funding against similar university programs. These competitions are usually solely focused on small satellites at universities, and they provide funding for a limited number of schools to continue their work. To win this competition for funding, the schools must prove that they have an acceptable amount of risk for the type of mission planned. One method used by the CubeSat Program at Cornell University[5] calculates risk based on Technology Readiness Levels, which they aim to make better by using hardware with flight-heritage and standard algorithms and processes. Depending on the goal of the mission, this technique may or may not be suitable. Risks can be inherent to the mission's goals, but by showing that the team is aware of the risks and is working to minimize them as much as possible, these risks do not necessarily impede the project's chance of funding.

Some student-run satellites must compete against non-universities for funding. In this type of competition, the schools must convince the funding source that sufficient risk mitigation strategies are in place to give better-than-expected results. The combination of this acceptable level of risk and lower cost could comprise a good arrangement for the funding agency. Low-cost student satellites can be seen as a good investment if they have an acceptable level of risk.

Competition for funding against any group is difficult for schools both with and without prior satellite design work, but it is even more difficult for the latter. Creating and implementing a satellite program at a university takes great effort and expense, which increases the risks associated with their program.

The lack of funding affects other aspects of the mission, including the schedule. The physically small size of university satellites, as well as the prohibitive cost of being the primary satellite in a launch, leads many university satellites to be secondary payloads on a launch vehicle. Opportunities to be a secondary payload may not be identified until relatively close to the launch, making it difficult for the secondary payload team to obtain an appropriate launch opportunity. This ties the development of the student project to that of the available primary object launches, making the spacecraft development fit a (potentially) tight schedule and, therefore, increasing technical risk. On the other hand, the satellite may have to wait until a ride is available, which increases the risk of components malfunctioning as well.

2.3.3 Risks Related to Experience

With little to no formal training or guidance, students lack the experience to identify risk and suggest mitigation strategies. Since students, especially undergraduates, work the most with the subsystems, they are best positioned to make observations about risk, but they usually do not have the experience to perform tasks related to risk. A method such as a risk template (see Section 4.2) would be useful as both a teaching aid and a design tool for these students.

The lack of experience is compounded by the short period that students usually participate in a satellite project. When students join a project, both their general subject knowledge and their familiarity with the project are usually minimal. The learning curve to obtain general knowledge and become familiar with a project uses a large portion of the time a student has to work on the program. Students can join projects later in their undergraduate career to minimize this learning curve, but it can take a couple of semesters to fully catch up with the rest of the team, depending on the progress of the project.

Students have the advantage of being highly motivated and energetic, driven by the enthusiasm of being part of an aerospace project. The desire to learn and master the material helps mitigate some of the negative effects that lack of experience brings to the project. However, this optimism, coupled with inexperience, can lead to a lack of focus.[6] It is necessary to balance the students' experience and energy with set goals and deadlines.

2.3.4 Staffing Risks

There are five main components to the staff that work on university satellites – undergraduate students, graduate students, university staff (such as technicians), professors, and industry professionals. These groups differ in responsibilities and size, but most programs are set up so that undergraduates are the largest group, and they work on specific tasks within subsystems. Graduate students act as subsystem leaders and managers, bringing the different teams together. Technicians can be employed to fabricate equipment, run tests and experiments, and, in general, help the students throughout the design and development phase. The professor(s) and industry counterparts oversee the project from a systems- and program-level perspective.

One of the largest problems for these programs is the fact that a student must focus on classes, which can make it difficult to devote enough time to the satellite project. The students must split their time between class, homework, extracurricular activities, research, and possibly work. With all of these activities, research programs often do not receive the continued and focused attention they need from students. Without a set of dedicated students or funding available to pay personnel, it can be hard to guarantee that the project will have people with all the required skills, which also increases risk. It is difficult to mitigate this risk, so it may be necessary to deal with this issue while trying to reduce risks in other areas.

In addition, much of the work being done on the development of a small satellite is focused on design and fabrication, which would not be suitable for doctoral research. Having doctoral students on the project helps because they can be on the project longer than most students, which helps to maintain consistency and oversight. However, because of the type of work performed on many small satellites, it may be difficult to find PhD students to work on small satellite projects.

Turnover and losing students after graduation makes it difficult to keep stability in a project. By the time a student has enough knowledge to be fully productive in a satellite program, they are approaching graduation. This is also a problem since students, especially undergraduates, are inexperienced and bring few previously-acquired skills to the project.

Many tasks have only one person assigned to them. These single-string workers pose a serious problem to any project because if that one person becomes too busy or leaves the project suddenly, the job they were assigned may be delayed while a new worker is found and trained. Having single-string workers also requires a large learning curve/hand-off time when the next person joins, delaying the schedule if this was not factored in. Documentation when students leave is also an issue, and that will be discussed in Section 2.3.7.

Oftentimes, a core group of student managers keep the entire project team together. These students usually have more education and design experience and have been with the program for a number of terms. Therefore, this small group of people has many responsibilities across the breadth of the project, and these people need to be careful of burn out. However, the management should aim to stay with the project over a number of years to provide continuity and corporate memory.

Projects that occur in a class instead of as an extracurricular activity can have even more challenges than the ones listed above, but they also have other benefits. These classes have from one to, at most, three semesters to work on the project before it must be completed or handed over. These timeframes lead to either short development and production time or a project that will most likely be given to an entirely new workforce, incurring not only documentation risks, but also learning curves for a whole team. These classes do have the advantage that the students' grades are tied to their work on the project; ensuring most of the students are devoted to its mission for a specified number of hours per week.

In addition to the problems a project faces using student workers, there are often few paid, professional employees working on the project. The expense of employing technicians or professors may prove too high for small projects, leading to issues with oversight, technical help, direction, etc. A dedicated staff, whether it be undergraduates, graduates, or staff, is key to maintaining a student project's direction and corporate knowledge. Some turnover is acceptable and necessary, but there are more advantages to keeping a long term, committed group of people on the project.

2.3.5 Lack of Direction

As in industry, student-run projects oftentimes have many sources of requirements. In these fast-paced, understaffed projects, it is hard to devote the time and proper attention that requirements need. Without proper direction, students will get sidetracked from the critical design work, or they might not believe that the satellite really will fly. In this case, it is speculated that the students could make poor design decisions because they don't think that the satellite has a chance to get to orbit. On the other hand, with too many directions and goals that are too lofty, it has been noted at multiple universities that their projects will have a difficult time making design decisions and completing the mission successfully.

2.3.6 Schedule Risks

As mentioned previously, schedules are tied closely with money, personnel, and available resources. At a school, setting a schedule can be difficult because the number of hours that students have available to work varies each week. High turnover is also an issue when scheduling because the average student involvement is much shorter than the development time of the project, and the turnover rate is difficult to predict when setting schedules. In addition, the students and professors have little corporate knowledge of how long a certain job will take. Companies usually have experience with similar projects, making the projections of how many person-hours should be devoted to each assignment easier. Without this knowledge, it is harder for student projects to determine how long a task will require and how many jobs will be completed in a given amount of time.

Yet, staying on schedule is important for both the project's success and for the school's reputation. The program needs to meet strict deadlines throughout the design, and if the project deviates from these, it could delay the launch, or miss the launch all together. A university can also increase its credibility by staying on schedule, making it more likely for partners and funding agencies to invest in a school's projects in the future.

Many small satellite projects have a short development time – one to two years in length – and don't have a full time staff. While small satellites should take less time to build due to their lower complexity, the issues of timeframe and staffing makes their design and development rushed. For example, a satellite program at Utah State University had problems with their telemetry and command subsystem four times, but they never looked at the failure modes because there was neither the time nor the personnel.[6]

As mentioned in Section 2.3.2, small satellites are usually a secondary payload on a launch vehicle. Many launches have unscheduled delays, pushing the deadline back for the satellite delivery date, but these changes cannot be planned for. Teams, though, often make quick fixes to problems that could have been solved differently if they had more control over their schedule. While quick fixes are a problem in industry and for primary payloads, it is even more of an issue, and it's more common, for the secondary satellites because they have no control over the launch.

Since small satellites are often subject to fast-paced development and short schedules, not all risk management techniques are applicable to these projects. Both assistance in identifying risk and streamlined methods to analyze a satellite's risks are needed to better study these types of projects.

2.3.7 Documentation

In small projects, where the lines of communication are shorter, it is tempting to be lax on documentation requirements. However, because of the high turnover and need for thorough explanation to new students, keeping track of work is critical to the success of a student project. Improper documentation is a risk area because without proper records, there is a high likelihood of losing valuable research and information, including critical items such as rationale and assumptions. Since students often enter and leave a program in two years or less, a major risk is the handover of information to other team members. Furthermore, documentation is not exciting and students are not motivated to put the time in to be thorough. It is difficult, but necessary, to find time in a student's schedule to follow through with the required knowledge transfer.

After a mission is completed, whether it was successful or had failures, it would be best for the team and for the small satellite community for the school to document its lessons learned. Post-mission documentation may be even more difficult to enforce because the team splits apart, and students and staff move on to other projects. If there is no funding leftover, there is little motivation for people to continue work on documentation other than for the benefit of future students, and that may not be a strong enough reason.

2.3.8 Recruitment

With the high turnover rate of students, recruitment on an extracurricular project is a large part of the management team's role. A recruitment drive is needed once to twice per year, on average, and students must devote many hours to advertising, information sessions, interviewing, etc. Due to all these activities, a substantial amount of time is spent recruiting new members.

Bringing young students onto the project is time-consuming because they sometimes lack the education and experience required to work on satellite projects. It is important, though, to bring these students on board because they could be with the project for a long time, giving continuity to the program.

Signing up experienced students can also be difficult because the students that are old enough to have experience are usually committed to other activities. In most cases, programs will take on younger members and train them as a way of obtaining the necessary workforce, but this strategy involves a lot of effort and time from the current team members.

2.4 Technical Risk Discussion

In addition to programmatic risks, student satellites also face traditional technical risks. This section will investigate whether student satellites are prone to certain types of failures and how these programs may be able to learn from industry experience when dealing with technical risks. However, despite the benefit of sharing failures, many programs, especially in industry, do not usually make failure information available. Most satellite programs seem reluctant to share information about failures in their systems because discussing these failures can increase the perceived risk associated with a company or university. It also highlights management and/or technical faults in a company, even though all programs have these issues. Therefore, most programs do not like to discuss what went wrong with a satellite program.

This section looks at the data available that is most relevant to the study of failure modes of student satellites. First, technical student satellite failures will be discussed. Second, the results of studies on industry failures from both large and small programs will be presented. Lastly, the differences between the types of satellites will be compared. Where possible, failures will be identified at the subsystem level (propulsion, power, etc.) to show relevant trends in the types of failure in spacecraft.

2.4.1 Sources of Information

To find more information on satellite failures, there are a couple of sources available to the public. None are perfect for gathering data on types of failures, but they are useful for getting information on a specific satellite or a specific type of failure. Table 1 has information on some of the sources of satellite information that are available online for public use.

2.4.2 Student Satellite Technical Failures

This section will discuss the information available on technical failures of student satellites. Table 2 shows 95 student-designed satellites from all over the world that have been launched through April 2007. Due to the additional difficulty of designing flight hardware for space, the satellites in this list must have been designed to operate in the space environment and not solely in an orbiting space lab. The information for this list was gathered from the sources in Table 1 and other websites, as listed in the References ("Ref") column.

Table 1. Public Sources of Satellite Information

Webpage Title	Purpose	Searchable?
The Satellite Encyclopedia[7]	Description of about 2500 satellites, some information on failures	Yes
Satellite News Digest[8]	Timeline and list of satellite failures	Yes
Mission and Spacecraft Library[9]	Not recently updated and little to no failure information	Not working
Airclaims SpaceTrak[10]	Subscription site that has failure rate information but is mostly used by insurance companies	No
Encyclopedia Astronautica[11]	Information on satellite missions and some failures	Yes
Gunter's Space Page[12]	Information on national and international satellites, includes technical information and some mission outcomes	Yes
NASA Lessons Learned[13]	Official, reviewed lessons learned and recommendations from NASA projects	Yes
Michael's List of Cubesat Satellite Missions[14]	Partial list of student satellite missions and some outcomes	No
The Radio Amateur Satellite Corporation[15]	Summary of the status of amateur satellites, including some university satellites	Yes

To categorize the successes and failures of the satellites, the duration of contact with the satellites was noted, and if possible, the reason for the failure was also recorded. Many of the satellites did not make it to orbit because of a launch vehicle failure ("LV failure"), and others were unable to be contacted once they reached their orbits ("No contact"). Satellites that, because of a failure, could only be contacted intermittently throughout the mission or failed within the first few weeks of the mission, are categorized under "Some contact." Finally, satellites that were fully functional on orbit, at least for the majority of the mission goal length, were put under "Full contact."

Not including the satellites with launch vehicle failure, the following failure rates result: 57 (78%) of them were able to be fully contacted, nine (12%) satellites had some contact but premature failure, and seven (10%) had no contact. This data is shown in Figure 2.

Table 2. University-class Satellites and On-orbit Status

Satellite Name	LV failure	No contact	Some contact	Full contact	Reason	Ref
UoSat-1 (UO-9)				X		12
UoSat-2 (UO-11)				X		12
NUSat				X		11
WeberSAT (WO-18)				X		12
KITSAT-1 (KO-23)				X		11
KITSAT-2 (KO-25)				X		11
BremSat				X		7
Falcon Gold				X		7
Sputnik 40 (RS-17)				X		7
Sputnik 41 (RS-18)				X		4
PANSAT (PO-34)				X		12
SUNSAT (SO-35)				X		7
KITSAT-3				X		11
Tsinghua 1				X		4
Saudisat 1A				X		12
Saudisat 1B				X		12
Saudisat-1C (SO-50)				X		15
SaudiSat 2				X		16
UNISAT 1				X		12
UNISAT 2				X		16
Kolibri-2000				X		15
MOST				X		17
STSAT-1				X		16
UNISAT 3				X		16
TUBSAT-A				X		12
TUBSAT-N/N1				X		12
DLR-Tubsat				X		12
LAPAN-Tubsat				X		12
Maroc-Tubsat				X		12
Mozhayets 3 (RS-20)				X		12
Mozhayets 4 (RS-22)				X		15
Opal (OO-38)				X		7
Techsat 1B (GO-32)				X		7
Echo				X		15
HITSat				X		15
CP3				X		15
Libertad 1				X		15
Sapphire (NO-45)				X		18
PCSat2				X		18
MARScom				X		18
RAFT				X		18
ANDE				X		18
MidSTAR-1				X		18
QuakeSat				X		12
UWE-1				X		12
XI-IV (CO-57)				X		12
XI-V (CO-58)				X		12

CUTE-1 (CO-55)				X		12
CUTE 1.7 (CO-56)				X		12
GeneSat-1				X		12
Naxing-1 (NS-1)				X		19
FalconSat-3				X	Fixed Problem: Software	20
PCSat 1 (NO-44)				X	Degradation: Power	18
CP4				X		22
LIBERTAD-1				X		22
MAST				X		22
Cape1				X	Battery Failure, works in sun	22
SSETI Express (XO-53)			X		Power	12
JAWSAT (WO-39)			X		Communication, Power	3, 7
SEDSAT-1 (SO-33)			X		Communication, Power	7
TUBSAT-B			X		Radiation	3
UNAMSAT-B			X		Power, Thermal	3,12
ASUSat (AO-37)			X		Power	12
FalconSat-1			X		Power	12
Munin			X		Command & Data Handling	17
AAU Cubesat			X		Communication, Power	12
NCube2		X				14
DTUsat		X				14
CANX-1		X				14
Thelma		X				7
Louise		X				7
JAK		X				7
Mozhayets 5		X			LV-Satellite not separated	7
Techsat 1	X					7
FalconSat-2	X					12
UNAMSAT-A	X					12
UNISAT 4	X					12
YES	X					12
3CornerSat	X					21
ION	X					14
SACRED	X					14
KUTEsat	X					14
ICE Cube 1	X					14
ICE Cube 2	X					14
SEEDS	X					14
HAUSAT 1	X					14
NCube1	X					14
MEROPE	X					14
CP1	X					14
CP2	X					14
RINCON 1	X					14
Mea Huaka'i	X					14
Baumanets 1	X					12
PicPot	X					12
AlMASat-1	X					17
CP3					Not heard from yet	22

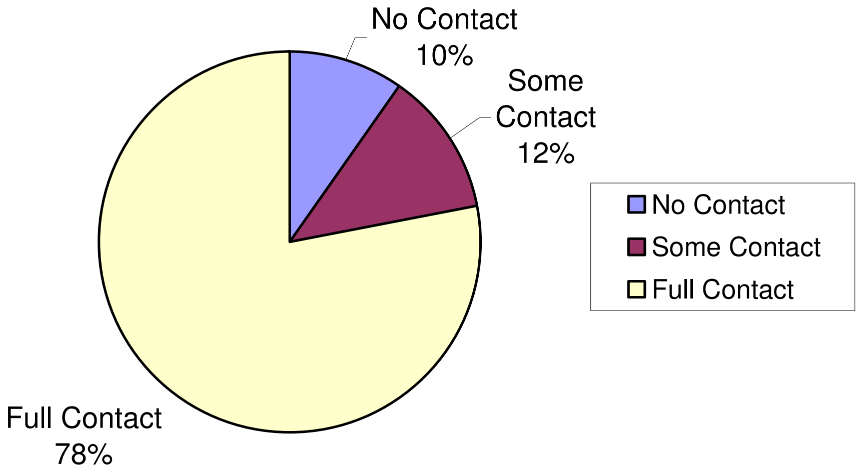


Figure 2. Failure Rates of Student-Run, Small Satellites, Excluding Launch Failures

For most of the satellites, no information was available on the reason for the failure, except for the "Some contact" group. In this case, eight out of the ten satellites reported failures due at least in part to the power subsystem. Two schools identified power and communication as the reasons for failure, one satellite's failure was Command and Data Handling (C&DH), and another had thermal problems, probably due to launch conditions in Russia. This information is important to study in order to understand what is causing small satellites to fail. It would be best to have information on all failures for the satellites, no matter what the final status of the satellite. However, this information is not readily available.

It is a significant trend that 80% of the satellites that had at least partial success failed due to power considerations. The fact that the power subsystem was identified as the failure in so many satellites could be because power failures may be easier to detect, or they may often be the real root cause. To better understand the failures these satellites are having and how programs can reduce the risk of power subsystem failures, the satellite with these failures will be discussed in more detail.

FalconSat-1[12] initially worked on orbit. However, cadets working in the ground station a few weeks into the mission found that the satellite's power system was not working properly and could not charge the batteries while in the sunlight. They could not solve the problem, and the mission was terminated early.

ASUSat[12] deployed properly, and the telemetry showed that the student-designed satellite components appeared to operate as designed. Then, half of a day into the mission, the satellite team received telemetry that there was a critical problem in the power subsystem that caused the solar panels to not charge the batteries. From the reports, it is unclear where the exact problem was other than in the power subsystem.

SSETI Express[12] also failed about half of a day into the mission. This satellite experienced problems with excess power dissipation due to a short-circuited transistor, which caused the batteries to not get sufficient charge.

JAWSAT had two sources claim two different failures for this mission. No information is available from the JAWSAT team, and their website is out of date and does not include any information about the on-orbit performance. One source says that the main battery failed after launch[7] and the other reports that it was a communication failure (either due to the transmitter or receiver)[3]. The failure may have been tied to both of these issues, but too little information is available to determine the root cause of the failure.

UNAMSAT-B had both thermal and power issues. Apparently, the satellite was not designed for the launch conditions in Russia, and the spacecraft's uplink oscillator was too cold before launch. Once the condition was noticed, the spacecraft could not be contacted in time to change the battery charging parameters for the cold conditions.[3,12] Due to this thermal and design problem, the power system failed.

PCSat lasted quite a long time in a semi-degraded state. The batteries became so weak that the satellite could not downlink anything during eclipse. Then, the satellite was found to work only during full sun or certain eclipses once every several months.[18] It was determined that the team did not plan for the worst case scenario of the need for instantaneous power during a re-boot from a reset condition. So the root cause may not have been power, but the design of the power system did not allow them to get out of this condition. Their lesson learned is that the start-up recovery mode must use low enough power so that the spacecraft can finish fully charging the batteries in the sun before an eclipse, so that the system does not reset while in eclipse due to low power. Teams should think about and discuss what tasks are necessary to design to the worst case scenario.[23]

The AAU CubeSat had an unidentified problem with the satellite transmitter, which resulted in only a small amount of data able to be downlinked due to a weak signal. This end state was most likely due to faulty antenna deployment, with two out of four antenna segments short circuiting. At about three months into the mission, the team was just beginning to have two-way communication again when the batteries failed because they could not store enough energy to continue operations.[24]

Prior to launch, SEDSAT-1[7,25] indicated potentially low solar panel performance, but there was nothing the program could do at that point due to time and funding. However, they had a long standing philosophy to have smart software that would manage the system at whatever performance level they got on orbit, and this was successful. The satellite developed power problems three days after launch because the battery capacity was not as high as expected. However, the major failure was the transponder on board because the ground stations could here the satellite, but they could not communicate with it to upload further commands and updates. Since this occurrence, control of the mission has not been regained.

These missions show the known types of failures experienced by small satellites so that schools can see the information available on technical risks for student satellites, which come from both hardware/software failure and inadequate design. While exact root causes were not determined for many of these missions, it is apparent that battery power is a main error source. Batteries may often fail because of their chemical nature and the ease with which they wear out. Failure to have adequate charge in the batteries could be due to the wiring, physical design, under-sizing, electrical power system, too much depth of discharge, etc., so universities should spend more resources on careful design and testing of this subsystem. The root cause of power failures might also be attributed to another subsystem, if for example the attitude control system could not keep the solar arrays in full sun, but there is not enough information to determine this. In any case, the power subsystem affects almost all other subsystems and, therefore, must be designed and tested carefully.

Mark Maier, a former professor at the University of Alabama-Huntsville, shared some thoughts on why power failures occur more in student satellites.[25] Most importantly, power systems are typically built by students with poor resources (as compared to industry satellites). The solar panels are constructed at the school in many cases, and they are not durable or very reliable. University projects cannot typically afford fully qualified space batteries, so they adapt another battery system, and reliability is likely to be low. Power electronics are harder to design that it might seem, and in addition, there are no good guides and few experts for power electronics. Also, power systems are fundamental to the spacecraft, so if it fails, the mission is over.

Another reason for common power system failures could be that this part of the satellites must handle large amounts of current, which stresses the electronic components and requires a lot of heat dissipation. Some piece of the power system is always turned on in, which also increases stress to this subsystem.

In contrast, communication systems, such as radios, are often procured and integrated as professionally built units. Structures are easy to build and test, and thermal demands are not very high on small satellites. Therefore, power subsystems are the student satellite weak point.

Swartout also investigated student (and some amateur) satellites and found 62 satellites that he identified as university-class satellites. Sixteen of these satellites (25.8%) failed prematurely (before the end of the nominal mission).[3] This information is different than that presented in Figure 2 because the types and number of satellites and the definition of failure between the studies varied.

Swartout also points out that certain subsystems are often not the primary cause of failures in a satellite. These include Commercial off the Shelf (COTS) hardware for such things as structures, thermal systems, batteries, and electronics.[3] Here, Swartout is saying that the actual hardware does not fail, while it is unclear in the university satellites discussed above what the actual root error source was – it might not have been the hardware, but it could have been the design of the battery or the rest of the power subsystem that failed. Therefore, Swartout's analysis can supplement the data from Table 2 when studying types of failures in the power subsystem. In addition, structural failure could be low due to required pre-launch vibration and static testing. The short mission duration of many of these satellites is another reason that COTS parts work well for student satellites – they don't need to be radiation hardened or have very long life expectancies.

Surrey Satellite Technology, Ltd[26] (SSTL) is somewhat of a bridge between the university and industry satellite program because they first started as a student-based group at the University of Surrey, but they currently make more advanced small satellites as a commercial enterprise. Their experience shows that mechanisms, power systems with high capacity, and propulsion systems often fail in general satellites. This anecdotal information only partially agrees with that from Swartout and Table 2. Swartout claims that batteries do not fail, while

SSTL claims that power systems do fail. These claims are not totally contradictory because there is more to a power system than just a battery, and Swartout mentions that the COTS hardware is reliable, while SSTL cautions the use of high capacity systems, many of which are not COTS products. Other than power failures, Table 2 does not have much information on failure modes, so it is hard to compare the other evidence from SSTL.

With such little information, though, it is still too early to draw conclusions on whether inferences on failure rate or subsystem failure tendency can be drawn from the data collected thus far. However, the studies agree that failures occur due to the complexity of spacecraft systems that are difficult to model or test on the ground.

2.4.3 Industry Failures

Small, student-run satellites might not be directly comparable to general industry satellites because of the inherent differences in their programs. However, it is interesting to see what the similarities and differences are between these two classes of satellites.

A few studies have been done to look at the failure rate of industry satellites. It can be very difficult to attain information from industry due to proprietary reasons and other information barriers. In addition to the same problems as industry, government programs also cannot share much information due to its many classified programs. In addition, all programs are somewhat tentative to share information on their failures for credibility reasons.

To try to remedy this situation, The Aerospace Corporation has begun collecting data on satellite failures. The Aerospace Corporation is a federally funded research and development center (FFRDC) that was created in the 1960 for the United States Air Force. It is a nonprofit corporation that provides both scientific and objective engineering support for the nation's launch, space, and related ground systems.

The Aerospace Corporation created the Space Systems Engineering Database (SSED) in order to organize and maintain many types of engineering data collected since the space program began. In 2002, SSED claimed to contain over 12,000 space system anomalies, which Aerospace uses to analyze space vehicle failures for its customers.[27] The SSED is still not complete because of the space community's unwillingness to share this information. In addition, the SSED and a few other databases at NASA and in industry are not for distribution to the general public. The next sections, though, discuss papers on industry satellite failure rates, and a few of these papers were based on information from the SSED.

2.4.3.1 Large Industry Programs

One area that has been a focus of study is general satellite failures from the commercial side and the government. The studies below discuss the failure rates and failure types for a number of different classes of satellites. Section 2.4.4 will compare the results of industry project failure rates and student satellite failure rates.

In 2002, a study was done on 1,080 commercial communication geosynchronous satellites that had been launched prior to 2001.[28] Data was used from the Public Record Satellite Anomaly Data Base, which is only for geosynchronous satellites and does not seem to be widely available any more. This study looked at the severity of the failure as well as the subsystem in which the failure occurred.

When considering failures where the spacecraft was permanently disabled, we can gather information on the likelihood for subsystems to fail for this particular class of satellites. The study found that most failures are due to high propulsion maneuvers (57% of catastrophic failures), such as launch, perigee kick motors, and apogee kick motors, of which the latter two don't pertain to university missions.

The next highest failure rate (15% of catastrophic failures) was in the Attitude Control Subsystem. The failures in this category were caused mostly by electronics and software issues as well as sensors and momentum wheels. Two other high failure areas include the Propulsion Subsystem (7% of catastrophic failures, usually from a leak or propellant depletion) and the Power Subsystem (6% of catastrophic failures, often due to electrical shorts).

A few subsystems had very few failures of any severity. These include the Structural subsystem (excluding mechanisms) and the Thermal subsystem. In addition, few errors were contributed to Operators, but this number might be low due to human influence and the proclivity to not report human error. This information is shown in Figure 3.

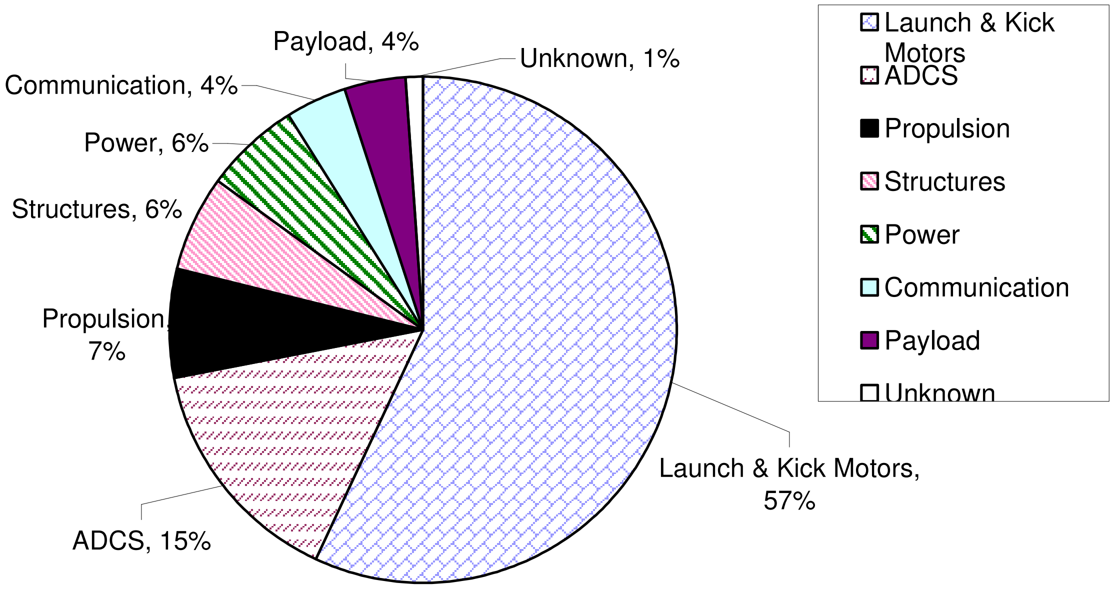


Figure 3. Geosynchronous Satellite Subsystem Failure Rate prior to 2001

Another 2002 paper looked at failure rates by subsystem primarily for 260 US Government and civil satellites from 1980-2002.[29] The space vehicles used in this study are Earth-orbiting spacecraft that were contacted after launch vehicle separation and survived at least the first day. In addition, they had to have a mass of greater than 100 kilograms with a design life of longer than one year to be included in the study. Failures were categorized by what caused the End of Life (EOL) – in other words, what caused the satellite to be inoperable (whether it resulted in the payload not working or the satellite not being able to support the mission requirements any longer). In general, more than 25% of space vehicles in this study that got to their intended orbit and functioned on the first day failed before they reached the end of their design lives. The data from this study is shown in Figures 4 and 5.

This study shows that the payload causes around 35% of mission-ending failures for Civil spacecraft and 26% for Military ones. Guidance, Navigation, and Control causes 28% of Civil satellite EOL failure and 24% of Military spacecraft failure, with reaction wheels and gyros being a root cause for a large portion of the failures. The Power subsystem contributes to 26% of

Civil failures and around 18% of Military ones. The Propulsion subsystem ends just 2% of Civil missions but nearly 20% of Military satellites.

In the GN&C area, a few specific items emerge as major contributors to EOL. In both Civil and Military missions, reaction wheels and gyroscopes were main contributors. A second large source of failure, for Military satellites in particular, is running out of fuel to perform guidance and attitude control. If this occurs before the end of the design life, it is most likely due to an under-design of the system or an overuse of fuel by the operators; however, it could also be due to a hardware or software malfunction that caused the expenditure of too much fuel.

For this category of satellites, the following subsystems rarely contributed to EOL failures: Structures and Mechanisms (no mission ending failures), Thermal Control (<1% Civil, none for Military), Commands & Data Handling (4% Civil, 2% Military), and Telemetry, Tracking, and Control (4% Civil, 10% Military).

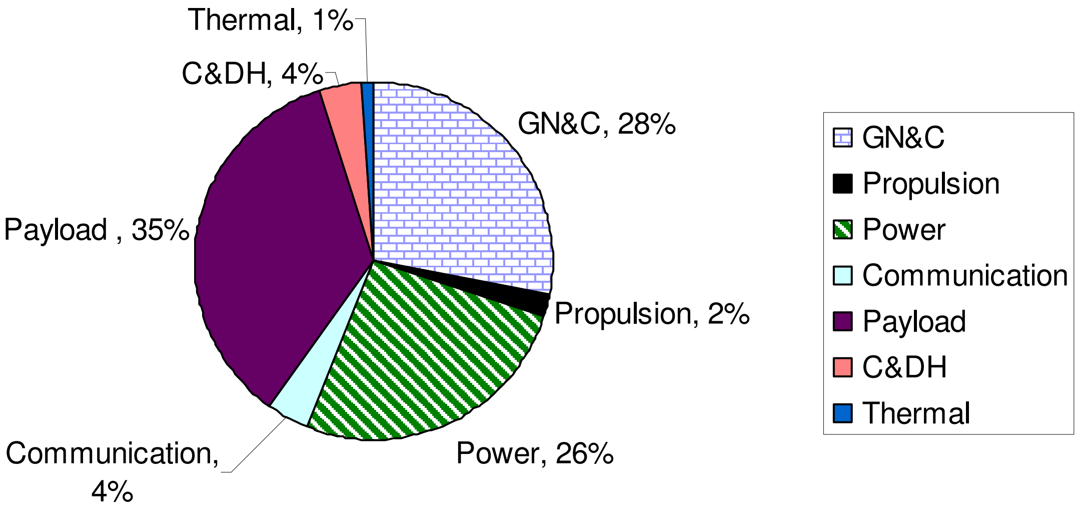


Figure 4. Civil Satellites Subsystem Failure Rates from 1980-2002

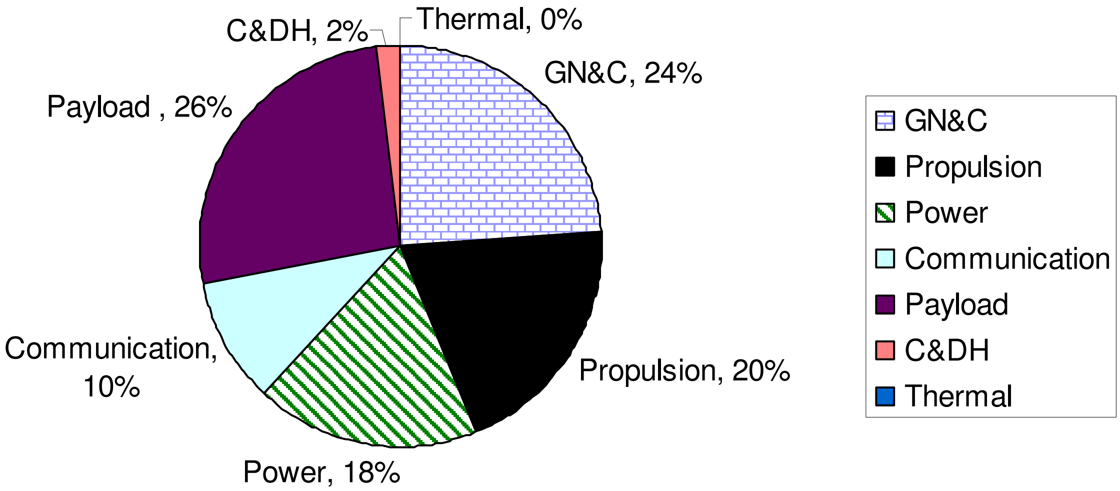


Figure 5. US Government Satellites Subsystem Failure Rates from 1980-2002

Reliability engineering has shown that throughout all types of industry, components tend to fail in a "bathtub curve" manner. This is characterized by early "infant mortality" that starts high and then decreases, a longer period of relatively constant failure rates, and then an increasing failure rate, often due to wear out. This study agrees with the "bathtub curve" result, with high failure rates during the first 90 days to one year, low failure rates from year one to six, and increasing end of life failures between years seven and ten. There were fewer satellites with design lives longer than ten years, so the failure rate decreases after year ten.

In 2003, further investigation into Guidance, Navigation, and Control (GN&C) failures in 764 satellites launched from 1990-2001 was done.[30] Satellite anomalies investigated include those from the US, Europe, Canada, and Japan that failed prior to the end of the design life. Here, a mission critical failure is defined as "the premature loss of a satellite or the loss of its ability to perform its primary mission during its design life."[30] GN&C is defined quite broadly and includes on-orbit subsystems that are directly involved in GN&C, such as the Attitude Control System (ACS), the Propulsion System, and software relating to the satellite's flight dynamics. It also includes all ground operations relating to GN&C, including trajectory planning, navigation, etc.

Here, 35 (29%) of all the anomalies in this satellite category were due to GN&C components, and 13 anomalies (37%) were recorded that resulted in total loss of the satellite. This report will focus on total loss of the satellite because it is unclear whether the other anomalies listed were mission critical. Separately, the ACS had approximately 23% of the mission ending anomalies, Payload had 20%, C&DH had 17%, and the Electrical Power Subsystem had about 14%. Data for these subsystem failures is shown in Figure 6.

Hardware contributes to the majority of the failures, while design, software, operations, and verification also have some influence on failures. The space environment causes the least number of failures when classified in this manner, and there are also some unknown failures that can't be placed into categories. When looking at hardware, seven anomalies occurred with reaction wheels, three each for pyrovalves, thrusters, and processors, two each for gyroscopes and GPS receivers, and one for tanks, Earth sensors, and nutation dampers. While these numbers are not very large, making it hard to deduce definite patterns, we can see that reaction wheels are again a large cause of failure.

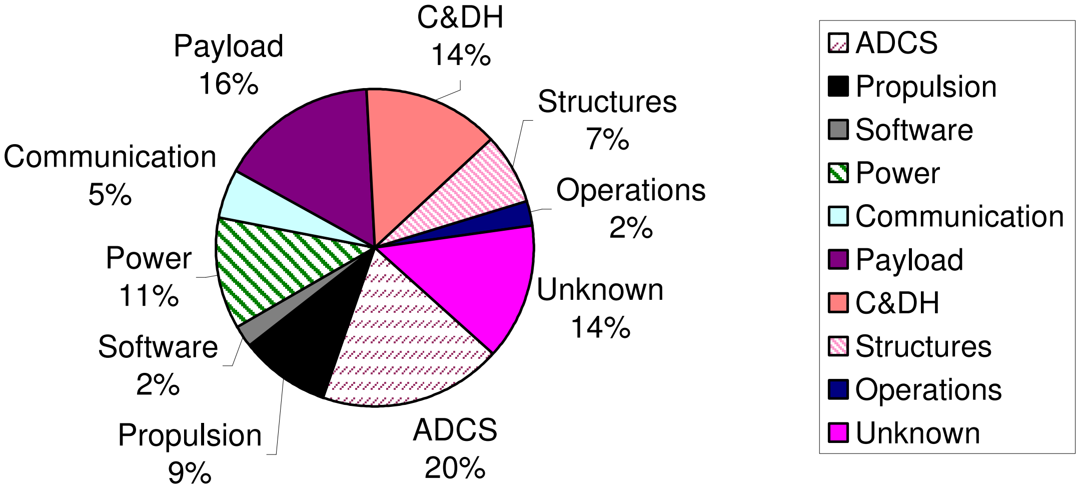


Figure 6. Satellite Subsystem Failures from 1990-2001, with a Focus on GN&C Failures

Pyrovalves seem to cause problems because of the mechanical or electrical shock generated by the pyrovalve. In each of these cases mentioned in the paper, the pyrovalve anomaly caused a total loss of the mission. Also to note, gyroscopes are lower on the list as compared to the 2002 Military/Civil study. One reason might be that gyros have been used less in recent years for attitude determination, so any gyro failures tend to result in partial performance loss or interruption in the mission rather than total failure.

This GN&C study shows that this set of satellites also contributes to the “bathtub curve” analogy; 51% of all anomalies (and 50% of GNC anomalies) occurred within the first 10% of the mission design life.

In summary, studies were performed on the reasons for mission ending errors in satellite design. These reports only cover the technical aspect and show that subsystems associated with Guidance, Navigation, and Control (including ACS) tend to end satellite missions before their design lives are over. How these failures relate to smaller programs as well as risk management will be discussed in Section 2.4.4.

2.4.3.2 Small Industry Programs

With the introduction of “Faster, Better, Cheaper” (FBC) in the aerospace industry, particularly at NASA, there has been a recent flux of satellites in industry on the smaller side. In 1999, one paper looked at the risk of FBC (small) versus traditional missions.[31] It was found that FBC missions do have a higher failure rate. The sample set for this study was small – just ten traditional NASA missions and 18 FBC missions.

The traditional missions investigated have a 10% catastrophic failure rate and a 30% total failure rate, which includes partial failures. FBC missions, on the other hand, have a 28% catastrophic failure rate and a 44% total failure rate. From this sample set of satellites, it appears that missions with a quick development schedule and a relatively small total budget are more risky than traditional missions. However, the FBC are cheaper, and the total investment lost is shown to be less with FBC failures than with the traditional NASA mission failures.

This study only looks briefly into the causes of failures. Figure 7 shows that hardware is the mission ending failure 41% of the time in these missions, software 27%, Launch Vehicle (LV) 9%, Program Management 9%, and Unknown 14%.

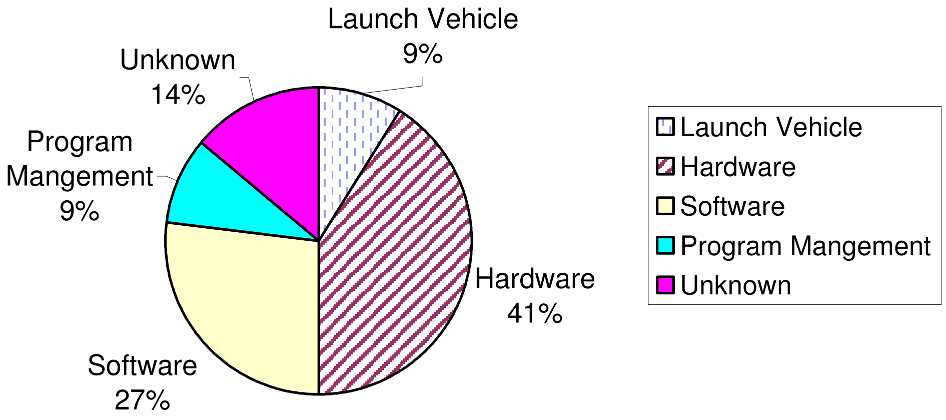


Figure 7. "Faster, Better, Cheaper" Satellite Subsystem Failures

2.4.4 Comparison of Failures

While the numbers of small industry and university satellites in these studies are small, it is helpful to compare the results directly to see trends in the data. This section discussed the total number of failures and the subsystem failure percentages for failures that caused the mission to end prematurely. Table 3 shows the overall mission-ending failure rates for satellites is the studies presented above for failures after the satellite reaches orbit. The range is quite large for on-orbit failures and does not prove consistent even within similar groups of satellites. Some reasons for this include the different classes and missions of satellites, the intended design lives of the satellites, and the way in which failures were investigated and recorded. The most important note is that university run satellites are not an outlier in the range of satellite failures.

The next two tables show the total number of failures per subsystem (Table 4) and the percentage of failures per subsystem (Table 5) for each study. These results were detailed in the previous section and were shown in Figures 3-7. In the tables, "--" signifies that no information was given, whereas "0" means there were zero failures in that subsystem. Unfortunately, little information is available for the university-class and FBC satellites, so it is difficult to directly compare them to the other categories.

Table 3. Overall Failure Rate for Satellite Studies

	Study Year and Focus					
Subsystem	Thesis Study	2002, Geosynch[28]	2002, US Military and Civil[29]		2003, GN&C[30]	1999, FBC[31]
Satellite Size	University	Large	Large Civil	Large Military	Large	Mix
Overall Failure Rate	23%	4.1%	24%	19%	4.6%	10% - Traditional 28% - FBC (small)
Total # Satellites	96	1080	186	74	764	28

Table 4. Total Number of Satellite Failures Causing Loss of Satellite

	Study Year and Focus					
Subsystem	Thesis Study	2002, Geosynch[28]	2002, US Military and Civil[29]		2003, GN&C[30]	1999, FBC[31]
Satellite Size	University	Large	Large Civil	Large Military	Large	Mix
Total # Satellites	96	1080	186	74	764	28
LV	22	38	--	--	--	1
Kick Motors	--	9	--	--	--	--
Payload	--	3	26	48	7	5
GN&C	--	--	21	45	13[1]	
ACS	--	12	--	--	7	
Power	8	5	19	33	5	
Propulsion	--	6	2	37	4	
Comm	3	3	3	17	2	
C&DH	1	--	3	4	6	
Thermal	1	0	1	0	--	
Structures	--	5	0	0	3	
Software	--	--	--	--	1	3
Operations	--	0	--	--	1	--
Program Management	--	--	--	--	--	1
Unknown	7	1	0	0	6	1

[1] The 2003 GN&C failure rate estimate includes the failure rate values for ACS, Propulsion, and Software, but the latter three are also broken down separately in their rows.

Table 5. Summary of Satellite Failure Rate Percentages for Loss of Satellite

	Study Year and Focus					
Subsystem	Thesis Study	2002, Geosynch[28]	2002, US Military and Civil[29]		2003, GN&C[30]	1999, FBC[31]
Satellite Size	University	Large	Large Civil	Large Military	Large	Small
Total # Satellites	96	1080	186	74	764	28
Launch Vehicle	58%	47%	--	--	--	9%
Kick Motors	--	10%	--	--	--	--
Payload	--	4%	35%	26%	20%	41%
GN&C	--	--	28%	24%	37%[2]	
ACS	--	15%	--	--	23%	
Power	18%	6%	26%	18%	14%	
Propulsion	--	7%	2%	20%	11%	
Communication	8%	4%	4%	10%	6%	
C&DH	3%	--	4%	2%	17%	
Thermal	3%	0%	1%	0%	--	
Structures	--	6%	0%	0%	9%	
Software	--	--	--	--	3%	27%
Operations	--	0%	--	--	3%	--
Program Management	--	--	--	--	--	9%
Unknown	18%	1%	0%	0%	17%	14%

It is important to note that there were different classes of satellites in each study. Different types of satellites have different kinds of failures, but comparing them will give a sense of whether a trend might exist. Studies so far have also looked at only a few small industry or small university satellites, potentially because this is a newer field with far fewer satellites. Other reasons could include that the information has not been published, or that the satellites are too new and have not yet failed. Nonetheless, there are a few interesting points of discussion.

Many of the subsystems have different failure rates across the studies that are hard to characterize. Some reasons for these differentiations might be the diverse types of satellites, the source of the data and what was included in its information, or how failures were recorded and classified. Complexity has been noted already to cause failures, and complex hardware can lead

[2] The 2003 GN&C failure rate estimate includes the failure rate values for ACS, Propulsion, and Software, but the latter three are also broken down separately in their rows.

to more advanced software, which then introduces even more failure modes. Longer lifetimes also require more advanced and complex systems, which can lead to more failures as well.

While the studies did not provide data to completely fill in these tables, one can get a good sense of some basic trends. Since GN&C, ACS, and Propulsion are very closely related, sometimes the papers report GN&C failures but not the rest, or vice versa. It is a common trend that these subsystems cause many of the EOL failures for satellites. Another difference between satellites groups is the propulsion failure rate, which is particularly high for military satellites. This high rate is probably due to the fact that military missions often keep using the satellite to point to different places until it runs out of fuel, whether it's exceeded its design life or not.

There are large differences between studies in the number of Payload, C&DH/Software, and Unknown failures. While not many studies reported on software failure, a large percentage of FBC missions failed catastrophically to what they claimed were software issues. It is a possibility that missions with shorter development times don't have enough time to adequately test and debug software. Software can also be expensive, and FBC missions might not have enough money to get systems capable of overcoming faults and working around errors. As mentioned above, complexity is also a cause for more failure modes, so complex software can be particularly hard to test and maintain on orbit.

Launch failures are a concern for every payload, but most notably for university satellites. This is partly due to the failure of a Dnepr rocket in 2006, which crashed while carrying 15 university satellites, which skewed the data for university satellites to be focused on launch vehicle failures.[12] The Federal Aviation Administration claims that, between 1989 and 2007, 89% of all launches were successful.[32] Aerospace Corporation looked at launches from 1957 to 1999, and found that of the launches conducted worldwide, 91.1% were successful.[33] This percentage includes launches early in the space race, many of which failed.

In summary, power subsystems are unreliable for small, student-run satellites, while large satellite programs tend to fail in the subsystems related to Guidance, Navigation, and Control (Navigation, ACS, Propulsion, and related software). However, more information is needed for small satellite failures to see if there is a real difference in how these types of satellites fail. The results from these studies may help focus the development process and analysis performed by student groups, and projects can apply risk management techniques from either other universities or industry, which are discussed in the next chapter, in order to try to avoid many of these technical pitfalls.

Chapter 3: Current and Improved Risk Management Methods

This chapter discusses risk management methods at universities across the country to demonstrate the wide variety of plans that schools use. Government risk management practices are also briefly analyzed. Suggestions are then made for improvements to both university risk programs and technical failures on small satellites. Lastly, alternate platforms for satellite missions are presented, and their advantages and disadvantages are compared to those of satellite programs.

3.1 Current Risk Management Methods at Universities

Formal risk management techniques often require a large amount of paperwork. While small programs might consider risk management a good idea, the overhead might be overwhelming. However, all satellite programs are encouraged to use some form of a risk management technique.

Among the universities surveyed, each school seems to use a different risk management method. At many programs, risk assessment is done at a systems level with the team leaders. Few formal techniques are used, and teams often just discuss the risks and modes of operation and then make design decisions accordingly. Bringing in outside help, via periodic industry reviews, seems to help teams focus their efforts on minimizing risk.

In a few cases, the risk management process is more developed, and the major risks are more formally identified and then traded as design choices. Projects with more defined risk management tend to track their risks and document their decisions better. In addition, programs with more guidance from an outside source (e.g. the United States Air Force) have a more defined and regular risk management plan. More formal guidance and regular reviews with an experienced adviser can lead to a better managed and less risky mission. The examples below describe various ways in which schools do use risk management strategies actively in their program.

3.1.1 The Top Risk List

An easy technique for teams to implement is the "Top Risk List" or "Stoplight Diagram," where the top risks are identified and managed. Usually, the risks are associated with a color – red, yellow, or green – to display the status of the risk and its mitigation plan. Green signifies that the plan is working as intended, and the risk is reduced. Yellow means that the plan is not working well and may need attention. Red shows that the plan is not working and that management needs to take action to bring the risk under control.

At the California Polytechnic State University, San Luis Obispo[34] (Cal Poly), the top ten risks for each subsystem and for the overall project are recorded with the team, and the managers of those aspects of the project review the risks each week. This risk management approach requires only a little effort, but it will help keep the entire team thinking about what risks exist and how to mitigate them.

3.1.2 Design-based Risk Decisions

At the University of Missouri-Rolla[35] (UMR), they do not use a sophisticated risk approach, but risk management is certainly a part of their design process. Design decisions are made while considering risk as an important factor. For example, risk is used as a metric in case studies, and the team debates risks as they come up. The students also discuss modes of operation and safe modes with the professors on the project as issues arise. Oftentimes, though, risk management at UMR comes down to whether the project can afford a redundant system and if the mass and power budgets allow it.

Queensland University of Technology[36] (QUT) approaches design-based discussions and decisions through weekly meetings. At these team meetings, issues and risks are brought up by professors and students, and these risks are recorded in the minutes of the meeting. Often, tasks are assigned as action items to someone on the team, and the assignment is left on the agenda until the risk has been taken care of. This method leaves the risk in the hands of a specified owner but also brings the topic up each meeting until the action item is closed. Both of these methods are good ways of getting the team to discuss risk, but they are still ad hoc and do not include analysis to show that all risks have been addressed.

3.1.3 Team Review

Stanford University[37], follows a plan similar to the five step process outlined in 2.1.1. Students assess risk by brainstorming the possible failure modes for the part of the satellite for which they are responsible. To control the risks, the faults are categorized as to whether they need large, minor, or no changes. The students also list the ways in which they can eliminate, reduce, or live with the risk. The lists are collected from the team members, and the system leads and mentors review them and try to incorporate the mitigation strategies into the design.

Throughout the project lifecycle, risks are re-reviewed to determine the current level of risk and the progress that has been made with the mitigation plans. To review overall program risks, the whole team is involved (including students, faculty, mentors, and former students) so that everyone is aware of the risks. A team review allows every student to be involved in the risk management process. This approach engages students in the awareness of and mitigation of risks while keeping the project leaders in charge of the risk management process and decisions.

3.1.4 Two-Phased Approach

Cornell University[5] has a risk plan that uses two different approaches – one for the conceptual and requirements-allocation period and one for the detailed-design phase. At the beginning of the design cycle, while in the conceptual phase, the subsystem functional leads help to identify the likelihood and severity of risks. This assessment decreases risk by identifying areas in which money and time should be invested. Failure modes at the highest level are given a risk assessment score, which is the product of the probability of the risk occurring and a measure of its consequence. Next, the leads look at cost-benefit analyses of the mitigation plans to find the strategy with the lowest combined risk and cost numbers.

When the team reaches the detailed-design phase, the major risks have been identified, so they use a fault tree analysis to continue to assess risk. A fault tree shows how initiating events can lead to functional failures of a design, such as a spacecraft.[38] From this analysis, a system-level probability of failure can be calculated to prove that the reliability requirement has been met at various phases throughout the mission. It is only necessary to perform the entire process once, and then the fault tree should be updated and monitored as the project progresses.

A two-phased approach is very useful because it recognizes the need for different risk management strategies at various stages in the design, but it is more time-consuming than strategies discussed thus far. At the beginning of a project, the subsystem leads use the identified risks to make informed design decisions, and in later phases, the program management can monitor and control risks.

3.1.5 Extensive Testing

A large number of schools rely on extensive testing to reduce risk in a program. Since many schools use commercial off the shelf (COTS) products, one major concern is the effect of the space environment on these components. In addition, many schools manufacture elements of the satellite on their own. Rather than spending time on computational analysis, many teams build components or subsystems and then test them.

Even though some money will be spent on parts that will be wasted, some schools find it is easier and cheaper to do testing rather than spending significant time and money analyzing the design. Cal Poly[34] combines risk assessment and testing in their approach. The Cal Poly team looks at failure modes to identify potential points of failure and critical functions and then focuses the testing on those areas. This combination of extensive testing with risk management saves critical time and money. They have also tried to develop logic diagrams, but it is difficult to identify all failure modes and their probabilities, even for COTS components. An additional benefit from this procedure is that students have the opportunity to learn both from the construction and manufacturing of the system as well as from the testing and analysis performed.

Another method universities use to reduce risk is to utilize space-proven parts, but this might be cost prohibitive. New Mexico State University[39] (NMSU) prefers their students use parts with flight heritage or that are "ruggedized" (parts that have wider temperature ranges, can withstand greater vibration, etc.). NMSU also uses the policy of testing components, and they intentionally buy spares and learn by testing throughout the design process.

Both the amount of funding and the level of experience affect hardware acquisition – whether the project will choose to make or buy some components. By fabricating some parts of the satellite, a school can reduce costs while providing hands-on experience to the students. However, until a team has sufficient experience with building satellite components, constructing elements of the satellite in-house increases risk greatly. Therefore, the choices about hardware acquisition are very situation-dependent, and risk should be an integral part of the decision.

3.1.6 Industry-guided Program

Many academic programs use industry professionals for advice on subsystems, program reviews, or software and hardware resources. For the schools, one of the main advantages of partnering with industry is to lessen the risk increase caused by the low experience level of students. While professors often lack satellite fabrication experience and are often focused on managing the project in addition to providing guidance, industry representatives can be there to answer questions while also challenging the design solutions. For industry, these partnerships with schools are advantageous as well because of the exposure in the satellite community and the experience with potential workforce candidates trained through the satellite program.

Industry representatives have varied roles and guidance between programs, but the QUT[36] uses these professionals as leaders in the risk management program. Industry partners can help to train students in risk management while handing over ownership of the risks to the student team. Business personnel are involved in identifying and reviewing risks, and Queensland University of Technology uses industry help in brainstorming, workshops, and reviews.

The fact that students have less experience than industry professionals is important in the management of projects at QUT. To reduce this risk, a safety officer and risk manager are assigned to the project from industry. This person has access to the documentation and talks with the cognizant engineers to compile a list of the relevant risks, which are included in a master risk log. To help motivate students for identifying risks, both to teach them and because the industry representative is usually a volunteer, the students are brought into this process as "risk hunters" and are not blamed for the risks.

After each major milestone in QUT's programs, a brainstorming session is held to identify all the ways in which the satellite can fail. To best capture all the risks, the university team works with an experienced engineer from industry. The attendees can use the work breakdown structure, operational concept, or subsystem items and then try to identify ways to "break" the satellite. It is important to also study the failure modes of project management and organizational issues in these brainstorming sessions. Once these sessions have been completed, the knowledge must be included in the risk management process by identifying mitigation actions and following through with those steps.

Another approach to include industry help is in risk identification during workshops, which are set up to be a more focused discussion of a risk topic. These sessions have a smaller scope and a fewer number of people. This meeting can be led by an industry representative, but the students set the agenda with guidance from the professors and industry representatives. This way, students receive answers to their questions while getting feedback and focus from experienced engineers or scientists.

Industry reviews are one of the most-used tools to get feedback from sources outside a university. The review board may consist of professors from other schools, representatives of sponsoring companies or organizations, and un-biased industry personnel. The main purpose of these reviews is to analyze the design of the entire project; risk is only a small subset of the work covered. However, due to the content covered by the reviews, these meetings are often not as useful as the previous two methods in terms of identifying and managing risk.

Many teams schedule reviews at the end of a student's work term or before a break from school, so the faults identified are left for the subsequent team. This can be a problem if the risk items are not documented properly or are not transferred to the next students. Risk review sessions or general review boards should be scheduled during work terms or at the beginning of the semester, before class workload increases again.

Student programs should take advantage of the valuable insight offered by industry professionals. Another useful method to review risk would be to have a smaller industry review specifically for risk. Through the various techniques mentioned, the expertise of professional representatives can help guide a program through the risk management process, and they can help minimize the risk associated with the fact that students have little technical experience.

3.1.7 Industry Partner

Some commercial businesses join together with schools very closely in order to develop small satellites. Most of these groups are small businesses, which allows them to apply for funding from NASA. This section will discuss further the balance in workforce and engineering tasks between two such groups.

3.1.7.1 Payload Systems Incorporated (PSI) and MIT's Space Systems Lab (SSL) Example [40]

PSI and SSL routinely join together for space systems development. They have developed a partnership that covers the complete life cycles of spaceflight experiments, and they include students, faculty, research staff, and industry partners. The combination and balance they have achieved allows students to be actively involved in a project while providing large value to the customer. For example, graduate students have written all of the design and control algorithms for these projects, which are often used to test advanced and groundbreaking control experiments, but PSI does the fabrication and other mechanical work.

A number of philosophies drive how the relationship between PSI and SSL has developed. First, the two teams recognize the fact that some aspects of building and developing satellites or their payloads are not well-suited for students due to their education, schedule, or interests. Therefore, students at the SSL work on roles fitting them, while professors, staff, and PSI handle the other tasks.

Second, these teams have a "build it wrong, build it right" philosophy, where building it wrong the first time is an accepted part of the process. Students are heavily involved at the beginning of a program when there are fewer deadlines and better tasks for students. As the development progresses, more professional staff and contractors get involved to prepare for design reviews.

As the design becomes more detailed, the SSL students work on prototype testing, operations plans, mission and data analysis tools, etc. Prototyping is an ideal area for students because they can perform hands-on work without needing a lot of experience. On the other hand, the professional staff and PSI focus on the hardware and software design and fabrication. For example, students build the prototype while PSI helps and tweaks that design from the beginning. PSI then works with MIT to make the design better, at which point PSI builds the spaceflight experiment. After the flight experiment is built, students focus on science and technical testing of the hardware, mission planning, data analysis, crew training, etc. PSI performs the integration and testing.

The students are involved throughout the design cycle, but PSI often performs the engineering hardware work (such as flight hardware fabrication, integration, and paperwork), while MIT focuses on research and analyses relevant to the payload. This sort of collaboration works well for the SSL, where the graduate students do more in-depth research on the purpose of the mission and do not focus on the subsystems and their layouts.

There are many advantages to partnering with a small company such as PSI, aside from potentially gaining Small Business Innovation Research grants. These companies often have many years of experience and advice. PSI, for example, is familiar with the Shuttle's and the International Space Stations' safety protocols, making the difficult and long process of Safety Assurance relatively easy.

PSI is only involved when necessary – students do most of the beginning work and then help throughout the project. PSI focuses on meeting fabrication and integration deadlines, implementing configuration control procedures, and other tasks not fitting for the academic environment. This plan helps to ensure that the project cost is kept as low as possible while maintaining a high-quality spaceflight experiment.

Third, a partner company can help keep the project on schedule. Universities can often get sidetracked by research, but a development-focused company can help be the project's schedule-keeper. Finally, PSI is located near MIT, and face-to-face meetings and reviews can also substantially help keep a project running.

3.1.7.2 Quakefinder and Stanford[41]

This project was a small business/university collaboration between Stanford, students in a Systems Engineering course at Lockheed Martin, and Quakefinder. This venture came about because Stanford wanted to build a small satellite, and Quakefinder had a payload they wanted to fly, but no satellite. In order to get this satellite off the ground, the small business offered money, equipment, and personnel, Stanford had students, and Lockheed Martin offered its facilities. The team also decided on a fast schedule – the class worked on the project for nine months, and with continued help from the project's supporters, it was built and tested within about 1.5 years.

In contrast to PSI, the staff at Quakefinder was specialized in different engineering and science fields and did not have experience in building spacecraft. Therefore, many people, even on the industry side, were doing tasks for the first time. A Lockheed Martin review board helped the student/industry team to focus their mission, noticing mission creep and high risk items. This review helped them to identify and manage risks.

Looking at their high risk items and “Lessons Learned” from other CubeSats, this team increased effort in those areas and employed redundancy where it seemed to be needed. They also performed fast, non-traditional testing, such as pointing the spacecraft to an antenna across rooftops, in order to ensure basic performance. This sufficed for their project, but this could have been a problem with some designs. These ad hoc tests did get the job done in this case, but they were challenging for risk managers to accept because the tests must demonstrate that it will verify the adequacy of the design or production.

Overall, the Quakefinder project was successful. The satellite is still working while exposed to the sun, and it has been collecting data, so the team’s unconventional yet thoughtful risk management and testing plan worked.

3.1.8 Intense Guidance

The University of Toronto Institute for Aerospace Studies’ Space Flight Laboratory[42] (UTIAS/SFL) has played a major role in developing, launching, and operating Canada’s first microsatellite and space telescope. This lab also maintains Canada’s only current nanosatellite program, which involves both staff and students in the development of satellites under ten kilograms for both technology demonstration and small satellite experiments.

The success of these programs are critical to the nation’s space program, so UTIAS/SFL uses a full-time staff of professionals (five to six people) due to the program’s mission reliability requirements. This staff manages the risk issues, creates mitigation strategies, and carries out those plans. To identify risk, systems engineers on the professional staff review the system and subsystem level design, and their work is reviewed by the rest of the team. Their philosophy is to ensure that mission critical failure modes are designed out of the satellite so that the system is recoverable regardless of software or command errors. The staff then continues to assess and monitor risk throughout the development process.

Intensely guided programs can be a good risk management learning experience for the students, but the students in these projects generally participate only in the discussions – the experienced engineers provide the risk analysis.

3.1.9 University Nanosatellite Program (UNP)

The University Nanosatellite Program[43] of the Air Force Research Laboratory (AFRL) is a two-year recurring university satellite competition. Student groups design and build satellites that go through a rigorous review process, and one project is chosen to proceed to final integration and test at AFRL. However, many programs continue on if not selected, helping students to continue to gain more experience.

Through partnerships with the Air Force, NASA, the American Institute of Aeronautics and Astronautics, and industry representatives, the universities receive funding and guidance from their participation in the UNP program. The Program Office at AFRL provides direction by encouraging certain design standards, program requirements, and constraints. They do not specify the type of risk management to be performed, but AFRL personnel also give significant management and systems engineering expertise to help with the latter stages of the design cycle in order to get the satellite into orbit.

The UNP process is working – approximately a dozen schools participate in each two year program, and the students are getting the larger amount of support that they need. The ultimate goal of getting their spacecraft through the competition and into orbit is an excellent source of motivation to make quality design decisions and to adhere to industry-standard best practices.

The AFRL is also working on their own projects in which they employ fairly relaxed risk management methods. TacSat-3 is a collaborative effort of the Air Force, Army, and Navy, but it still is considered a small, cheap, and fast satellite program by government standards. This type of program is ARFL's step to responsive space. For this type of program, AFRL uses a more "ad hoc" risk management approach because they don't have the time or resources to implement military standard risk management programs.[44]

For TacSat, the three systems engineers on the AFRL team created a subjective list of the 14 top risks of the program. This method was chosen because it is a bottom-up review process with minimum cost and schedule overhead. They next formulated a plan to periodically measure, judge, and mitigate the risks. Other factors in their plans put an emphasis on forming experienced review teams to look at risk as opposed to the costly and time consuming modeling and testing done so often in industry. AFRL is a good example that industry is loosening its risk management policies for smaller missions, but the result is to be determined.

3.1.10 Tools

Many schools use simple and widespread programs such as Microsoft Excel to track risk, while The Cornell[5] teams use Microsoft Project as a server application for risk management. These school use a tool that requires very little learning curve. Logic diagrams and fault trees can be created in Microsoft Visio, which is widely available at universities.

In industry, more complex programs are used to perform failure mode identification. For example, FaultTree+ and RiskSpectrum allow the user to do event tree or fault tree style risk assessments. BlockSim is a simulation code that is used to help do reliability analyses, and the DecisionTools suite can be used to perform general work with probabilistic distributions, and uncertainty and decision analysis.

3.2 Risk Management in Industry

Risk management in industry often goes beyond the five step process outlined in Section 2.1.1. All military and government satellites must follow military risk management strategies, which are available to the public. There is little public information about the exact process that industry sponsored satellite programs use, but the techniques are most likely very similar. Therefore, this section will look into the risk management strategies set forth by the government.

The risk management, or system safety, procedures are outlined in a military standard (MIL-STD) document entitled Standard Practice for System Safety.[45] There are also other plans that apply to system safety and failure mode prevention, including MIL-STD-498 and DOD-STD-2167A on Software Development and DOD-STD-499B on Engineering Management

In the System Safety plan, at least one person (if not more) is assigned to be a risk manager, who is responsible for everything relating to system safety, which is a highly involved process. This one MIL-STD document is 113 pages, and it requires no less than 14 separate forms, with multiple kinds of categorizations for mishaps, risks, failures, safety levels, etc. The MIL-STD-882E system safety requirements, which also each have multiple action items, consist of the following:

- Documentation of the system safety approach
- Identification of hazards
- Assessment of mishap risk
- Identification of mishap risk mitigation measures

- Eliminate hazards or reduce hazard risk through design selection
- Incorporate safety devices
- Provide warning devices
- Develop procedures and training
- Reduction of mishap risk to an acceptable level
- Verification of mishap risk reduction
- Review of hazards and acceptance of mishap risk by the appropriate authority
- Tracking of hazards, their closures, and mishap risk

It can be inferred that this list of requirements and all the associated work to fulfill the requirements would easily overwhelm a student group. There would be advantages, though, to utilizing this complex system on student projects. First, the MIL-STD method is very comprehensive. Every step is recorded, and there are plans in place to deal with all issues that arise. The process also makes the systems engineers think about every form of risk that may occur and how everything is in need of mitigation and monitoring. Second, there is someone always dedicated to the task of system safety, which includes risk management. Large companies often have dedicated resources for system safety, especially for high-value missions.

On the other hand, there are many disadvantages to applying the MIL-STD process to student satellites. If the entire plan were followed, all the resources available to university satellites would be spent on risk management, leaving nothing for the engineering development. Since this is obviously a problem, the students could focus in on a subset of the requirements laid out by the government process. Some of the requirements are very important for any project, but others may be of limited use to a small satellite program. It might be time-consuming for a student program to pick and choose what's needed from the MIL-STD list since they have little to no experience with these topics. To identify what students should do for risk management, one can notice that there are similarities with the MIL-STD list and the definition of risk management detailed in the first section. It would be best to apply those steps to small satellites since they are both manageable for a student group and common in industry.

While government risk management approaches may help reduce risk, success is not guaranteed, and it is impossible to apply their techniques to student programs. There have also not been any public studies on whether these system safety guidelines have been the reason for a reduced number of failures in satellites or if they mitigate certain kinds of risk. However, students should judiciously apply risk management of some form to their projects, and it might be best for them to borrow practices and standards from industry or government where applicable.

3.3 Suggestions for Improvement of Programmatic Risks in University Risk Management

Throughout this paper, risk management at universities has been shown to be inconsistent. If a university would like to reduce risk and hopefully increase their success rate, the program should improve their risk management methods, borrowing from current university, government, and industry practices. Some very basic strategies can be used to reduce overall risk of mission failure. The following practices should be emphasized in a program from the beginning.

3.3.1 Funding and Competition

Demonstrating a good grasp on risk management is necessary to get funding and win competitions. Reducing the risk of an overall mission can be accomplished by allowing significant risk only in new research areas. Risk can be minimized elsewhere by using hardware with flight-heritage as well as standard algorithms and processes. Design re-use must be done carefully, but it can be valuable for high-level systems as well as at the electronic component level. In cases where the risks are inherent to the mission, the program should maximize the satellite's value for the sponsor by focusing on the needs of the sponsor and designing to meet those requirements.

3.3.2 Experience

Continuity in student leadership is important in order to keep corporate memory as well as push the project forward. It is necessary to prepare students for these positions. One way to do this would be to have a training/mentoring program. Other options include having students take technical classes, participate in leadership seminars, or attend conferences.

To reduce the risk associated with a student's lack of experience, it is necessary to train people working directly with the subsystems to identify risk. Application-based training or academic classes are one option, but this might be too difficult to implement in a small program. A different option is to create a database of failures that can serve as a reference for students who must identify risk. By building a template of risk failures, students have a list from which to work. This idea will be discussed further in Section 4.2.

3.3.3 Staff

Personnel management is an area in which consistency will lower risk. One option is to hire a core group of people to ensure the most important work is being completed. By tying the staff's work to a salary, they are obliged to work a set number of hours per week, and there will be more consistency in the program.

Even though staffing can be difficult, student groups should try to not have only one person working on each subsystem. By having every member of the team working on two tasks, there will be no single-string workers, and the members of the team have someone to help and motivate them. This may not be feasible if there are too many jobs and not enough students, but that might also point to the idea that the task is too complex or the team size needs to be bigger. Sharing assignments also helps communication among groups since people are working on more than one task. This distribution of jobs might be difficult to maintain, but it would be beneficial for the students and for the team's communication.

When working with students, there are times in the semester when workload is lower, and more time and energy can be devoted to extra-curricular activities such as satellite programs. During these times, the projects should focus on critical items such as risk mitigation. After student breaks and at the start of a semester are ideal times to review risk and ensure that students are staying focused on risk management.

3.3.4 Direction

The professors and students must work with all of the groups that oversee the project to define the mission and its requirements from the start of the program. It is then necessary to keep track of requirements, their sources, and the rationale behind them as well as flow them down to the rest of the system.

To maintain the proper focus, the mission and its requirements must be clearly defined, and the students must have oversight of their work. Setting mission goals and strict requirements helps students to take the project seriously, and project goals and supervision help students make progress in the correct direction.

3.3.5 Schedule

Falling behind schedule at any point in the design life can be a huge barrier to meeting a launch date. This is because not being on schedule will drive up costs and encourage risk taking in order to meet a (most likely inflexible) launch date. From the start, everyone must take the mentality that they are on a flight program, and deadlines must be met, but not by dropping risk management. The team should build a reasonable schedule, with margin, and one way is to step backwards from the launch date using key milestones (driven by integration), and stick to the review schedule that is set by the launch date. This extremely important job should be the job of someone with experience and that can enforce the limits on the team.

The strict deadlines imposed by an immovable launch date can lead to poor and hasty fixes of a student-satellite design. One solution to this problem is to implement a tiered design. By using spiral requirements, the team can do multiple releases of the satellite based on the tiered requirements, but every release can fly. This approach can be done for each subsystem or for the system as a whole. However, this staged release approach can be a hard method to implement for teams with little experience and low funding.

NASA Langley Research Center suggests that small satellite programs should focus on risks preventing the completion of a milestone. Instead of focusing on the risks that threaten individual technological advancements, the team should examine what the risks are that must be overcome to reach program milestone completions. The resources can be reallocated to better ensure that the next objective is reached.[46]

Oftentimes, the staff of a project can get discouraged or pessimistic about a project, leading to its delay and eventual cancellation. The Quakefinder[41] group tried not to let that happen to them, and they have the following suggestions. A project must be approached with the mindset that the team will succeed, and even if there isn’t enough time, the team must be creative in its solutions. A solution that is “good enough” is often the way to proceed, as Quakefinder demonstrated. In addition, creating infrastructure in a class or program for multi-year programs would be beneficial to avoid “reinventing the wheel” and slowing progress.

3.3.6 Documentation

Throughout the aerospace industry, its professionals are not good about passing on information to future generations. This will be a large problem as a large portion of the workforce nears retirement, and the same issue plagues student satellite projects because of their high turnover rates. Retaining students as they transition from undergraduate to graduate school is a great advantage for a satellite program. Many schools encourage students to stay for their graduate work, but this cannot be guaranteed.

Therefore, it is imperative that the universities have a good method for transferring information from one year to the next as students graduate or otherwise leave the program. A consistent and enforced policy should be created in order to pass along the important issues in designing, integrating, testing, risk management, and operating space missions. This knowledge transfer can be better controlled using consistent documentation and knowledge-management tools throughout the program.

Consistent application of well-defined procedures, guidelines, and documentation can help to minimize careless errors while maintaining knowledge transfer. The management of the program should also stress that documentation is a necessary aspect of the engineering design process. An online documentation system can be used to keep documents in a central location with version and access control.

Concept tracking and keeping up-to-date information is aided by a configuration management process. Configuration management, in which revision control, change control, and release control are regulated, can address many common problems before they escalate. Many schools do not use a system to control their documents, but without this method, students will not know what information is current, how and when decisions were made, or what aspects of the design have been set. It would be best to have the system run by students but be based on a proven system, using guidelines from industry. This is one example of taking standard practices and adapting them to student situations. These processes reduce risk by helping to document work, communicate changes, and maintain consistency as the staff varies.

Another way to minimize the effects of turnover is to break work into smaller tasks that can be completed in a semester or two. Then, the student can write a final report on that specific section of the work before he or she leaves the program.

Programs that are similar at a large number of schools (e.g. CubeSat) have a unique ability to reduce risk. The teams should share information on lessons learned, probabilities of failure, high risk areas, etc. Many people may have thoughts on their own failures or suggestions for other schools, but they have not translated these into a resource for other programs to use, or if they have, they have not publicly shared them. By distributing this information, schools are less likely to make the same mistake, saving time, money, and frustration and reducing risk. An infrastructure, which could be hosted on a common website, is needed for universities to actively carry out this plan.

3.3.7 Recruitment

Student-led programs should continually train and recruit new members, especially younger undergraduates that have the potential to be involved for many years. It is possible to recruit older students by offering credit or research for their senior theses or independent studies, but only some schools have those options.

Since university teams have difficulty getting a full team of experienced students, these teams should aim to have a three-tiered student structure.[6] First, it is necessary to have student experts with knowledge of all aspects of the design. Second, people with corporate memory are valuable to a program to help discuss design decisions and rationale. Third, a large group of semi-knowledgeable students can help with design tasks and when a large staff is needed. Recruiting and maintaining this type of team will help keep the project balanced and headed in the right path.

3.4 Mitigation of Technical Risks

While this paper focuses more on the identification of risks and not necessarily their mitigation, understanding the mitigations of risks helps one to identify possible failure mechanisms. Therefore, in this section, some mitigation strategies for technical risks will be presented. This discussion will not be comprehensive of all types of technical failures, but it will cover mitigation strategies for most of the technical failures that have been discussed thus far. For more information on technical risk mitigation, see the sources outlined in Table 1 as well as other published papers on lessons learned.

3.4.1 General

Presented in this section are overall design considerations that can reduce the technical risk of a mission. Later, risk reductions for systems engineering, launch, and specific subsystems will be discussed.

Payload failure is seen as a large cause of a mission ending early (Table 5). It should be obvious that well-tested payloads are necessary for mission success. In most cases, a satellite can still function without its payload, but the point of the mission is lost. For university satellites, which are sometimes more technology demonstrations rather than science missions, payload failure is slightly less important. However, more schools are launching scientific payloads, so a reliable payload design is needed.

Commercial off the Shelf (COTS) parts can greatly reduce the technical risks in a program. Having experience with a component can be invaluable. By using COTS parts, missions can usually achieve higher performance and reliability with lower cost, as well as potentially less mass, power, and volume, all of which are important to spacecraft design. COTS parts were shown to be reliable in the areas of structures, thermal systems, batteries, and electronics.[3] Universities should use this heritage to decrease technical risk in their programs. These types of components are more readily available and can be more reliable, at least in the environment for which they were designed. Therefore, it is necessary for someone who understands how the space environment affects components to evaluate whether the COTS part is a good match for the satellite.

Hardware failure is a large issue for any project. Redundancy is often considered as a means to reduce risk, especially to design out single point failures, but one must first pick the correct form of redundancy (if any at all) for their needs. The types of redundancy include the following: same design redundancy, k-out-of-n redundancy, diverse design redundancy, functional redundancy, and temporal redundancy.[47]

Same design redundancy involves installing two identical components with a switch to make one active. K-out-of-n redundancy, where a pool of spares can be assigned to replace any one of the active units, is similar to same design redundancy, but it's sometimes cheaper when used for components such as data storage, multi-cell batteries, solar panels, etc. In diverse design redundancy, two or more components that have different designs are used to provide the same service. This gives high protection against random and design failures. Functional redundancy (sometimes called analytic redundancy) is when one component can serve a different purpose from its intent. The backup components may result in reduced performance, but it avoids increasing cost and mass of the other types of redundancies, and there is also protection against some random and design failure. Finally, temporal redundancy is the repetition of an unsuccessful operation. In addition, one can also change the software or the operations to accommodate an unplanned-for event or failure.

Redundancy increases the complexity of the spacecraft and is not always desirable for student spacecraft, which aim to be simple and are highly constrained on volume and mass. To increase the likelihood of success, resources can be focused on increasing component reliability instead of incorporating backup systems. Teams must remember that redundant systems can add mass, consume power, require more support hardware and software, and cost more. Therefore, designers should evaluate these factors when considering the options for redundancy in their systems.

To reduce the risk of using an unproven technology, new components can be used alongside space-proven parts. This diverse design redundancy includes new space hardware or software, but it is relatively safe for the mission, if done correctly, and it is an excellent method to achieve technology insertion. This practice has been performed for years at Surrey Satellite Technology Ltd[48] (SSTL) and works well for their business. Universities could also capitalize on this idea and aim to get funding specifically for testing new technologies in a non-threatening manner.

Whether the satellite has redundant components or not, it may prove valuable to buy backup systems of long lead or risky items in advance. Therefore, the program should be structured around component for which the project can get backups and not utilize unique components. As for the expense of backups, Mark Maier of the Aerospace Corporation states “If you can't afford three of something, you can't afford one of them.” [24] In other words, the team should not buy parts if they can’t afford buying three of the same component.

Robust design can also prove to be a valuable risk mitigation method. If the mission allows, over-designing subsystems can save the mission from failure. For example, SSTL’s three axis spacecraft often carry solar panels on all sides – not just the side facing the sun.[48] This kind of robust design allows for the mission to continue in, and hopefully recover from, some off-nominal situations.

If the university is planning on doing multiple missions, it may make sense to follow SSTL’s[48] practice of creating modular microsatellites. Their satellites in this class range from ten to 100 kilograms, and their platforms are driven by the desire to have modular and flexible designs. These standard satellite platforms can be used for multiple missions. This method keeps cost low by re-using designs that are appropriate for different missions and that have proven past performance. This practice should also reduce the amount of documentation, analysis, qualification, etc. needed for each satellite. Modularity can also make replacing old parts easier, and assembly and testing will be quicker.

3.4.2 Systems Engineering

Many satellite failures come from inadequate or incomplete systems engineering. This type of oversight can come either as a lack of systems supervision in the design process or a lack of systems testing. It is necessary for projects to have a competent and trained systems engineer in order to ensure that the entire spacecraft will function as a whole and to ensure that adequate testing is performed.

3.4.2.1 Systems Engineering Supervision

Swartout[3] suggests using small spacecraft with fewer parts as well as common interfaces, especially between the launch vehicle and the spacecraft, to reduce risk. Other program- and systems-level risk reduction strategies can include short mission duration, large operational margins, and thorough functional and environmental testing.

Managing and monitoring the design from a systems-engineering perspective will also help reduce technical risks. Flowing down requirements and ensuring verification of each requirement will help to ensure that the mission goals are met. Trade studies can also be performed while considering risk. This method is useful to look at the cost-benefit analysis of certain designs given their risk levels. In addition, trade studies can help determine what types of mitigation solutions should be utilized.

A good systems engineer should help focus attention on risks that affect the entire system, such as designing all subsystems to withstand the space environment to a specified level, preventing contamination, quality control, etc. Another important aspect of systems engineering is making sure that changes or repairs do not propagate negatively into the system. Management techniques such as configuration control help prevent this problem, but a competent systems engineer must be part of the team that evaluates changes that may affect the whole system.

A de-scope plan can be used in conjunction with a risk management program to lower risk. De-scope options are technical areas of the satellite that be changed to reduce cost, mass, power, etc. Even though de-scope options reduce performance of the mission, it is important to identify elements that can be de-scoped while still maintaining the baseline mission. A large area of programmatic risk is the underestimation of schedule or money, and having viable de-scope options can help keep the satellite within its allocations.

3.4.2.2 Systems Testing

Testing components on the ground is critical in any satellite engineering project. There are a few suggestions that may help catch more problems in the testing phases. First, subsystem vacuum and thermal testing will help to identify areas of the design that need improvement or rework early on. Second, care must be taken to model the space environment well so that the design margin will be insufficient.

Third, the team should test components based on observed anomalies from other, preferably similar, flights. While integrated testing of the satellite is necessary and useful, testing components that have a higher likelihood of failure is also valuable. For example, pyrotechnics and high-velocity propulsion have shown to cause many in-flight catastrophic failures.[22, 28] By testing these components on the ground, if possible, the project may be able to reduce the risk of early mission failure. Focusing testing on components that have a high likelihood of failure could prove less costly and more advantageous to the project than testing the entire system early on in the design life.

Finally, a functional prototype (a replica of the flight hardware used to test the design) has been suggested by many as necessary to perform ground testing prior to finalizing the design. If the flight hardware is not too expensive, then testing before the design has been completed can help identify many errors that would have come up in the pre-launch testing, which is sometimes too late to fix the problems before launch. Of course, all of these activities must be done early enough in the design cycle to ensure that enough time exists to fix issues that arise.

3.4.3 Launch

Even though catastrophic launch failure is outside the team's area of responsibility, it can be unavoidable, and the students should minimize launch risks by choosing a launch provider wisely. The appropriate launch vehicle for a project will be a balance of risk, capability, and cost. In addition, two of the studies presented in Section 2.4 have failures that stem from satellite-launch vehicle interface failure.[3,22] However, risks associated with the satellite-launcher interface can be mitigated partially by the satellite team. As with any interface, close attention must be paid to requirements for mass and power as well as the reliability of mechanical and electrical connections, including release mechanisms.

One option to get to a space on a launch vehicle or the Space Shuttle is through the Department of Defense (DoD) Space Test Program. The Space Experiments Review Board annually compares and then ranks the proposed experiments and satellites based on their military relevance, and they then try to find a launch opportunity for those payloads. The Space Test Program provides even more for its chosen payloads, such as launch integration (including multiple payloads on a multi-payload bus), representation in meetings with the launch vehicle providers, communicating with the launch ranges, etc. This program eases the difficulty of finding a launch vehicle and working with launch vehicle integration, provided that the mission is relevant to the DoD.[49]

3.4.4 Structures

While structures fail less than other subsystems, complex structures with less heritage and more moving parts (such as inflatable structures, deployable booms, etc.), are probably more likely to fail than traditional structures. Components such as telescoping parts, deployable panels, weak springs, tethers, etc. can more easily malfunction and cause the satellite to fail. Attention should be paid to careful design and testing of structural mechanisms to avoid failures from complex systems.

Mechanical systems are also usually places for single-string failures because redundancy is hard to design into the system. Testing mechanical systems can also be difficult because many components in the structures subsystem are single-use, making testing of the flight components difficult. The space environment, with different loading, vibration, radiation, etc., can also be hard to simulate for student satellite with limited testing budgets. Finally, with long lead times to launch as well as launch delays, mechanical systems can lose lubricant or can otherwise corrode, so the team must take those delays into account when designing the system and planning operations.

Testing designed for sensitive mechanical structures and batch testing (testing one-use components from the same production line to determine reliability) could help decrease the risk associated with the structural subsystem. In addition, pre-launch testing of devices with lubricant, if possible, would ascertain whether the vehicle is ready to fly.

3.4.5 Power

While every subsystem is important in the spacecraft, the power subsystem is extremely critical to achieving full design life, and this subsystem has been shown to commonly fail in university satellites. While further study must be performed to better understand power failures, it was discussed in Section 2.4 that battery charging and high-density power sources fail more than other parts of the power satellite subsystem. With the complex systems required for energy collection and high degradation of solar panels from extreme environments of space, solar panels should be designed and thoroughly tested for the mission.[22] In addition, since there are so many power system failures on orbit for small satellites, schools must make an effort to better test these systems before orbit. Testing practices may help identify failures that would cause failure on orbit, allowing teams to fix the problems prior to launch and proceed with the mission. For example, during pre-launch testing, it was discovered that the Quakefinder satellite had one solar panel not working. They could design enough margin into the rest of the system, though, to be able to fly with a broken panel and not delay the mission. [41]

Perhaps with limited budgets power testing is difficult to perform, but other means of verification may be possible. For example, better modeling and analysis may be a way to reduce failures on orbit. Or, demonstration of the working system under a few extreme conditions may be helpful to locate any inter-subsystem problems. Other difficulties in testing may be due to the power systems complexity and its reliance on other subsystems, which may be hard to model.

3.4.6 Software

Robust software can prevent many anomalies from causing critical failures. It is possible that some mechanical or design failures could potentially be fixed by software control, so software should be designed to eliminate or overcome all failure modes. Software can be a very expensive and time-consuming part of the mission in order to be well-designed and tested. Despite the resources needed to develop robust software, it is important to rigorously manage and test software and all its components, starting early in the design life.

3.4.7 Thermal

Most likely due to a good understanding of the space thermal environment and extensive thermal testing prior to launch, thermal failures were not high on the list of mission-ending failures (Table 5). However, the thermal subsystem must make sure that components are in their operating temperature; otherwise, an off-nominal thermal condition may lead to another subsystem's failure. Thermal cycling can also be an issue for components in the spacecraft. For example, one mission did not realize that flexible solar arrays are susceptible to thermally induced vibrations.[22] This kind of thermal error can lead to solar array degradation because of the thermal cycling, or it can lead to total power (or structural) failure if the vibrations get too large. This example demonstrates the need for comprehensive thermal testing prior to launch.

3.4.8 GN&C

The Guidance, Navigation, and Control subsystem is shown to be a large cause of EOL failures across satellite groups (Table 5). Remember, GN&C also includes subsystems that are directly involved in GN&C activities, including ACS, Propulsion, software, and ground operations.

The main area for improvement in GN&C is the hardware. While not in the scope of student missions, gyroscopes and reaction wheels with longer intended design lives must be developed to help mitigate risk. Universities, though, should make an effort to utilize hardware with heritage and ample expected design life if these components are mission critical. Other hardware, including pyrovalves, thrusters, processors, receivers, etc. also cause mission ending failures, and therefore, need to be carefully chosen or designed and tested.

3.4.9 Radiation

Radiation can affect any spacecraft in orbit, but how radiation affects the electronics onboard depends on the type of radiation event encountered, which includes single-event upsets and single-event latchups. Single-event upsets, or bit-flips, are the most common radiation-caused events, and they can be fixed with error detection and correction software that locates and reverses the event. Single-event latchups are of more concern, causing the part to draw excessive power and to no longer operate until the power is shut off and then turned back on again. While resetting the circuit by cycling the power often works, a single-event latchup can destroy the component or other power-sensitive electronics if the power supply cannot handle the current, and they are an annoyance to the team because they disrupt the mission. The last type is a single-event burnout, but these are not recoverable.[50]

3.5 Alternatives to Satellite Platforms

While satellite projects can be engaging and inspiring, they are also very expensive for a project that has a primary objective of teaching engineering to students. Other engineering projects exist that will provide similar experience to students, but they can cost substantially less. These projects have similar risks and failure modes and, therefore, can be an excellent learning experience for students. They are also a good method to both train students on engineering and risk management aspects and to put an infrastructure in place before working on a satellite project. Platforms such as high-altitude balloons, sounding rockets, CanSats (satellite-like payloads that fit in a soda can but do not reach space), or piggyback space experiments are good alternatives to building and launching a satellite.

Satellite programs are useful for collecting certain kinds of information and for long-duration missions, but the non-satellite platforms have other advantages. Balloons can obtain unique types of data from the same location for multiple days, and the other platforms can collect different types of data and perform tests on new technologies. Students may also be able to do more hands-on work with the hardware and try new and unproven instruments in a cheaper manner.

In addition, these different platforms have fewer programmatic risks – they can have lower cost, less-stringent schedules, less technical risk, more relaxed interface requirements, fewer managerial tasks, and a good match of project length to the time a student can spend on the project.[51] However, in many people's eyes, they are less prestigious than satellite missions.

Of high concern to all projects is cost. For comparison throughout the next sections, the cost of student-run satellites must first be discussed. University satellites cost tens of thousands to millions of dollars, but most are in the low hundreds of thousands of dollars. Cal Poly estimates that a CubeSat (a 10 cm cube with a mass of up to 1 kg) of "medium-sophistication" costs around $100,000-$125,000, including launch (which is about $40,000, currently). Schools have produced CubeSats for less but with much cheaper and lower quality components, while other schools have produced satellites for much more.[52] The next sections will investigate the alternatives to satellite platforms, how they perform risk management, their cost, and their failure rate, where the information is available.

3.5.1 Balloons

Balloon projects tend to involve more professor participation while utilizing both engineering and science students. The main focus of these programs is to collect research quality measurements for further study by the professors and graduate students. Most schools develop just the payload with a commercially-supplied balloon, but others work more on the engineering design of the whole system. Many programs do show, though, that they are able to combine the goals of engineering and science students to create a challenging and relevant project.

Just as in satellite projects, there are a variety of ways in which programs handle their management process, but all of the schools reviewed do perform risk management, exposing their students to this important practice. At the University of Washington[51], the managerial aspect is under the control of the faculty and classroom teaching assistants. Students focus on the development of their instrument, but the systems engineering (including risk management), logistics, program management, etc. are handled by the professors working with the project. University of Washington has noted that balloon missions are better than satellites to test potential failures because components and test flights can be less expensive.

On the other hand, a joint project between the University of Hawai'i[53] and the University of New Hampshire has an extensive risk management program that involves their students. The balloon mission is broken up into multiple sectors, and the people working on that section identify the potential risks. Then, the team comes together to discuss each risk. This process is started from the beginning and continued throughout the design with constant communication between sectors to ensure risks are dealt with. However, it should be noted that this project has nearly one-to-one professor to student ratio as well as employees from the National Oceanic and Atmospheric Administration (NOAA). This high ratio may help to have a more defined risk program.

One balloon project that is run similar to satellite projects is at Iowa State University[54] in the Spacecraft Systems and Operations Lab. Their project team adheres to written procedures similar to what other high-altitude balloons and NASA use throughout the lifecycle of the project, which is in the form of a checklist. They also focus on risk at a series of reviews. To prepare the checklist, the students do a fault-tree diagram of the project to show where the risks could be. Their students are also working on a method to track risks once they've been identified.

Oftentimes, there is a large difference between the intent of a satellite project and the intent of a balloon project. Balloon missions, and their students, are more focused on the science than on the engineering, and the platform is just a means to get to the science. Sometimes, the students learn little about the engineering design process because they are involved in the science aspects of the mission, but it depends on the project and its purpose. A good balance would be to put a team of engineers and scientists together so that both sets of students could take away valuable lessons.

Balloon programs span a wide range of costs, from very affordable to more expensive than CubeSats, but they might not necessarily be "cheap." According to The University of Washington[51], a grant for the development of a single instrument over three years is typically around $200,000, and a whole balloon project will be around $500,000. These grants support everything needed for the project, including one to two graduate students, materials, professors, launch, and operations. Balloon handling is sometimes leveraged from launching services for large projects, and for smaller projects that can be hand-launched, the team handles the logistics or has to pay around $100,000 for industry support.

On the other hand, Iowa State University[54] runs smaller projects (1200-3000 gram balloons) for around $400-$900 per flight for the balloon, helium, vehicles, and miscellaneous hardware. For operations, the students and professors are paid, which brings the cost up to about $1000-$1500 per flight. The research and development costs are separate, but they vary depending on the mission and can't be generally categorized.

Failure rates can be difficult to classify in the same way as was done for student satellites. On balloon projects, students have extensive guidance from professors and are therefore usually not responsible for the whole design. Nonetheless, there have been cases of student-led hardware failing in flight. In one instance, a student's system to rotate the balloon payload failed in one flight out of five.[51] In another, a component failed in one flight, was modified, and worked in the next.[51]

At Iowa State[54], the failure rate has been low – about 5% of over 90 flights thus far, including about 70% student-designed electronics. This low failure rate may be due to their strict observance to standard industry procedures as well as their focus on everyone's involvement in the risk management process. There have only been two major failures (neither in the electronics) for the graduate research group. In addition, there have been four failures for cutdown of the payload, but these have not been attributed to a specific source. On the other hand, the senior design class had a 60% failure rate, but they were much less experienced at the tasks assigned. In general, this university shows that students can be involved in many balloon flights in a cost-effective and reliable manner.

Industry-supplied balloons and support are not failure-free either, and these can contribute to mission degradation or mission failure. Some failures include premature payload cutdown, the balloon not reaching the correct altitude, balloon power system failure, telemetry ground station failure, and temperature-induced failure of the communication system.[51]

With this limited data on balloon missions, it is hard to compare their "worth" to satellite projects. However, with launches and test flights easier to perform on balloon experiments, it is possible to turn failed hardware around and get another flight sooner. This, of course, is not free, but since the payload is often returned to the team, a re-flight for a balloon payload can be more cost effective than building and flying a whole new satellite. In addition, this returned hardware can provide an opportunity for corroboration of failure analysis with the flight hardware, which is something most satellite missions can not do. Failures are still costly enough to teams, though, that they are a good incentive for focusing on mission success.

3.5.2 Sounding Rockets

In the same manner as balloon experiments, sounding rockets have many advantages over satellite missions. Sounding rockets provide a way in which payloads can get to high, suborbital altitudes to run experiments or test new technologies for a moderate cost. Experiments can be run for space, microgravity, or Earth science research, and the payloads return to Earth, landing with a parachute system. The capability to return the payload to the owner can further reduce cost by being able to reuse the payload or some of its components.

Students in their undergraduate or high school career can participate in the Student Launch Program or the NASA Student Involvement Program, which gets students involved in the full research process. Sounding rockets can provide a great research opportunity, especially for graduate students, who can focus attention on the payload design and not on the engineering of the sounding rocket. In addition, sounding rocket launches are easier to obtain than launch vehicles, which can be useful for scheduling for student groups. Sounding rockets are also a lower risk investment because they are less expensive and have more heritage than university satellites. As with balloons, they also have faster turnaround time and provide good information on failure modes if they payload is returned to the university.

No information was available on the risk management practices or of failure rates for schools using sounding rockets. Further investigation in this area is necessary.

3.5.3 CanSats

The Annual CanSat Competition[55] is open to university and high school students across the United States, Mexico, and Canada. The students are involved in the project over the entire life cycle, giving them hands-on learning from design to integration and test to satellite operations. While these CanSats do not reach orbit, they do provide a platform for designing, building, and launching space projects and are an excellent way for a university to get started in the satellite business.

CanSats are by far the cheapest platform – schools have a limit of $1000 that they are allowed to spend on the materials for the satellite and other outside services to enter the CanSat competition. The rockets were provided by the competition organizers. More investigation is needed as to why, but the CanSats have a high failure rate. The first year, all three entries failed. The next year, four out of the seven that launched failed. This year, 19 of the 27 teams are still left after the preliminary design review, but the launch will occur later this year.[55] Despite high failure rates, this program is a great way to get schools into satellite design in an extremely low-cost manner.

Risk management has not been an important focus for the teams doing CanSat project. It would be possible to reduce risk by implementing a simple risk management plan based on the options presented thus far. These projects do not need a very formal program, but some risk management could help their success rates and help students understand the risk management process.

3.5.4 Piggyback Experiments

Piggyback space experiments are small payloads that can fly out to operational orbits on a host satellite or be attached to a rocket upper stage, but they do not have to be jettisoned from the host carrier. These experiments could also fly inside the Space Shuttle or International Space Station. The SPHERES[56] (Synchronized Position Hold Engage and Reorient Experimental Satellites) project at MIT is a current example of this type of experiment because it only operates in the International Space Station. One of its goals is to reduce risk of the emerging technology of autonomous formation flight. Instead of applying a high degree of risk to multi-million dollar missions, SPHERES serves as an intermediate vehicle on which they can develop and test algorithms.

SPHERES is a highly involved and complex project, so the program has been expensive by small satellite standards. However, the project has kept cost and risk down by being a piggyback experiment and by designing satellites with user control. The cost expenditure for development, launch, and operations, over about a four year period, was approximately $2.5 million.[57] Other schools can use the piggyback method for lower cost given a less complex mission.

Despite the cost and the complexity of the mission, SPHERES does not discuss risk in a formalized manner. Failure modes are brought up and discussed when students encounter these issues, and most often fixes are made in the software. Hardware and software testing are performed both before launch as well as at the beginning of each test session on the Station. This method has proven to work for their project; only one problem was found after launch, and it was fixed with a software upload.

3.6 Advantages of Student-Run Satellite Programs

While alternate platforms are a good method to get students involved in hands-on space missions, there are numerous advantages to designing and building satellites through university programs. Students gain experience and are motivated by satellite engineering and from working with industry, and the schools benefit too from publicity and research funding.

While experience on other platforms is useful, it is not quite the same as working with a true satellite mission. Students are exposed to hardware, software, quality assurance, testing specified to the space environment, systems engineering, and much more. Not all of these tasks are present on missions based on other platforms. In a satellite project, students can work on detail-oriented technical work, systems engineering for subsystems, systems engineering for the whole project, program management, or some combination of all these tasks. This is very valuable experience that students are not likely to get otherwise. Therefore, to get the best understanding and to prepare the students as well as possible for a career in aerospace engineering, small satellites are the best option.

With student satellites, there are usually a smaller number of people working on the project (as compared to an industry program), so changes can be implemented quickly. These small groups also lead to shorter lines of communication, both formally and informally. Because of this, students learn a lot about the design decisions and processes of the other subsystems within the satellite, increasing their understanding of the system as a whole.

Satellite projects allow students to relate their education to a real hands-on satellite project, which can be an inspiration to students. Professors have another opportunity to teach satellite engineering in a manner that will relate to the student's interest – getting their satellite to space. The students are more motivated to learn the material while participating on a satellite that is really going to fly. Students are more likely participate and help the project succeed if they are working on an exciting mission.

Students are also able to work with companies specific to the many fields included in aerospace engineering. This way, students are able to learn more about real world engineering, both for technical and career-related reasons, since employers want to select new graduates with relevant work experience. Companies also like the ability to work with students, one reason for which is to gain prospects for future employees.

Satellite projects are also advantageous for the university. Satellite programs carry prestige, and schools can use their involvement in satellite design to increase the school's visibility. Universities will bring in more research money to the school with a satellite program, also bettering its reputation. Schools are flying more advanced payloads, so that they are working with payload providers and getting data back that can be used in publications and further studies, which benefits both the universities and the students. It is also possible that the school could find a niche in small satellite development and provide a service to the community, hence bringing in more money to the program. One example of this is at Cal Poly, where they provide a deployment mechanism, integration, acceptance testing, and shipment to the launch vehicle for

CubeSats. Surrey Satellite Technology, Ltd also started as a lab of University of Surrey and spun-off as a separate business.

Companies looking to do space research may soon want to work with universities. Student satellite projects are usually cheaper but more risky than their industry counterparts. As universities get more experience and put good management practices in place, student-run, small satellites will be an even more appealing option.

Whatever the platform, working with students has its advantages because of their determination, passion, and optimism. Students can also be more creative as well as more open to hearing new ideas and innovative solutions to problems. In the future, hopefully more businesses will be interested in partnering with students to create inexpensive and novel space missions.

Chapter 4: Failure Mode Analysis Options and the Master Logic Diagram (MLD)

Chapter 4 begins with a comparison of failure mode analysis options. The master logic diagram (MLD) will then be studied in depth to show how a general MLD can be developed to apply to any student-run small satellite program. Uses, restrictions, and benefits of the MLD are all discussed in detail.

4.1 Failure Mode Analysis Options

Small satellite programs that are run by students require guidance that ad hoc risk management approaches cannot provide. It is necessary to have a tool more applicable to these programs, which can help guide the users to identify failure modes. It would be most beneficial to have a way to better identify technical risks, but the program must also keep in mind that programmatic risks can cause technical failures as well because of the management, configuration control, lack of oversight, etc. To recognize technical risks, one method that might be suitable for a student team is a framework for risk identification that the students can then modify to suit their needs.

The method chosen for the framework should meet a number of guidelines in order to be most fitting for student use. First, the option should be one of the common ways that engineers in industry analyze failure modes. By using a tool that is common in the engineering world, students will be able to get experience that will be valuable to them in their jobs. In addition, students have more places to seek help if they are analyzing risk with a tool that the industry can help support.

Second, the method that students use should be easy to learn and consume as little time as possible to implement. It should also be a technique that can be generalized for small satellites but then be made to be applicable to a specific mission. Third, due to the lack of experience of students, the risk assessment method should help identify initiating events. Students can probably think of critical end states of the satellite, but determining root causes can be more difficult. Finally, the tool must also identify events in a systematic way. Otherwise, it is likely that students will miss certain risks affecting their design.

Many tools exist for the multiple stages of risk management; different techniques can be used to perform a combination of risk identification, assessment, control, or monitoring. Some common techniques include event tree analysis, fault tree analysis, probabilistic risk assessment (PRA), and master logic diagrams (MLD), and more. In this section, the different tools traditionally used in risk management will be discussed.

4.1.1 The Top Risk List/Stoplight Diagram

This method, discussed in Section 3.1.1, is easy for students to understand and create. It is also simple and inexpensive. However, this process may be too simple because there is no systematic way of identifying the entire set of risks. Risks could also be missed if the team identifies the top risks and then does not update the list as the project proceeds. Combining this type of list with a more systematic failure mode identification method could prove useful, though.

4.1.2 Event Tree Analysis

An event tree examines accident scenarios and looks at the consequences of those events. In other words, it is a technique that first identifies root causes of failures and then critical events that affect the progression of the initiating event into an unwanted end state. As the number of events increases, the diagram spreads out like the branches of a tree.

An event is a statement that can be true or false according to the current state of the system. For example, the event "Batteries provide backup to solar panel failure" is either true or false, regardless of whether the user knows this information. A qualitative diagram can be made to describe the potential outcomes of the initiating incident. The possible consequences of the primary event are considered, and each of those outcomes is analyzed to determine their potential results. This process is continued until the end states are reached. Figure 8 is an example of a very simplified event-tree for part of a power subsystem of a satellite.

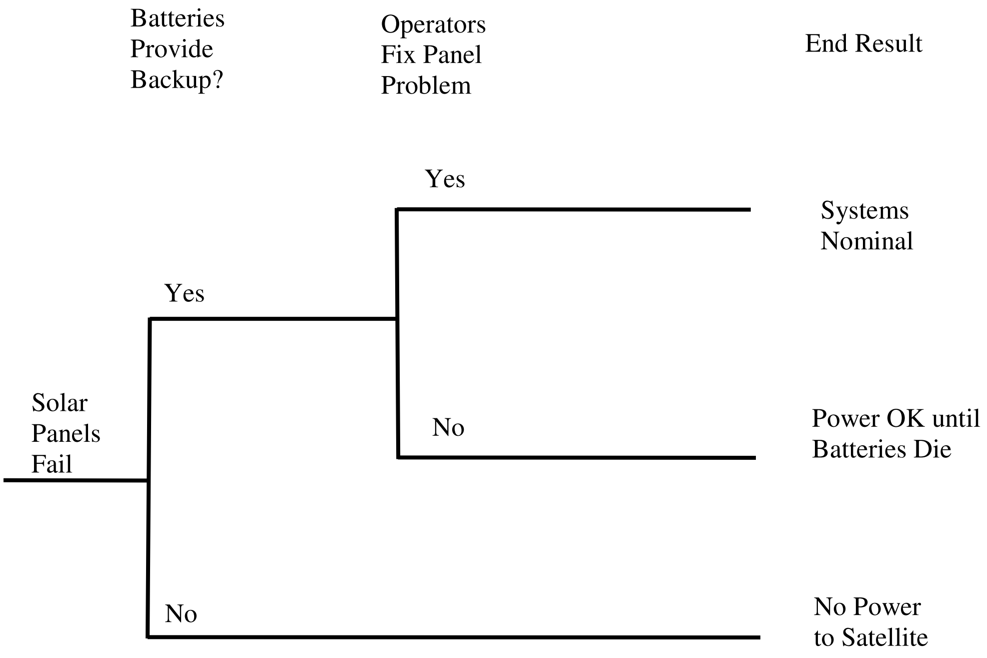

Batteries Provide Backup?
Operators Fix Panel Problem
End Result
Yes
Systems Nominal
Yes
Solar Panels Fail
No
Power OK until Batteries Die
No
No Power to Satellite

Figure 8. Example of an Event Tree Analysis for a Solar Panel Failure

As the reader advances from left to right in the diagram, the events are advancing in time as well. Once the initiating event has occurred, the next "barrier" is reached. The barrier, often a back-up system, is expected to work in the case that the initiating event occurs. Branching upward in the event-tree means that the event at that stage is true, and branching downward means the event failed. At the end of each path, we have reached an end state, which detail all the possible outcomes stemming from the initial event.

Event trees are useful because they take into account the dynamic nature of complex systems and can be easily updated as the design changes. They also identify risk drivers and end states of interest, which can be used in a probabilistic risk assessment (Section 4.1.5). Event trees lead to a large number of outcomes, and oftentimes only some of them are relevant to the engineers designing the system. Therefore, event trees can be inefficient and difficult to manage. One valuable check, though, is that the probability of all the events at the top of the tress has to equal one, which can be useful during analysis, as long as the probability equaling one is not artificially forced to be true.

4.1.3 Fault Tree and Master Logic Diagram Analysis

A fault-tree is a graphical technique for top-down analysis used to evaluate specific undesired events. In a fault-tree analysis, the engineer asks the question: How does it fail? This type of analysis works well when the user knows what failures they are afraid of and would like to examine how likely they are to occur. The tree uses deductive logic (and logic gates) to map component level failures to a system-wide failure.

Master logic diagrams (MLD) are similar to fault trees, but they are created at a higher level than for what fault trees are typically used. In addition, fault trees usually utilize formal probabilistic analysis, whereas MLDs typically do not.[58] Both master logic diagrams and fault trees, though, help identify the initiating events that can lead to critical accidents or mission failure.

MLDs and fault trees are started by identifying critical end states, and then failures leading to those causes are found. The top level identifies faults of the system, while the intermediate levels are subsystem failures, and the lower levels identify failure modes or causes, called initiating events. While these analyses can be done at different levels of detail, the diagrams are considered complete when breaking down a component leads to the same response as the next higher level.[38] In complex projects, an event tree is developed for each initiating event in the MLD.

The process of creating an MLD or fault tree is iterative. Once the risk manager or engineer begins developing scenarios, they should note similarities and difference in the system response. Through the observations, they can refine the diagram and its fault scenarios until a streamlined, coherent tree is formed.

Both master logic diagrams and fault trees use symbols to provide more information about the type of failure that is occurring. Table 6 and Table 7 show the type of gates and blocks used in MLDs and fault trees.[59] For an example of what MLDs and fault trees look like, see Figure 9.

Table 6. Gates Used in Master Logic Diagrams and Fault Trees

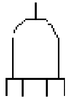
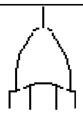
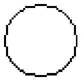
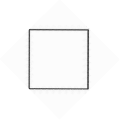
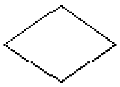
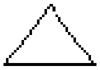
Gate Name	Classic Symbol	Description
AND		The output (top) event occurs if all input (bottom) events occur
OR		The output (top) event occurs if at least one input (bottom) event occurs

Table 7. Blocks Used in the Master Logic Diagrams and Fault Trees

Block Name	Classic Symbol	Description
Basic Event		An initiating failure or fault
Non-basic Failure		A failure or fault that is not classified as "Basic"
Undeveloped Event		An event that is not developed further
Transfer		A block that transfers the user to a sub tree

The basic MLD or fault tree analysis is a commonly used technique, so students can easily communicate with professionals about risk with this type of analysis. They are also less open to interpretation than an event tree because of its deductive nature. A useful aspect of creating fault trees is that the probability of the top event can be found by ascribing probabilities to each of the events below it. If these probabilities are known, then the system unavailability or the failure frequency can be found.

Some disadvantages of fault tree analyses are that they tend to require absolute failure modes, and they do not model sequence-dependent failures. Dynamic fault trees or Markov models are needed for dependent failures.

4.1.4 Event Sequence Diagram

An event sequence diagram (ESD) is a qualitative graphic representation of the sequence events leading up to the end state. The end state of the sequence may be a failure, a degraded state, or a command to move to the next sequence because there has been no failure. Scenarios are typically developed for each phase of the mission: deployment, cruise, reentry, etc. These diagrams use symbols to depict the flow of activity in the sequence, similar to the symbols in a fault tree.

ESDs are good for dynamic systems and can help identify end states of interest. However, they are normally used as part of a larger failure mode identification process and are not very useful by themselves. With a focus on operations, though, event sequence diagrams can also be used later in the mission to understand consequences of failures during the mission.

4.1.5 Probabilistic Risk Assessment

A Probabilistic Risk Assessment (PRA)[60] is a much more complex failure mode identification method than those discussed so far. It was developed for risk assessment, but it is also used throughout the design phase in trade-off studies to base decisions at least partly on the risk of each option.

A PRA aims to answer more than just what can go wrong. A PRA helps identify initiating events and possible outcomes, calculating the severity of the consequences. It also will determine how likely the failure is to occur, and with what frequency. Therefore, the PRA analyzes every possible failure mode both qualitatively and quantitatively at great detail.

Probabilistic Risk Assessments include many methods of risk analysis, making it time-consuming and laborious. PRAs use master logic diagrams to focus in on the most important initiating factors. Then engineering analysis is performed to study what could happen after the initiating event occurs. Likelihood is determined by inductive methods such as event tree analyses or event sequence diagrams or by the deductive fault tree analysis.

If the PRA is qualitative, the result of the assessment can be given as a two-dimensional matrix with consequences and their probabilities. If the PRA is quantitative, the result is a risk curve, which plots the frequency of exceeding a consequence value against the categories of consequences. While a PRA answers many questions relating to risk and safety, it understandably consumes time and personnel, and it requires substantial component performance data and insight, both of which can be hard to obtain. Therefore, the PRA is potentially not a good match for small projects.

4.1.6 Failure Modes and Effects Analysis

A Failure Modes and Effects Analysis (FMEA)[61] is a complex and even more involved way to identify failure modes, determine their effect on the operation of the system, and identify actions to mitigate the risks of failures. The FMEA works to solve each of these issues and can therefore be a time-consuming assessment as well. It also is not proficient at modeling common cause failures, which occurs when there are two failures that arise because of one initial failure. More information on the detailed procedures for FMEA can be seen in the reference.

4.1.7 Analysis of Options

The options presented thus far include the top risk list, event trees, fault trees, master logic diagrams, event sequence diagrams, probabilistic risk assessment, and failure modes and effects analysis. These are the most common ways to analyze failure modes in industry, so all of these tools meet this requirement that was set out in the beginning.

To decrease the amount of time spent on learning how to do risk management, the method chosen should be easy to learn and minimally time consuming when applying it to a specific mission. The PRA and FMEA do not meet either of these needs. Fault trees are usually very detailed and use probabilistic information, making them difficult for a project to tailor the method to their need.

The risk assessment method should help the student identify root causes. The event tree begins by having the engineer identify initiating events, and the event sequence diagram uses the operations plan of the mission to find end states, neither of which is best for student use.

The tool must also help the engineer systematically identify all appropriate risks. This rules out the stoplight diagram since a systematic method for identifying risk is not used.

Therefore, the option that meets all of these needs is the master logic diagram. Master logic diagrams are not as complex as an option like a PRA, but they are more advanced than a simple Top Risk List. Like fault trees, they are widely used in industry, and if made correctly, they will not take long to apply to a specific mission. In addition, MLDs do not require probabilistic data, and they identify initiating events in a methodical and thorough manner, making them a good match for this application to student satellite programs. Restrictions of the MLD will be discussed in Section 4.3.5.

4.2 Development of a Master Logic Diagram for Small Satellites

A master logic diagram has been identified as a good framework to outline failures relating to small satellites in order to identify pertinent risks. To create the MLD, major end states must be identified first. Any satellite program has a few end states that would be catastrophic to the program. These include:

1 - Catastrophic failure at or near launch (no data, no proof of function)
2 - No response from either the bus or the payload on orbit, but otherwise potentially well functioning

The result of either of these events happening is that the satellite mission is basically over. There are measures that can be tried on orbit to recover the satellite in certain circumstances, but if those do not work, the mission has failed. While the end state is the same, the causes of these two types of failures can be very different, but they are equally important.

Another important consideration in determining the severity of the second end state is when the failure occurs. The success of the mission greatly depends on how much data is returned before the failure occurs. Obviously each of these scenarios offers different amounts of engineering or scientific output for the team. Considering that the second end state can still be catastrophic if it occurs before the design life has ended, the MLD will consider both of these end states.

The next issue to consider is how to make the MLD general enough for any small satellite program. To accomplish this, the master logic diagram has to be broad to be applicable to all small programs, but it also needs to be specific enough to adequately help identify failure modes. This balance is hard to achieve, but it was accomplished through careful creation and review.

To create the MLD, the high level failures relating to the failed end states were laid out to cover all types of small satellites as well as their interfaces. At each branch point, observable failures were identified in order to fill out the MLD in a complete and user-friendly way. For example, for the end state "No data from the satellite," one of the observable reasons for this might be that there is a problem with the actual satellite because there is no signal transfer. The reasons for the lack of a signal were outlined at the next level, and this process was continued, expanding the tree further.

As the initiating events were identified to be subsystems, such as the thermal or power system, the text Space Mission Analysis and Design[47] was used to get a basic yet comprehensive overview of the components in a satellite. Since this reference focuses on general satellite engineering, it helped to create a universal MLD that covers all satellite designs. Detailed investigations were made into each subsystem to determine what components small satellites may use and how they interface with other subsystems. In addition, an MLD for programmatic risks was laid out as a stand-alone diagram. The MLD was then reviewed at MIT to ensure that the layout was consistent and the identification of initiating events was correct. In the end, 17 pages of technical risks and one page of programmatic risks were documented. Figure 9 shows part of a tree for a top-level failure of no data from the satellite.

This method for creating the MLD is very useful for two reasons: identifying failures that will cause a bad end state during design and problem solving on-orbit. While designing the satellites, students can trace failure modes through the system and take these failures into account when making design, hardware, and software choices. When something goes wrong after launch, given the telemetry that university satellites often receive, the students can use the MLD to investigate what could have caused the failure in this way. All the student satellite failures presented in Section 2.4.2 can be found in the MLD by tracing failures through the diagram. Therefore, this MLD template is versatile and useful at any stage in the mission.

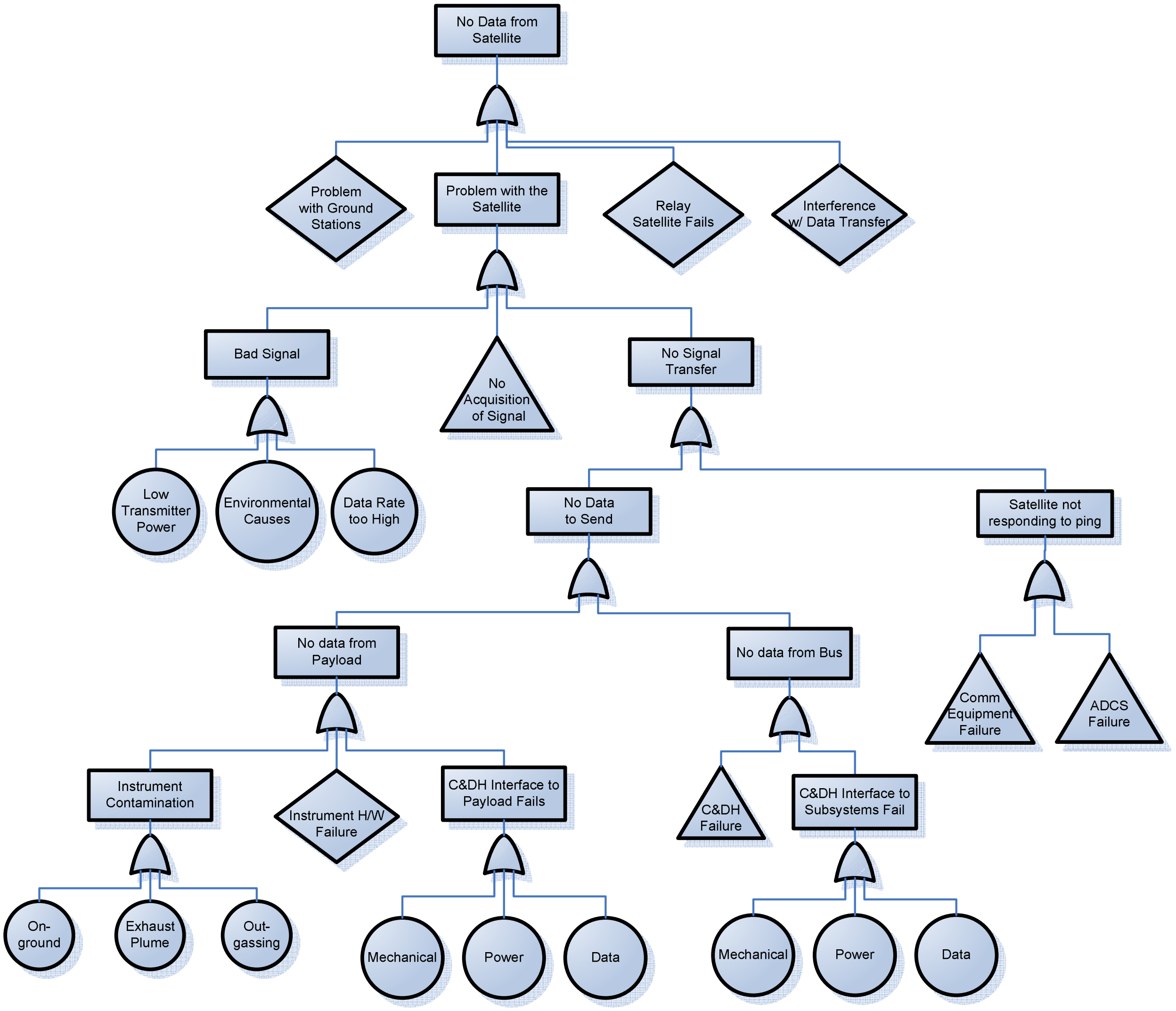


Figure 9. Section of a Master Logic Diagram for Small Satellites

4.3 Uses of the Master Logic Diagram

The master logic diagram is a useful tool at the start of a project, or it can be applied to a design if the team has already begun working on the satellite. The MLD can be used for troubleshooting on-orbit, and it can also be used to help teach students about satellite engineering in general.

4.3.1 Beginning a Design

This risk template can be used in many ways. When beginning a design, students can learn about the satellite as a whole while learning about the types of risks associated with each subsystem and their interfaces. Students can also identify where most of the risk falls in order to plan their resources accordingly.

The MLD helps give students a broad picture of the satellite, including the types of interactions subsystems have and the possible ways a satellite might fail. Because of these interactions between subsystems, the MLD can show how design decisions made for one subsystem can affect other parts of the satellite. Knowledge of the entire system gives context to its technical risks. Without an all-encompassing view of the risks of a program, the students lack sufficient comprehension of the risks and how to mitigate them.

Interfaces are often poorly specified, and this diagram can help to demonstrate the type and number of interfaces that exist. The MLD can prepare students for what could go wrong in the satellite, eliminating some of the surprise of a malfunction and helping the students because they have been exposed to the idea of the failure.

By looking at where the potential greatest risks are for a certain mission, students and faculty are more aware of the project's needs. Students can learn about specific subsystems and their failure modes by studying the relevant portions of the risk template. This is helpful for subsystem engineers in order to better understand their assignment as well as to recognize what is risky about their subsystem. It is also helpful to systems engineers, who need to allocate resources such as time, money, and personnel.

4.3.2 Working with a Design

The main goal of creating the MLD is to help with risk management when working with a satellite that is being designed. Since the students lack experience in designing satellites and identifying risks, they can apply this framework to their project to identify failure modes in their program. This method is intended to help give guidelines to students and to help them brainstorm risk modes, but they will still need to tailor the MLD to their own project. The MLD is universal and applicable to nearly every student-run small satellite project, so the students then can choose what parts of the diagram are needed based on the design of their satellite; the result is an MLD for their project.

In addition, students can use the MLD to identify single point failures, which many programs try to minimize. With this template, the teams can discuss what single point failures exist early on in the project's life. Hopefully, by identifying problems early, changes can be made to reduce these failure modes without drastically affecting the design or without costing the program time and money. While all students can use the MLD in some way, one person (or a small team) should be dedicated to maintaining and updating the MLD as well as tracking any changes to the satellite that result from the use of the MLD.

4.3.3 In the Classroom

The MLD can also be used as a teaching tool in the classroom setting, much in the same way as students use the MLD at the beginning of a satellite design. The class can learn the types of failures satellite face in addition to learning about graphical fault trees and how these diagrams work. By learning about these methods of risk identification, students will be able to better understand how to logically break down a failure into its possible causes and will learn a way to diagnose failures in a system. These skills can extend beyond satellite engineering, but the fundamentals can be taught with the MLD. As mentioned previously, the MLD can help students to better understand how the satellite fits together as a whole and how subsystems interact.

4.3.4 Troubleshooting a Design

Whether during testing or on-orbit, the MLD can be used to troubleshoot a problem. As mentioned before, when a component fails, the engineer tries to find root causes for that failure. The MLD helps with this process in many cases because of the way it's laid out, with upper level failures that the team might notice, and then lower level failures branching out from there. Solving a problem that occurs in a test or a failure after launch would be easier, faster, and more thorough if the MLD is used.

4.3.5 Restrictions of the MLD

Using the MLD can help student teams to identify failure modes, but there are some caveats to using the MLD. First, there are some areas that the MLD does not cover. For example, it might be necessary to add items to the MLD for certain projects for complex subsystems or missions that are testing a new technology. Or, a team may wish to make the MLD into more of a fault tree, outlining failures down to extreme detail. Adding items to the MLD is best done by the senior members on the project or during a team review. The high risk of a catastrophic launch vehicle failure does not appear in the MLD in an amount proportional to how often this failure occurs in the real world. Schools must remember that launch vehicle failure is a very prominent threat and should plan accordingly.

Some design risks are included, but operational risks are not included everywhere in the MLD because there are just too many variations. For example, the team can decide to put solar panels on all sides or increase battery capability in case the first orientation of the solar panels fails. These types of design decisions are everywhere in an engineering project and cannot be accounted for in a set framework such as this. The MLD does include under-design and component failures as risks, and given the purpose of the MLD, this is suitable. One other area not included in the MLD is organizational and programmatic risks. Those sorts of risks can cause a failure at any point, and a team must just be aware of those reasons for failure even though they are not in the MLD. Most failure mode analysis options cannot include these types of failures, so the MLD is not inadequate for the job; however, the teams must just keep this in mind when designing their satellite using any failure mode analysis option, including the MLD.

Other areas of risk not apparent in the MLD are technology development risk and human error in design. Technology development risk is included in component failures since every part can fail due to not having enough maturity in the design. The engineers just must be aware of the ways in which a part can fail, and its development level is a part of that list. The latter risk is hard to incorporate into any failure mode analysis program, just like design, operational, or programmatic risks. All of these types of failures must be considered by the team, but they are not included in the MLD.

Second, the MLD only fulfills the first two steps of risk management – understanding risk in relation to a project and risk identification. There are still three more steps – analyzing the impact of the failure modes, implementing a mitigation strategy, and tracking and updating risks. The schools do have to create an overarching risk mitigation plan and implement that in conjunction with the MLD, but the MLD provides the necessary start. It may be that starting a risk program is difficult for student teams, and that's why not many schools have risk strategies, so having the MLD focus on the first two steps is helpful. Schools can choose how to implement the last three steps of the risk management process to best suit their needs. It is imperative that the program management or systems engineering team not only use the MLD, but also decide how to incorporate the MLD into a larger risk management program. Some suggestions can be found in Section 3.1, and the process is detailed in Section 5.4.

Third, even though the MLD is high-level and will be scaled down to apply just to the specific project, it is possible that an MLD may be overwhelming to non-experienced students because of its size. Given this template, students may attempt to fix as many of the risks as they can. This could waste time and resources needed elsewhere on the project. However, if managed correctly, this weakness could be avoided, and the MLD could be used as intended.

4.4 Benefits in Risk Mitigation Using the MLD

This MLD framework allows engineers to reduce much of the risk associated with university-run satellite programs. While not all the risks detailed in previous sections will be mitigated, the MLD does reduce risks associated with programmatic differences, funding and schedule, experience, and documentation.

4.4.1 Programmatic Difference Benefits

Universities have fewer resources to dedicate to all the tasks needed to create a satellite, and risk management is a task that often gets pushed aside. The MLD begins the process of risk management, giving the programs a way to identify failure modes of the satellite with little experience or effort on the part of the school. However, the MLD is still able to teach about failure modes and make students more aware of risks since the entire template is laid out for them.

Since many universities have informal risk management programs, the MLD is useful because it provides a structured risk identification format. Due to the method for creating the MLD, it can identify all the risks related to a critical end state in a systematic way. This makes the diagram beneficial for students who often don’t fully understand all the components of a satellite yet. In essence, universities have a programmatic risk of risk management failure, and the MLD helps to reduce this programmatic risk.

4.4.2 Funding and Schedule Benefits

While the MLD template does not directly help a project receive funding, risk management is a necessary part of a program review. When competing for funding, whether against other universities or not, a program with a clear risk management plan will portray a more advanced and well-managed program, increasing its chances of funding. When university programs are applying for funding, the MLD will help to fulfill the need for identifying risks.

Funding affects nearly every aspect of the design, and when programs are deciding where to allocate funding, more effort should be focused on risky components than other less-risky systems. By using the MLD, high-risk areas can be identified and given more resources in terms of funding and schedule. If done early on in the program, resources can be allocated more wisely, saving the team money and helping to keep the project on schedule.

4.4.3 Experience-related Benefits

Many of the benefits relating to the lack of experience of students were discussed in Section 4.4, where it is explained that the MLD helps students identify failure modes and understand subsystem interactions better. Students can better make observations about risk on their subsystems if they have a template from which to work. The MLD also provides a bigger picture of the risks facing the satellite, which decreases the students' learning curve and increases their knowledge of the entire system. With this broad view of all the failure modes, the students can better determine how risk mitigation techniques affect the whole project and their subsystems.

4.4.4 Documentation Benefits

Documentation is a fundamental part of transferring knowledge from one team to the next. Using an MLD such as this provides a better way to document risk so that communicating failure modes to future members is easier. Turnover is a large problem in student organizations, and having a template in which to document risks can help transfer knowledge because it will take less time to record the information. By providing a template for the identification of risks, the reporting of risks will have a consistent format across the program. The MLD also gives a central place to store information so that information on risks is not lost within a team. In addition, if schools want to share information, lessons learned, current risks to subsystems, etc., they can do so more easily with a consistent layout.

Chapter 5: Application of the MLD to the Mars Gravity Biosatellite

Chapter 5 details the application of the MLD to the Mars Gravity Biosatellite (MGB). The background of the Mars Gravity project is discussed, including its attempts at risk management. Three experiments to measure the value of the MLD are presented, and the five step risk management process is outlined again, this time focused on student projects. Finally, the outcomes and benefits of utilizing the MLD on the Mars Gravity Biosatellite are detailed.

5.1 The Mars Gravity Biosatellite

5.1.1 Program Overview

The Mars Gravity Biosatellite is a project run jointly between the Massachusetts Institute of Technology (MIT) and the Georgia Institute of Technology. Its goal is to build a satellite system that will send 15 mice into low earth orbit, spin the spacecraft to simulate the gravity of Mars, and then bring the mice back to study the effects of partial gravity on mammalian physiology. This research satellite will help us to understand current medical uncertainties regarding long-term spaceflight, helping engineers to better design systems to keep astronauts healthy during their voyages to the Moon and Mars.

The Mars Gravity Biosatellite program was started in August of 2001, and over 500 students have been involved to date. The team consists of approximately 30 people each semester, with the majority being undergraduates led by about five graduate students. The program manager is involved in the project part-time, so the effort is mostly student-led. The team gets ad hoc support from industry throughout the design process, and there is more formal participation of industry personnel at reviews. The project is funded mostly on grants through NASA, the National Space Biomedical Research Institute, and the participating schools. Since the program's inception, it has raised approximately 1.3 million dollars in direct funding and $300,000 as in-kind donations, but this does not nearly meet the needs of the program to reach launch.[62]

The Mars Gravity Biosatellite consists of three main sections: the Payload, the Entry, Descent, and Landing (EDL), and the Bus. The Payload provides life support to the mice and collects scientific data for the investigators. The EDL helps bring the Payload back safely through the Earth's atmosphere to the ground. The EDL and Payload together form the Reentry Vehicle. The purpose of the Bus is to help the satellite stay in orbit and function off the Bus' power, communication, cooling, and other support services. In addition to the satellite itself, the system also requires a launch vehicle, a ground station network, and ground support equipment. With a mass of approximately 500 kilograms, this satellite is much larger than normal student missions and just fits within the range of satellites that is commonly classified as small (0-500 kg).[4]

No major technical setbacks have occurred on Mars Gravity, but the program has had its share of programmatic risks over its lifetime. The satellite was supposed to launch in 2005, but primarily due to funding, the schedule slipped considerably. Another factor in the launch delay was the underestimation of time that it takes for inexperienced students to design and build such a complex satellite.

The turnover rate of students is also extremely high (usually at least 50% per semester), and one reason may be that the demands of class work are too much for the undergraduates to balance research as well as class. The high turnover rate means that the graduate student managers must be involved in recruitment and training every semester, which uses up a lot of time. Finally, documentation is a serious problem with the Mars Gravity program. With busy students and high turnover, adequately documenting work sometimes does not happen. Efforts to better this situation, such as configuration control and end of semester reports and presentations, have helped this problem some.

As can be seen from its challenging mission and program issues, Mars Gravity is a complex student mission, but the focus on student education and the lack of resources is the same as other university projects. Therefore, this satellite is considered to be in the same class as the other small, student satellite missions discussed so far.

5.1.2 Previous Risk Management on Mars Gravity

Throughout its five years in existence, the Mars Gravity program has had very little focus on risk management. It is difficult to ascertain whether the lack of risk management has caused programmatic or technical failures, but it has certainly not helped.

From early on in the program, high-level science and engineering requirements were specified. The only way that risk was tracked was through these requirements. The requirements were tracked in an Excel spreadsheet and were tied to four items – priority, risk, maturity, and status. The intent of this list was to make sure that the requirements were focused on in a way that kept the students thinking broadly about the systems engineering issues at hand. While only one of these categories is actually labeled "risk," all of them do deal with risk management, as identified in the process outlined in Section 2.1.1. This system did not work well because it was not maintained nor was it a complete risk assessment. By using the requirements, the student team got a good look at some of the risks, but identifying risks in this manner does not cover all the possible risks. Also, of the requirements identified, not all of them were assigned risk levels. In general, this plan was not thorough, and the Mars Gravity team lacked the systems engineering focus to put a better system in place.

Another risk management method attempted was to use a descope plan. This kind of plan does not cover the actual risk management steps, but it does help the team think about other options for the project if something goes wrong, reducing some of the programmatic risks discussed earlier. A descope plans was started for MGB, but it was not kept up to date because of a lack of strong systems engineering focus on the team at the time.

5.2 Why the MLD was Applied to Mars Gravity

The Mars Gravity Biosatellite was in need of a consistent and easy-to-follow risk management program. It has been seen, in the MGB program, that students are young and inexperienced, and they have trouble maintaining documentation with such a high rate of turnover. It was also a problem in the past that the systems engineering team could not adequately manage risks, and these plans often were neglected. As shown in Section 4.1, a master logic diagram is a good match for university groups in order to help the students identify failure modes in the program in a consistent and well-documented manner. In addition, it is necessary to test the usefulness of the MLD before suggesting it as a tool to the small satellite community. Therefore, it was decided to apply the MLD to Mars Gravity in order to see the utility of the master logic diagram tool.

The two end states presented in Section 4.2 apply to Mars Gravity. These states are "Catastrophic failure at or near launch" and "No response on orbit but otherwise potentially well functioning." On Mars Gravity, science output is roughly seen as[62]:

- 24% in-flight collected data
- 60% post-flight dissection and analysis
- 9% post-flight behavioral/function (i.e. live) analysis
- 7% post-flight specimen recovery from waste collection module

As one can see, especially on this science focused mission, keeping the payload in good condition and getting back data is key to even partial mission success. To have no response from the satellite would lead to complete science data loss and potential reentry problems because of the lack of communication. For the mission even to reach half of its goals, the payload must be returned, meaning that either of the two previous end states would be disastrous for this mission. Therefore, the general MLD will be useful for the Mars Gravity Biosatellite.

One other catastrophic failure would be to have the satellite hit something outside of its landing ellipse once it has reentered. Since Mars Gravity is returning the payload to Earth, a number of failure modes that are not standard for university missions are introduced. These types of risks are not captured in the MLD that was created for the end states above; therefore, a failure of hitting something outside of the landing ellipse would require a smaller, different MLD. Since this type of failure is so specific to Mars Gravity, it will not be discussed further.

Despite the few pitfalls of the MLD, it was applied to the Mars Gravity project to help understand the context of the risks in the project and the possible failure modes. To do this, only those sections that apply to the Mars Gravity project were kept in the framework. In this manner, the MLD was used to start the risk management process for Mars Gravity, and its incorporation into the rest of the five step process is discussed in Section 5.4. Before proceeding with the rest of the risk management process, the next section shows how the MLD proved to be good at identifying both the types and the number of failure modes.

5.3 Comparison of Identification of Risks Using the MLD and Other Common Methods

To see whether the MLD is better than other risk identification techniques, two tests were conducted. The first was to compare brainstorming risks with using the MLD. The second was to compare how well students could identify risk with little to no proper training. To do both of these tests, the focus was on a power system failure because power system failures are a common occurrence in many mechanical systems. This gave the subjects in the tests a more equal basis for being able to identify risks since students across multiple majors and levels of degrees were used in the experiments. Then, the MLD will be analyzed to show that it can be useful for identifying the number of failures in each subsystem once the satellite is on-orbit.

5.3.1 Brainstorming vs. the MLD

In the first experiment, the result of a brainstorming session was compared to the MLD for the power subsystem. A brainstorming activity to identify reasons that there might be a power problem on the satellite was done by a graduate student member of the systems engineering team that was responsible for risk identification. This risk assessment was done without proper training or a framework for risk identification. No resources were used for this brainstorming diagram. Figure 10 shows the risks associated with no power in a preliminary brainstorm for the Mars Gravity Biosatellite, where no formal technique was used to identify risks.

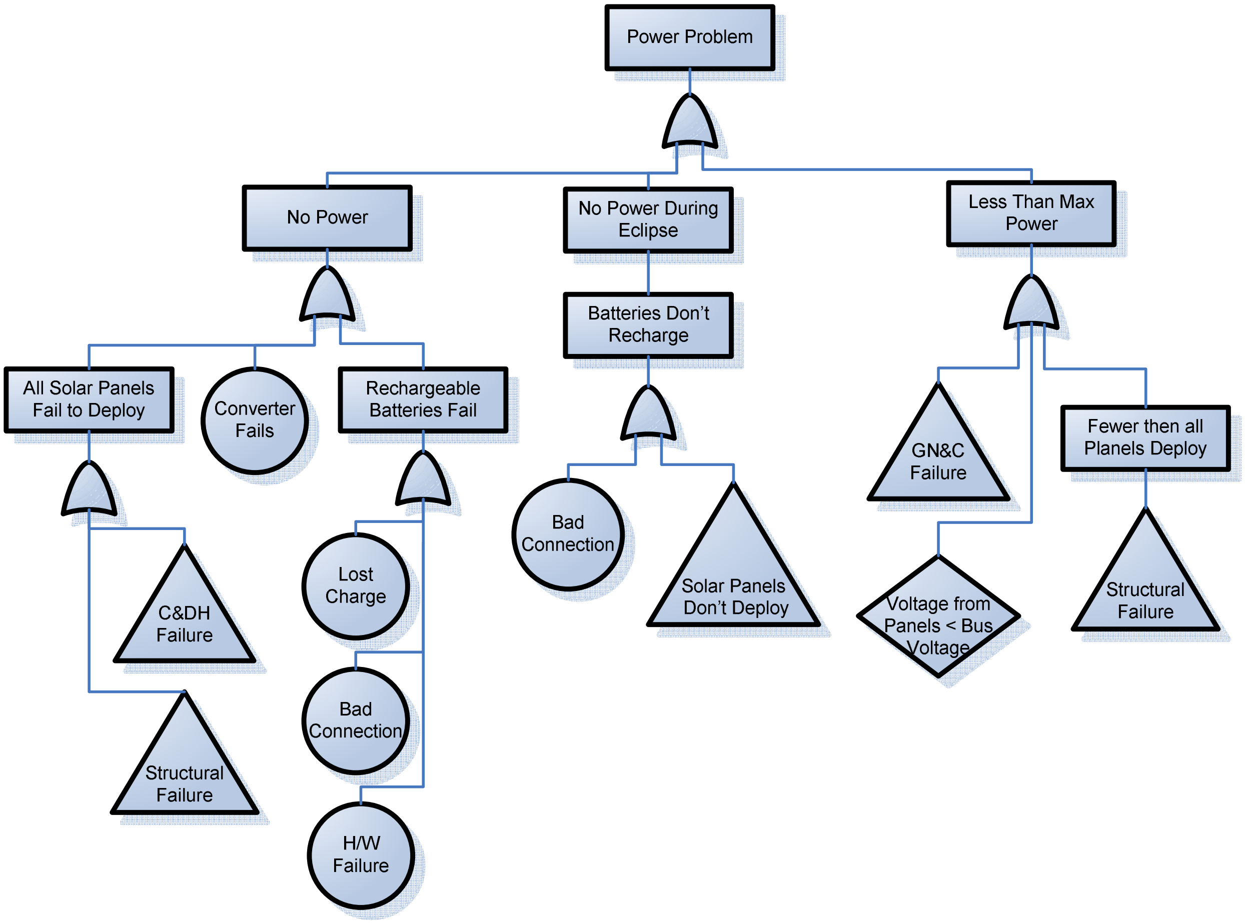


Figure 10. Preliminary Risk Brainstorm for Power Failures

It can be seen that there is a varying level of detail in the types and number of components identified. For example, the converter is included as an initiating failure, but items such as the regulator or distribution system are not. Mechanical hardware is not really considered in this brainstorm, and some interfaces are missing, such as the thermal issues, which might prevent components from working because the temperature is out of the appropriate range. In short, this brainstorm of power failures is missing many key failure modes.

Next, the MLD was applied to the Mars Gravity project in the area of power to show how part of the MLD would be applied for a real application and to compare the results to the brainstorming activity. Figure 11 displays the final MLD for a power failure on the Mars Gravity satellite.

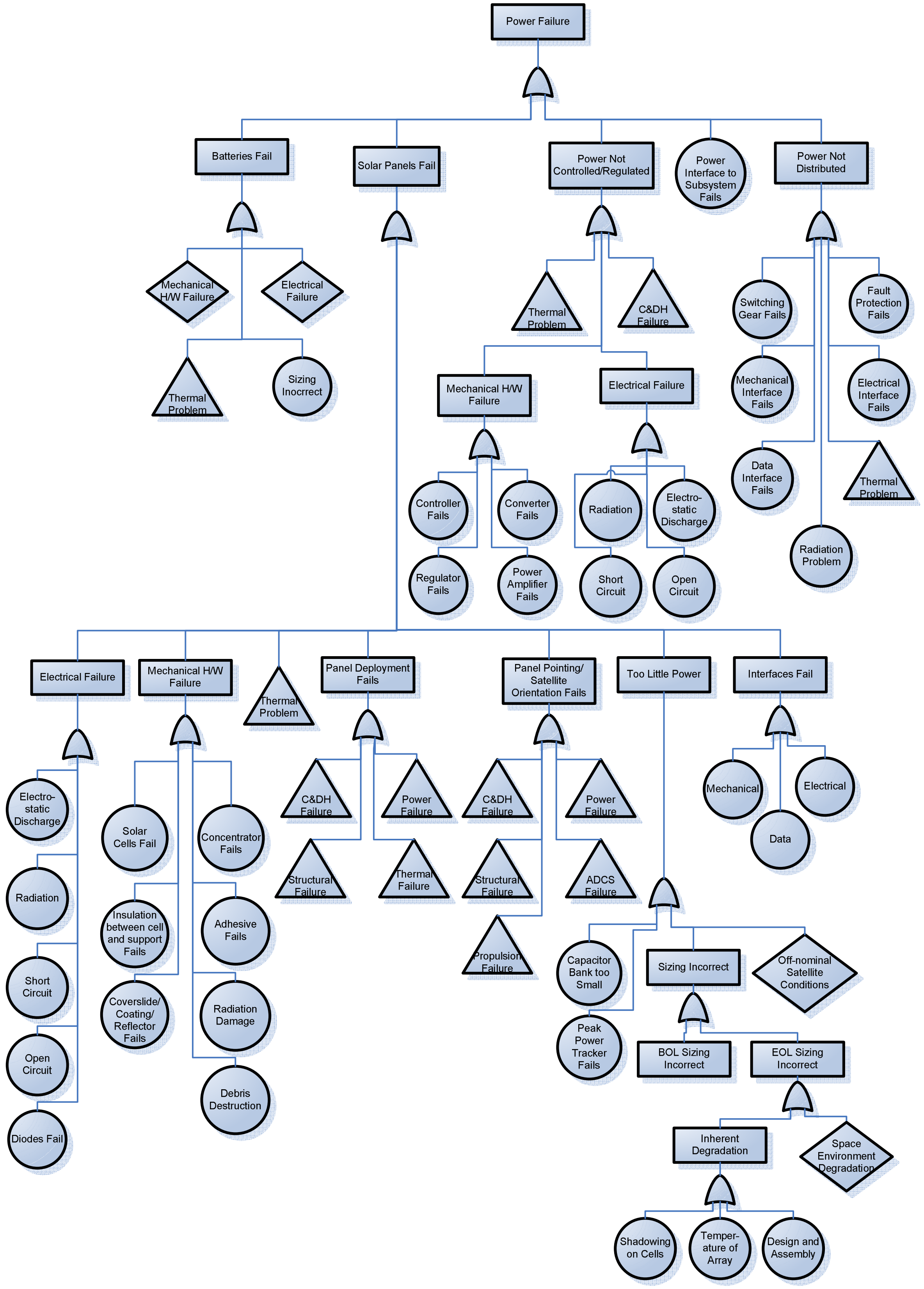


Figure 11. "No Power" branch of the MLD for the Mars Gravity Biosatellite

Figure 11 only shows a few levels of detail, but it aims to be consistent about the level of detail it includes. Additions could easily be made to make the MLD more in depth, if the project desires. For example, under degradation due to the space environment, additional events could be added for materials, lubricant, coating, thermal expansion, and space debris. An MLD is supposed to be a high-level fault tree, so a program can determine how much detail is needed. With the MLD in general, though, students are better aware of the specific threats within more general categories.

To see the utility of the MLD, one can compare Figure 10 and Figure 11. The master logic diagram returns considerably more data on the risks relevant to the satellite. It is better able to capture a complete set of risks over the entire satellite by providing a consistent and comprehensive framework for identifying risks. This framework also follows a logical path from the end state to the initiating events.

Conversely, a simple brainstorming activity, such as in Figure 10, has less utility because there is no unifying factor, such as an program-level detrimental end state, motivating the assessment. It also might be incomplete and biased toward the experiences of the people identifying the risks.

As with the identification of risks in any technique, the risks and their mitigation strategies will have an effect on the design of the satellite. The MLD ensures that risks over the entire satellite are noted, which is important information to have when considering design choices that affect an entire satellite.

These figures show that the MLD is able to better capture a complete set of risks for a satellite project than a brainstorming technique can. In general, this is true of many current university practices because none of them provide a consistent method for identifying risks, while the MLD does. Thus far, it seems that a master logic diagram provides a good framework with which to work.

5.3.2 Risk Identification Experiment

To confirm whether the MLD is able to help undergraduate students identify failure modes, an experiment was carried out at MIT. Thirteen mechanical, operations research, and aerospace engineering students were asked to create an MLD for the failure modes that might be associated with there being no power on the satellite. They were given an example for the communication subsystem to help guide them on the level of detail desired. The students were allowed to use any resources except the MLD, and they were asked to return the experiment within a few days.

To see how well the students could identify risks, the results from the students were compared with the MLD section shown in Figure 11, which has 56 failure modes identified for the end state of "No Power." When comparing the student work and the MLD, the student was counted as identifying a failure mode correctly when their lowest level of failure mode in a chain also occurred for the same reason in the MLD, even if their level of detail was not the same as the MLD. For example, if the student specified that an interface could fail, but in reality you have three different interfaces (mechanical, data, and electrical), the student was given one point. If they correctly identified all three interfaces, then they got three points. An incorrect identification of a failure mode resulted in a point in the "Incorrect" column. This occurred when the student flowed events incorrectly, gave the wrong reason for failure, or included parts not used on satellites (e.g. turbine blades). If a student identified the same risk more than once, that was ok, but if they only included the risk twice because the organization was poor, they did not receive points for the second time they identified the same risk. Table 8 shows the results when the students performed this risk identification task.

Table 8. Student Experience & Number of Failure Modes Identified for "No Power"

Student Schooling	Student Experience	# Correct	# Incorrect
1. Undergrad Freshman	Undecided	8	1
2. Undergrad Freshman	Undecided	18	1
3. Undergrad Freshman	Aerospace Eng major	14	1
4. Undergrad Freshman	Aerospace Eng major	19	0
5. Undergrad Junior	Mech Eng major, 1 summer on satellite project	14	0
6. Undergrad Senior	Mech Eng major, 1 summer on satellite project	7	0
7. Bachelor's Degree	BS in Operations Engineering	10	0
8. Master's Student	BS in Mechanical Eng, MS in Aerospace Engineering (in progress)	9	0
9. Master's Student	BS in Aerospace Engineering, MS in Aerospace Engineering (in progress)	6	2
10. Master's Student	BS in Aerospace Engineering, MS in Aerospace Engineering (in progress)	17	2
11. Master's Student	BS in Mechanical Eng, MS in Aerospace Engineering (in progress), 1 year on satellite project	23	0
12. Master's Student	BS in Mechanical Eng, 1 year on project	29	0
13. PhD Student	BS and MS in Mechanical Eng	8	0

It should be noted first that neither age nor experience seem to help identify failure modes. This is counterintuitive, but it is probably due to the small sample size of students that participated in the study. This result is not as important, though, as how the students responded and what their strengths and weaknesses were. Some of the students showed good insight into technical problems and the design of a satellite, but this was not true for the majority. In addition to the fact that students could not identify many of the failure modes, they also fell into a number of pitfalls.

While Student 1 focused solely on the solar panels not working, he was able to identify other parts of the subsystem as being a cause of the failure, including regulation and distribution failures. He was also able to see that the problem might be recursive. Students 1 and 2 understood that the power subsystem connects to other subsystems, but they did not know what those subsystems were. These two students showed a characteristic of beginning to understand the complexity of the system, but they could not yet see the big picture.

The third student not only identified major components of the power subsystem, but he was also able to identify other aspects of the power management system as well. His knowledge of the entire satellite was lacking, so his ability to identify failure modes in the power subsystem could have been from previous experience and not from general satellite design knowledge. This student demonstrated another common student problem – organization. The students were given an example of an MLD and told how to go through the process, but some students had a lot of repetition and were not well organized.

Student 5 was an undergraduate that had worked for a summer on the power subsystem of a satellite project. He identified most of the general failure methods but did not go into enough detail (even though the level of detail was shown in the instructions). It was expected that this student would have been able to identify many of the failure modes because of his previous experience with the power system, but it was surprising that he did not. This could have potentially been either because he did not have enough time or enough knowledge.

Student 6 did not include a number of major components, including the solar panels, and he only identified a few of the smaller components (such as converters, but not regulators). This student's experience was in the thermal subsystem, and most of the failure modes that he identified related to thermal issues, which indicates that the student was biased toward his experience and was unable to identify the breadth of failure modes. Finally, the list of failure modes had very little organization.

Student 7 did not have much experience with space systems, although he does follow its major news, and he admitted to knowing little about the internal workings of the power subsystem. However, this person is familiar with cars and performed his failure mode identification using cars as a reference, which helped to at least identify some of the failure modes. The major problem in this case was that he did not of know the components that comprise the power subsystem nor of the resources in which to find this information. He did, though, logically step through many of the potential power options and some of the subsystems they were connected to.

Student 8 was the first student to not identify any of the interfaces to other subsystems. Most students do see some connections and include those, but it is also likely that some students are not able to identify connections to other parts of the satellite at all. Student 9 has extensive experience in the airline industry, and his failure mode identifications were highly skewed toward that knowledge. In this case, he did not seek further information on satellites and created his failure mode tree with incorrect knowledge, resulting in a nearly useless list.

Student 11 was able to identify many of the failure modes and had good grasp on how the power subsystem can fail due to other subsystems. However, he was missing a number of the components of the power subsystem. His diagram did have decent organization and a sensible hierarchy. Jumping ahead to Student 13, he was unable to identify many of the components of satellite's power system, but this might been because of his background or perhaps a hurried attempt to complete the form, even though the students were given a few days to fill out their fault tree.

Most likely due to his education and experience with satellite projects, Student 12 was able to identify many more of the failure modes of the satellite. There were no major errors, omissions, or biases in the identification of power failures. In a couple of places, the student missed a few risks or did not go into enough detail, and that is why he wasn't able to get the full set of risks.

While this experiment had a small pool of subjects, it still shows a number of interesting trends for when students try to identify failure modes. First, students need a structure to help them organize their thoughts on a complex system. Lack of experience with satellites combined with inconsistent organization led to a hodgepodge identification of failure modes. Second, if the students did have experience in a field outside aerospace engineering, it biased their results when asked to identify failure modes for a satellite's power system. Sometimes this knowledge helped them to identify any failure modes at all, but other times it hindered their thinking about satellite missions.

Third, students could not fully identify interfaces to other parts of the satellite. Oftentimes, they understood the mission's complexity, but they were not able to list out all of those relations. This is a major problem because many failures come from other subsystems, so those interfaces must be identified. Lastly, it's not surprising that students couldn't identify all of the components of a power system. Many of the students hadn't studied this subsystem, and some of the students only had in classes. However, in all cases, the students could have used any resources for identifying failure modes, but they chose not to or did not know where to find the information.

While students in satellite projects will have more time and potentially more subject-area knowledge compared with the subjects in this study, this experiment still shows that an MLD can be a useful tool in identifying failure modes. While the Mars Gravity Biosatellite has not yet flown, these two case studies prove that a general yet adaptable tool for failure mode identification, such as a master logic diagram, is needed for student-run satellite programs.

5.3.3 MLD Comparison with Launched Satellite Failures

The MLD has proven to be useful for recognizing the types of failures in student-satellite missions, but it may also be useful for identifying the number of failures in each subsystem once the satellite is on-orbit. To see if the probability of a subsystem failing manifested itself in the MLD, the number of failure modes per subsystem was compared to the average failure mode rates of satellites that have flown. The data for launched satellites comes from Section 2.4.4. In the MLD, each subsystem's fault trees were investigated, and the number of failure modes was counted. To better compare the theoretical master logic diagram with actual space missions, only failure modes were counted that were not redundant. For example, if five different types of insulation were listed, only "insulation" was counted in the number of failure modes. It was assumed that student satellites would not use a lot of redundant parts and that not all types of, for example, ACS equipment were needed, so a sample set was chosen. Therefore, it can be assumed that the number of failure modes in each system is also an estimated average. Table 9 shows the results of this comparison.

In Table 9, the third line shows the sum of the ACS and GN&C failures because the majority of the systems that have flown record GN&C, ACS, and related software failures all together. The sum of the GN&C and ACS failures is not included in the calculations of the percentages of subsystem failures in the MLD since they are already included independently, so that column does not add up to 100%.

Table 9. Percentage of Failure Modes per Subsystem in the MLD Compared to the Percentage of Failures per Subsystem for Launched Satellites[3]

Subsystem	# Failure Modes Per Subsystem in MLD	Subsystem Failure Mode % in MLD	On-Orbit Satellite Subsystem Failure %
GN&C	24	7.3%	n/a
ACS	54	16.5%	19.0%
Sum of GNC & ACS	78	23.8%	26.0%
Propulsion	47	14.3%	10.0%
Structures	34	10.4%	3.8%
C&DH	28	8.5%	6.5%
Communication	22	6.7%	5.6%
Thermal	63	19.2%	1.0%
Power	56	17.1%	16.4%

This comparison is valid only in general terms because there are a number of differences between the MLD and missions that have flown. For the on-orbit failure rate, other subsystems that caused failure are not included in this comparison. These include the launch vehicle, kick motors, the payload, program management, operations, and unknown error sources. The launch vehicle is not counted in this table because the comparison would be totally different – the MLD shows how the launch vehicle can damage the satellite, whereas the percentage from industry shows the number of catastrophic failures due to LV errors. The payload can cause many mission-ending failures, but it is not included much in the MLD because the payload's failures depend highly on the design and the mission chosen, making it a bad fit for a general MLD.

[iii] The On-Orbit "Sum of GNC & ACS" average includes GN&C, ACS, and related software failures, as is normally recorded by industry.

Software is included in the subsystem failures, and operations are scattered throughout but not often included as a root error since that is hard to classify before the mission. So, many of the failures that on-orbit satellites see are not considered for the MLD because their rates of occurrence in the MLD are low due to the focus of the MLD and the way it's laid out.

Another difference is that the mission purpose leads to certain types of failures, which skews the percentage of failures for a given subsystem. For example, missions that need to slew a lot run out of propulsion often, and that subsystem is held responsible for the mission failure. Since student missions have different goals, their failure modes might not be exactly the same as the missions that have flown. Finally, failures have different probabilities, but the MLD assumes that everything in the diagram has the same probability. Many components are more reliable than others, but assigning probability in the MLD is out of the scope of this study and is unnecessary for this comparison. Therefore, equal probability of failures will be used, and the two sets of percentages can be generally compared.

Looking at Table 9, one can see that most of the subsystems have percentages similar to each other when comparing the two columns. There are major differences, though, for the structures and thermal subsystems. In these two, the number of possible failure modes is much higher than the failures experienced on orbit. This result has multiple reasons, but the best explanation for reduced failures on orbit is pre-launch testing. Failures in these two subsystems are usually due to infant mortality and are caught by testing. Every satellite goes through extensive structural loading, vibration, vacuum, and thermal testing to qualify for flight. Because of these tests, which have a long and mature history, problems are caught before the satellite is launched, reducing the number of on-orbit failures drastically. The environment for structures and thermal is also well-known, so there are fewer surprises on orbit, also reducing the number of failures. Other subsystems have a harder time testing as if they were in the space environment. Most of the other systems include sensitive components, including software and electronics. These systems are hard to fully test before launch, and many aspects of space can be detrimental to those parts, making failure more likely.

It can be seen from this study that the number of times a subsystem failure appears in the MLD may be related to the probability of failure on orbit, except for structures and thermal. Since the MLD shows the probability of a subsystem occurring, another step in the risk management process is partially done by the MLD. Universities should take this into consideration when thinking about what their high risks items are and where to focus their attention.

5.4 Risk Management Process

It has been shown that the master logic diagram is a useful tool for helping students to identify risks in a thorough, organized, and unbiased manner. Now that the MLD is developed, it is necessary to incorporate it into the risk management process. As discussed in Section 2.1.1, the steps of risk management are:

1. Understand the system and what constitutes a risk
2. Identify the risks
3. Analyze the risks
4. Make a plan for risk mitigation
5. Monitor and update the risks

The MLD helps cover the first two steps of this process; if the students have in mind what the program defines as risky, they can study the MLD and apply it to the university's program. There are many ways to continue the process of risk management, and some of those methods were outlined and discussed in Section 3.1.

To keep track of the MLD, it was put on the internet for general viewing purposes. This way, all students would have access to the template, so they have the advantages of using and seeing it without being able to edit it without approval. The MLD should be kept under configuration control, requiring a change order for anyone on the team to make changes to it. In fact, all important design documents should be kept under configuration control. This ensures that the whole team agrees on the modifications and is aware of them. It is best for one person to be in charge of the MLD, and for the Mars Gravity mission, this is the Systems Engineer. This person is the main one responsible for the MLD's upkeep and making sure that the entire process is flowing as needed.

The third step of the management plan, analyzing the risks, can be very time consuming and overwhelming for student projects. Usually this analysis involves identifying the probability, severity, and timeframe of the risks. Some general guidelines for these steps follow. Assigning probability to every risk can be horribly time consuming because of the difficulty in gathering information. It is recommended not to assign probabilities to each independent risk because student groups do not have the experience or the resources to spend on this task. Instead, the team should look at the likelihood of the risks occurring. This likelihood was shown in the MLD by the number of failure modes that occur in each subsystem. The team can use this information in a general way by giving the risks qualitative rankings of low, medium, and high, or numerical rankings based on those types of categories. The likelihoods can be color-coded (usually red, yellow, and green) to make the table easier to read. The university does not have to use the probabilities from the MLD, of course, but they should classify likelihood in some way in order to have a better idea of how possible the failures are.

The MLD also helps evaluate the severity of each risk. First, the risks are important because each relates to a catastrophic end state. Second, the MLD shows the impact of a failure on other subsystems since they are interconnected in the MLD. The team should indicate the severity of the risks so that resources such as time, money, and personnel can be allocated accordingly. MITRE, a Federally Funded Research and Development Corporation that has a large focus in risk management, details the levels of severity as follows[62]:

1. Catastrophic - Complete mission failure, death, or loss of system (inability to achieve minimum acceptable requirements).
2. Critical - Major mission degradation, major cost or schedule increases, severe injury, occupational illness or major system damage.
3. Moderate - Minor mission degradation, small cost or schedule increase, injury, minor occupational illness, or minor system damage.
4. Negligible - Less than minor mission degradation, injury, occupational illness, or minor system damage.

The labels or numbers for severity can be used when assigning this attribute to the failure modes, and color can again be used to help distinguish between categories (where both "moderate" and "negligible" are green). Lastly, the timeframe of each risk can also be tracked to make sure that the risks don't delay the project.

It is important to note that not all of the risks must be maintained in this manner. The engineering and program management can decide on the top risks and only analyze those failure modes for likelihood, severity, and timeframe. This method is used by many universities and industry projects, but the MLD must be used beforehand to identify the complete set of risks related to the satellite. Using the Top Risk List to focus in on the most important risks will save time over working with all the risks in the MLD, but the team must also periodically review the list to make sure that the top risks are still correct. The Mars Gravity team took the Top Risk List approach – the systems engineering lead of the satellite studied the MLD and identified the top risks perceived to be affecting the mission, but the exact number of risks to be specified was not set beforehand. The payload and EDL engineers also identified risks from their systems that were not already identified by the MLD. This was necessary since the complexity of the Mars Gravity project is not fully captured by the MLD, but this step should not be needed for simpler, smaller missions.

Then, the likelihood, severity, and timeframe of each risk were evaluated and recorded on the same sheet. The list was agreed upon by the program manager, science director, and the leads of the payload, EDL, bus, and systems engineering teams. This entire step took just about 1.5 hours to complete, and the risk management plan was officially in place. This was a nonscientific means to identify the top risks, but for a student project, it certainly got everyone thinking about risk as a team.

Once the risks are analyzed for these factors, a mitigation approach is needed. Preferably in the same document, the team should decide on the plan of action for each risk. This group should outline the mitigation method, decision points, and tests to verify that the risk has been reduced. If it works out in the timing, the decision points can be all at the same time, allowing the team to review the risks together.

The last critical step in the risk management process is to maintain the risk lists. It was decided on the Mars Gravity team that the process of creating the top risk list should be repeated once a semester. However, both the MLD and the top risk list must be monitored throughout the semester by a systems engineer to add changes or updates to the MLD and the risk statuses.

Continuing with Mars Gravity's implementation, the next step was to have the team leads decide on a mitigation strategy for the risks pertaining to their subsystems. To show an example, the risk list and mitigation plans for Mars Gravity are shown in Table 10. If these risks also affected other subsystems, the first team brought that to the attention of the affected team to decide upon a solution. At each systems team meeting, the agenda would include a risk discussion, and any concerns the team was having would be brought to the group. If the systems engineer noticed any deadlines coming up, he or she could also bring those risks up for discussion.

Table 10. Mars Gravity Top Risk List and Mitigation Plan

Top Risk List	How this failure occurs	Likelihood (1-3)	Severity (1-4)	Time-frame	Mitigation Plan
Stuck thruster	Parts not reliable	3	1	PDR	Space proven parts
Electrical Failure	Radiation, Parts not reliable	3	1	PDR	Space proven parts, redundancy
Bus batteries fail	Physical and electric system not well designed	2	1	PDR	Space proven parts, ground testing
Software failure	Lack of testing and not robust enough	1	2	CDR	Contract out software, test thoroughly
On-board communication failure	Comm equipment fails, or C&DH fails	3	2	PDR	Space proven parts
Solar panels not drawing enough power	System not designed correctly, or parts fail	1	1	PDR	Account for all power needs while designing panels; heritage
Not enough power for deployment series	Batteries under-designed, panels not deployed fast enough	2	1	PDR	Design operations sequence for nominal and off-nominal, size batteries for this purpose
CG unstable	Mass over budget, mass not well distributed	3	1	PDR	Account for all mass, move mass toward nose, ballast in heatshield nose, verify CG location through ground testing
LV catastrophic failure	LV fails independent of satellite	1	1	PDR	Choose low cost option with risk as a secondary concern
Thermal Interface, Bus to Payload	System not designed correctly, or parts fail	1	1	CDR	Model and analyze the system with FEM and Matlab; heritage; re-do calculations for how much thermal energy will really exit payload
Computer fails, can't re-enter or run payload	Single-string computer fails	2	1	PDR/ CDR	In case main processor fails: redundant reentry microcontroller system and redundant life support microcontrollers system
Payload batteries fail	Physical and electric system not well designed	2	1	PDR	Space proven parts, ground testing
Mice die of radiation	Solar flare	3	1	Launch	Launch at a time when solar flares are not predicted
Water leaks in payload and floods mouse	Punctured water reservoir	3	2	PDR	Design 15 reservoirs so that if one fails we still have 14 mice alive
TPS failure	Entry not the designed correctly, heat load exceeds design loads	1	1	PDR/ CDR	Arcjet testing of TPS, high fidelity aerothermal simulation
Failure of parachute deployment	Computer never sends command, trigger mechanism fails	2	1	PDR	Redundant triggers: computer actuated, timer, mechanical accelerometer
Aeroshell depressurizes during EDL	Seal between aftbody and forebody fails (maybe after TPS jettison)	2	2	PDR	Ground testing, add additional bolts to seal plate
Recovery beacon fails to turn on	Parts not reliable, computer never sends command	1	4	PDR	Space proven parts

The risk management plan of combining the MLD with a top risk list and mitigation plan is certainly not the only method for incorporating the master logic diagram into a full management plan. However, it combines the already systematic MLD with an easy it implement Top Risk List, making this process a good match for students. As was detailed above, this was the plan chosen for Mars Gravity, and its results are detailed in the next section.

5.5 Benefit of the MLD for the Mars Gravity Biosatellite

The Mars Gravity program distributed the master logic diagram to the students of the Bus Engineering team for use in their subsystems and for understanding the big picture of the satellite program. The response to the MLD was very positive, and students were able to learn about both objectives. In meeting with the students after they had a chance to review the MLD and think about its application to their subsystem, they were asked the questions below. Their responses are also included.

- "What risks are in the MLD that you hadn't thought of before? Did you learn anything from looking over the MLD and thinking about risks for your subsystem?"

The students stated that the MLD helps to more clearly and logically show the potential problems, especially ones that the students had not thought of or did not think were important. A common failure mode that students had not thought of was inter-related errors. For example, a thermal student realized from the MLD that he needs to consider what would happen if a component fails, reducing its heat output, and unbalancing the thermal environment that he's carefully designed for the nominal case. Similarly, a student on the communications team had not considered operating temperatures for her hardware as a mission critical specification. Now, as she considers hardware choices, she knows what parameters are important for the component. As discussed in the advantages of the MLD, it can show students that most of the systems are closely linked, and if one thing goes wrong, everything else is affected.

Another student mentioned that she did not think of sensor failures on the propulsion system she had been working on. She knows there are sensors feeding back information about temperature and pressure, but this communications/data aspect was more foreign to her, and therefore she had not thought about them failing and the affect that would have on the system. Electronics are also an area that students focus less on. A student on the power subsystem noted that he had not yet thought about short or open circuits in the design of the solar panels and their distribution system. More generally, it is important to note that each student, from freshmen to graduate students, who used the MLD recognized failure modes that they had not yet considered in their design.

Finally, through the MLD, each student was exposed to the programmatic risks. Often, risks at the program level are not shared or well communicated to the entire team, and having them documented, even in a separate page of the MLD, made students better aware of the risks that their student satellite faces because of the way the program is set up.

In summary, the MLD has helped Mars Gravity increase awareness about failure modes at the technical and programmatic level while also helping to identify ways that the system can fail that the students had not yet considered.

- “Do you think anything is missing from the MLD that's needed to show how the system can fail?”

Since the MLD was reviewed before being distributed, it was expected that students would not be able to identify failure modes that were not in the MLD. However, they were still asked to think about that issue. As anticipated, none of the students had suggestions for other major risks that needed to be included, but a few graduate students suggested areas of further detail on their own subsystem, such as the types of mechanical and electrical failures for batteries. These were not included to try to maintain the level of detail for the failures of components.

- “What do you like about working with the MLD?”

The students liked the structure of the MLD the most. They thought it was easy to use, clearly showing the path of failures and that a failure’s flow to a particular subsystem can be mapped out in a logical manner. The cause and effect chain also presented itself through this structure, so the students had a better picture of information flow through the system. This could help determine the number of state-of-health sensors as well. Students also thought training of ground operations crews would be easier using this satellite failure diagram.

Students commented that the MLD makes design easier because it shows what is directly affected by a design choice. By using the MLD when designing a system, the students know who they need to talk to in order to discuss changes. This learning process through the master logic diagram was a benefit of the MLD that the students liked a lot.

- "What do you dislike about working with the MLD?"

There were only a few minor comments about dislikes relating to the MLD. First, the diagram uses the symbols commonly found in fault trees, and these symbols are probably new to students and not necessarily intuitive. Even though a legend is provided, it was found that having to reference it was annoying, but manageable. Another student mentioned that the MLD should take into account the magnitudes, or severity, of failure. The MLD does not take that into account, but the top risk list does, and incorporating severity (and probability) into the MLD could be future work items.

As expected, one student mentioned that the MLD is "kinda scary" due to the number of failure modes and the fact that it is recursive, but she also thought it was very useful. Some students to want to be able to solve all of the problems shown, but they have neither the time nor the knowledge to do so. On the other hand, one student wanted even more information on every possible way his subsystem could fail, but that request for information has to be balanced with the apprehension of providing too much information and the need of applicability to all satellites. The size of the MLD can be overwhelming, but having students browse the whole thing while only focusing on their subsystems can hopefully prevent the MLD from being too overpowering.

From this feedback from the students on Mars Gravity, it is apparent that the MLD helps students see failure modes in their subsystems. Each subsystem had risks that were identified because of the MLD, and plans were made within the subsystem teams to mitigate the risks that the MLD caught. For example, interface failures were better understood after using the MLD, and this risk was mitigated through better communication at team meetings about inter-subsystem issues. Meetings across systems, such as between payload and bus, were also begun to facilitate communication about thermal issues between the teams, including plans to mitigate risks related to the payload-bus thermal system. Data and software interfaces were another part of the system that students were not familiar with in terms of failure modes. At the system level, an effort was started to better understand software requirements, making sure to recognize to the requirements that affected more than one system. This information was then discussed with the students on each team so that they were more aware of data and software failure modes in relation to their subsystem.

The most important comment to note is that the students liked working with the MLD. It provided them a clear picture of failures and brought risks to their attention that they had not considered. Risk management is often viewed as an overwhelming and tedious task, but it seems that the MLD is not viewed as such by students. Therefore, the MLD should be used for student-run small satellites because it is easy to implement, and it is useful for students to identify risks in a logical manner.

Chapter 6: Summary and Recommendations

6.1 Summary and Contributions

This thesis discussed programmatic and technical risks related to student-run satellites and suggested a new way for university programs to do risk management, including failure mode identification.

Chapter 1 outlined the motivation for the thesis – student groups need to do risk management to maintain cost and schedule as well as to learn an important part of systems engineering. Chapter 2 overviewed the steps of risk management, and the programmatic differences between small satellites and industry satellites were investigated. Then, unique risks facing university satellites were identified and discussed; these risks included funding and competition, experience, staff, direction, schedule, documentation, and recruitment. Next, technical risks for university and industry satellites were researched, and the two sets of results were compared. It was seen that it is still hard to compare student and industry satellite failures because of the lack of student satellite data, but from the available information, the power subsystem is often a failure mode for student satellites, potentially because of its high energy levels. Universities should also utilize the failure rate information presented when planning their technical risk management plans.

Chapter 3 presented methods used at other universities for risk management to see how schools tackle the problem of managing risk. It was seen that many schools perform ad hoc risk management, if they do any risk management at all. This section also discusses government risk management techniques, but those were shown to be too involved and time-consuming for student. It was then determined that a framework for failure mode identification would help universities maintain a consistent risk management process since student groups use informal and not well-planned risk management processes in many cases. Suggestions for improvements of the programmatic and technical risks, including alternate platforms for space or near-space missions, were laid out as good options for student missions in relation to the problems presented thus far.

Chapter 4 discussed multiple techniques for failure mode identification, and the MLD was chosen as a good method for risk identification for small satellite projects with engineers that lack experience. The development of the MLD, along with its uses and restrictions, is presented. Finally, the benefits of the MLD for student satellite use are discussed.

The MLD was applied to the Mars Gravity Biosatellite in Chapter 5. First, a program overview was given to explain why the MLD was applied to this project and to show that any student project can utilize this risk management technique. Then, the MLD was demonstrated to be better at identifying failure modes than other risk identification techniques used by Mars Gravity students, and the MLD matches the outcome for percentage of failures per subsystem for satellites that have flown. Next, the MLD was applied to the Mars Gravity Biosatellite in detail, which demonstrated how the MLD can work for a real, complex student project. In this application, the MLD is shown to reduce risk and increase failure mode awareness for the Mars Gravity project, and students also enjoyed working with the MLD on their subsystems.

It has been shown that the MLD can help university satellite projects to implement a risk management process and identify both technical and programmatic risks. Then, the MLD can be combined with other simple risk management techniques to mitigate and monitor the risks. Overall, the master logic diagram has proven to be a useful framework for risk identification that can be easily adapted for student-run, small satellites.

6.2 Recommendations for Future Work

While the work to date is sufficient to apply the MLD to university programs, some future work is suggested. One deficiency of the current MLD is that it does not show which failure paths are more likely. Probabilities of failure or quantitative risk assessments should be added to the MLD to determine the probability of the failure modes. By associating probabilities with failures, even at a high level, the major drawback of the MLD would be eliminated.

To proceed with measuring the utility of the MLD, a group of metrics should be set to assist in measuring the quality of the MLD for a student project. This data would provide valuable feedback in determining how the MLD is being applied, what is working well, and where it needs improvement.

To make the MLD even more appropriate to all universities, the MLD should be tested at several other universities. Dr. Helen Reed of Texas A&M generously offered their participation in testing out the MLD, but there was not enough time to complete that phase of the study. This process would involve applying the MLD on a student project outside of MIT and then updating the MLD based on their suggestions.

Once this update to the MLD is complete, an online community for using the MLD should be established. It is important to keep control over the MLD template, but a means for feedback should be established. One option would be to establish a licensing structure that allows the users to recommend adjustments to the template, and updated versions would be put online to share. Also on this website, a repository for sharing information on failure modes and lessons learned should be created.

For other areas of this study, it would be best to have information on all failures for small satellites, no matter what the final status of the satellite was. This information would help to better understand how and why small satellites fail in order to see what processes might have prevented failure. This information can also be compared to industry satellites so that best practices can be borrowed from their. Failure information is difficult and time-consuming to get, but having a centralized, shareable collection of failure data would benefit all who do satellite projects at universities.

Appendix A: Master Logic Diagram Charts

The complete MLD that was created for student satellites can be seen in this Appendix. The master logic diagram is laid out in three levels. The top level is for the end state failure, and the second level is for subsystem failure. The third level is broken out for lower level failures, but some failures are on their own page solely because there was not enough room on the previous "Level 2" page. There are 17 pages of technical failure modes, and there is one independent programmatic risk page.

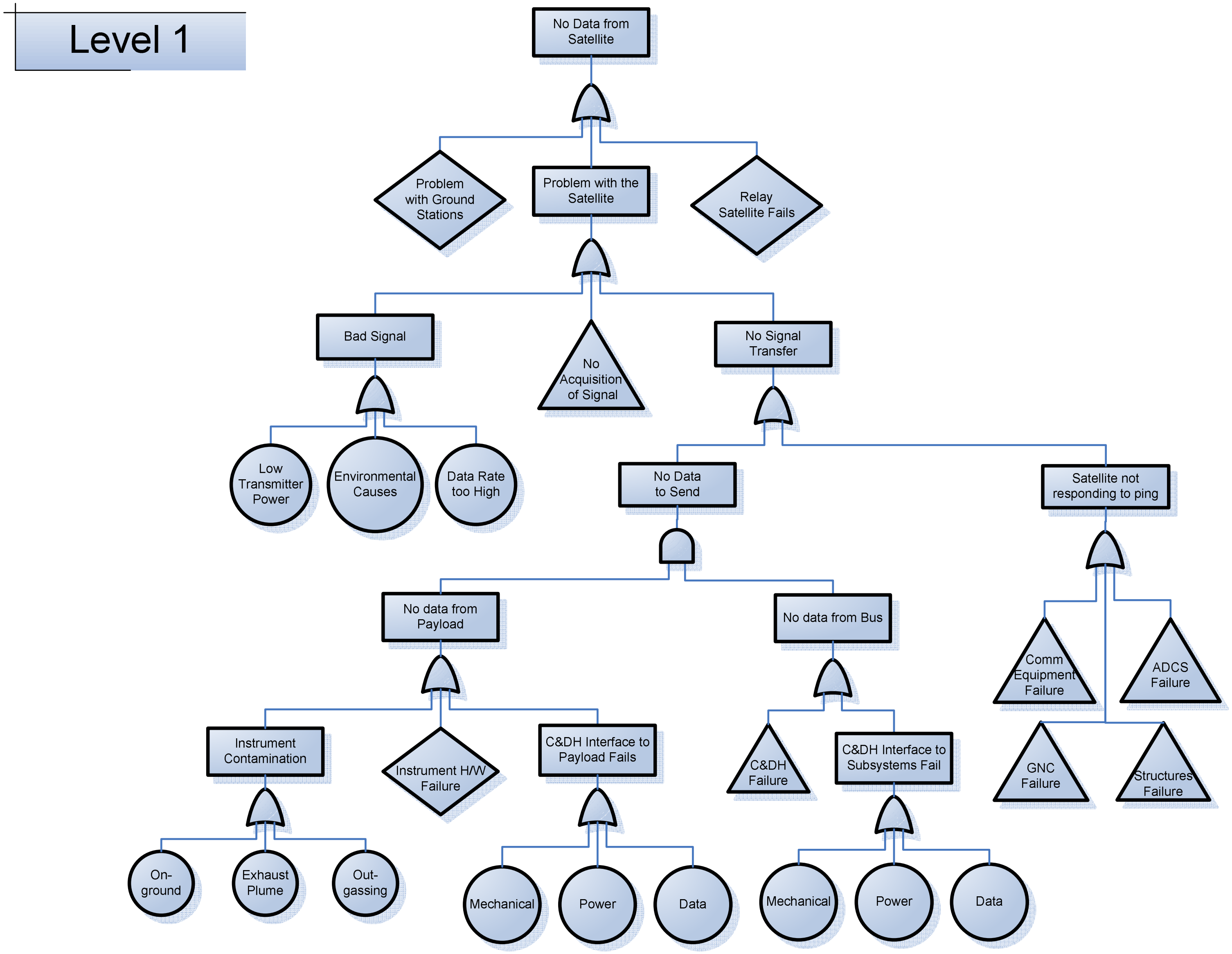


Figure 12. MLD for "No Data from Satellite"

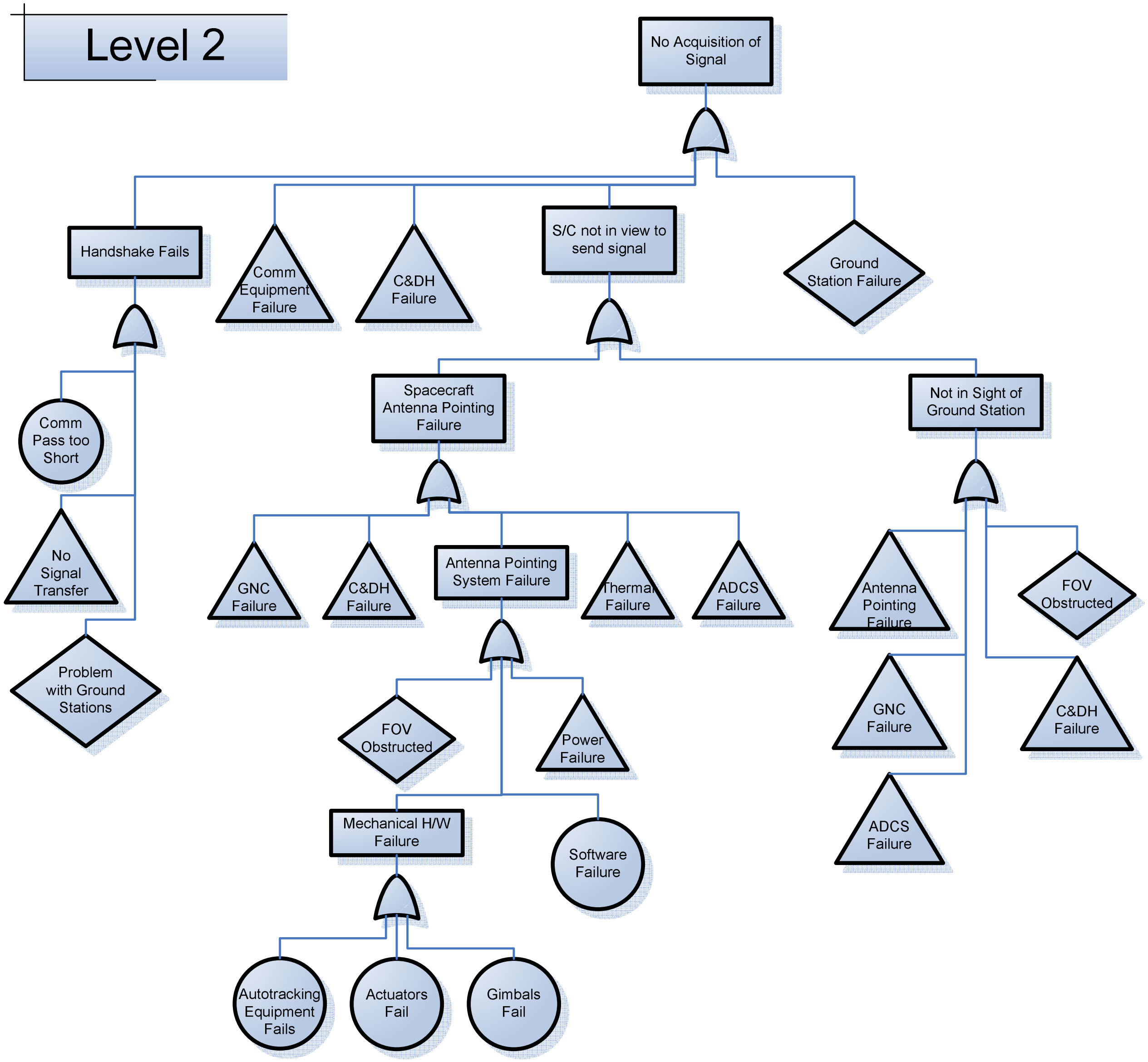


Figure 13. MLD for "No Acquisition of Signal"

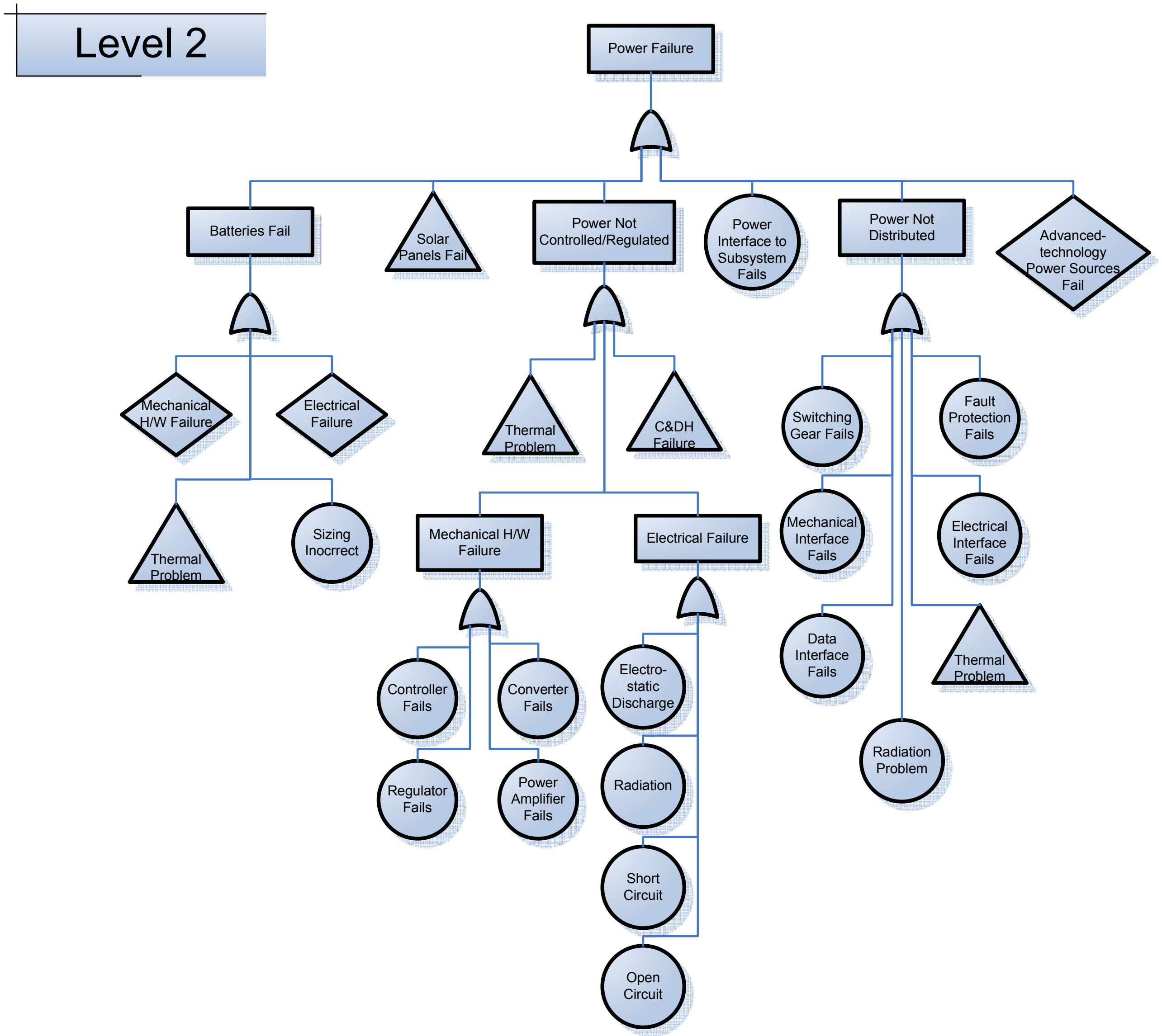


Figure 14. MLD for “Power Failure”

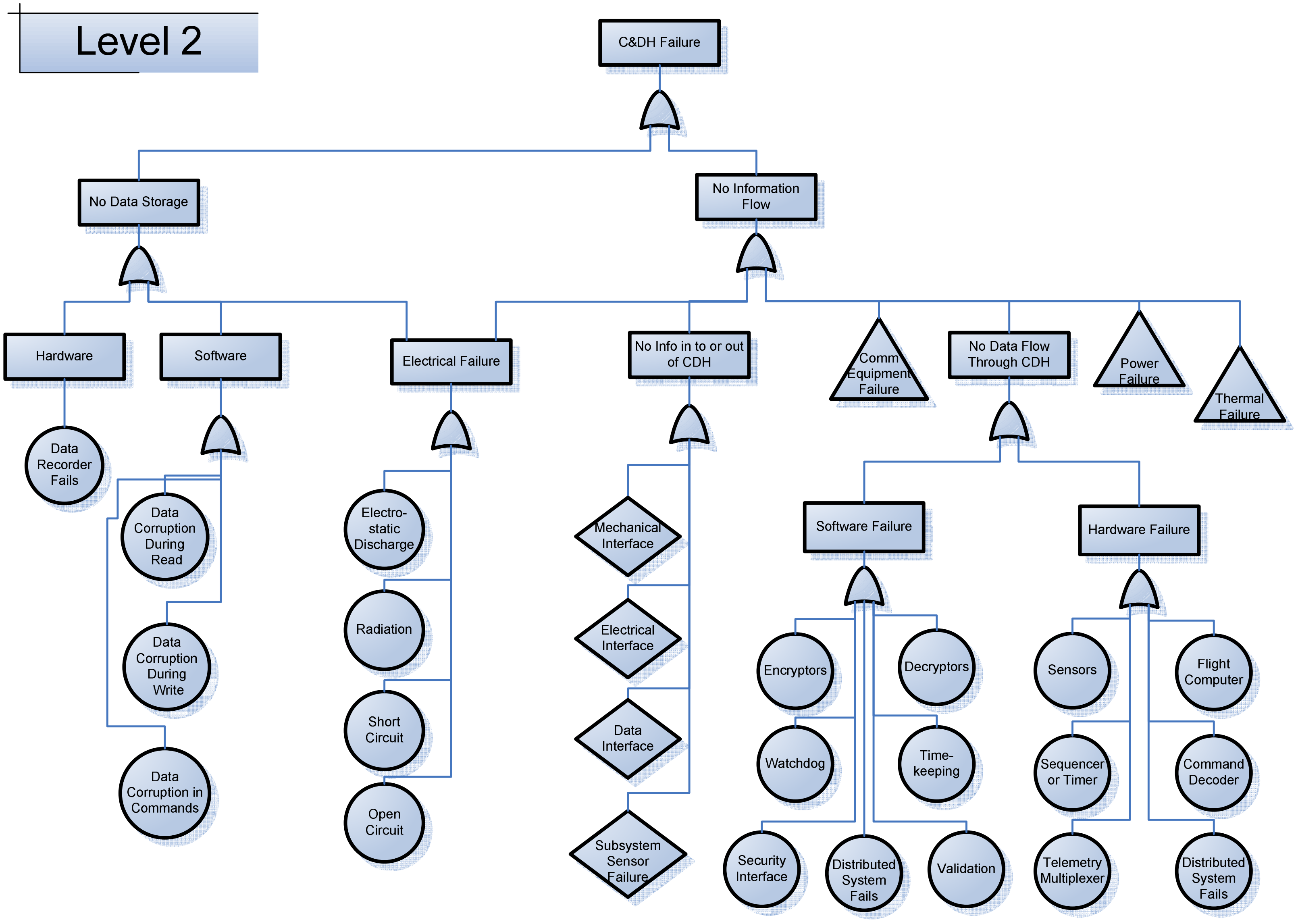


Figure 15. MLD for “C&DH Failure”

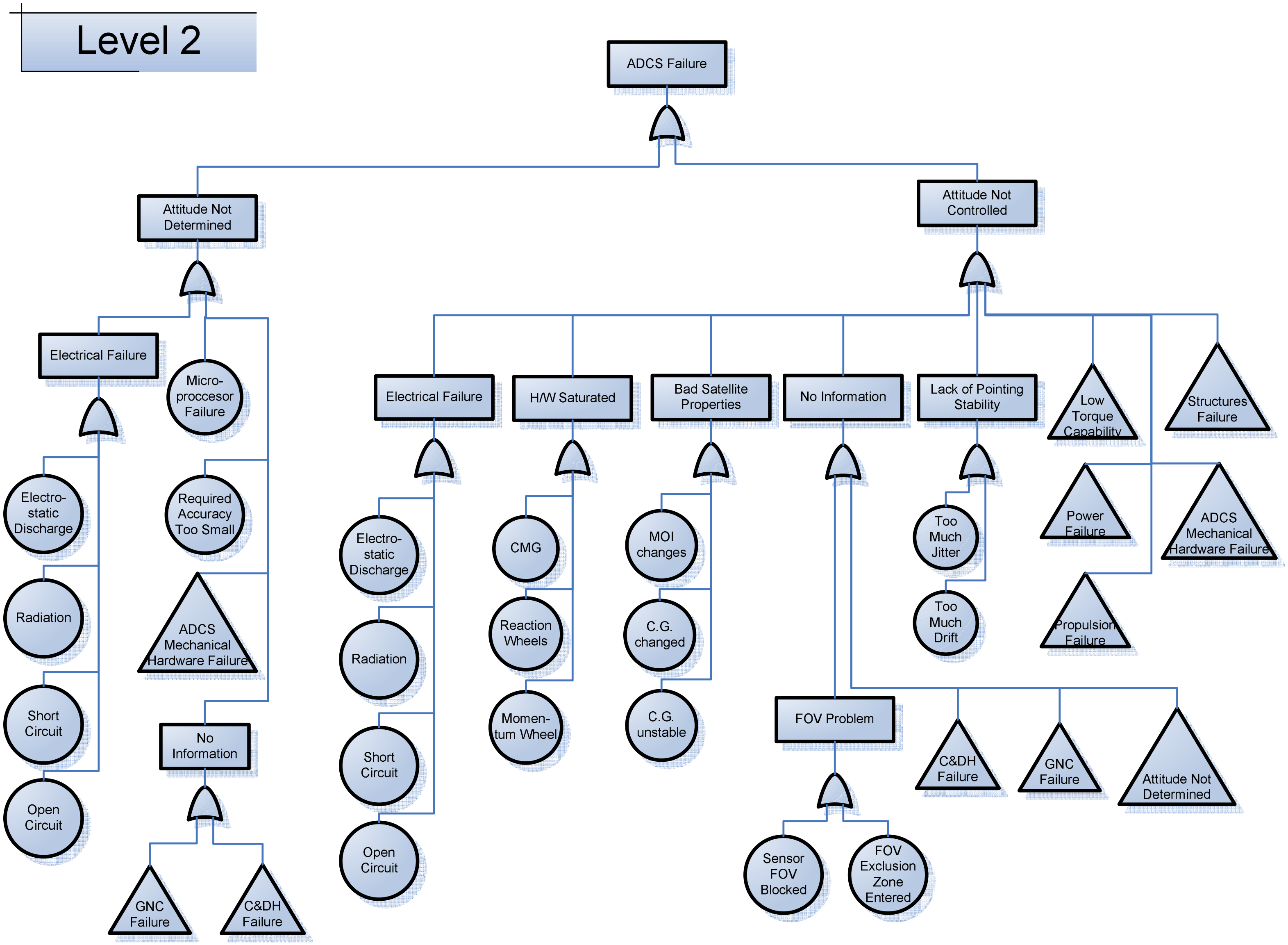


Figure 16. MLD for "ADCS Failure"

Level 2

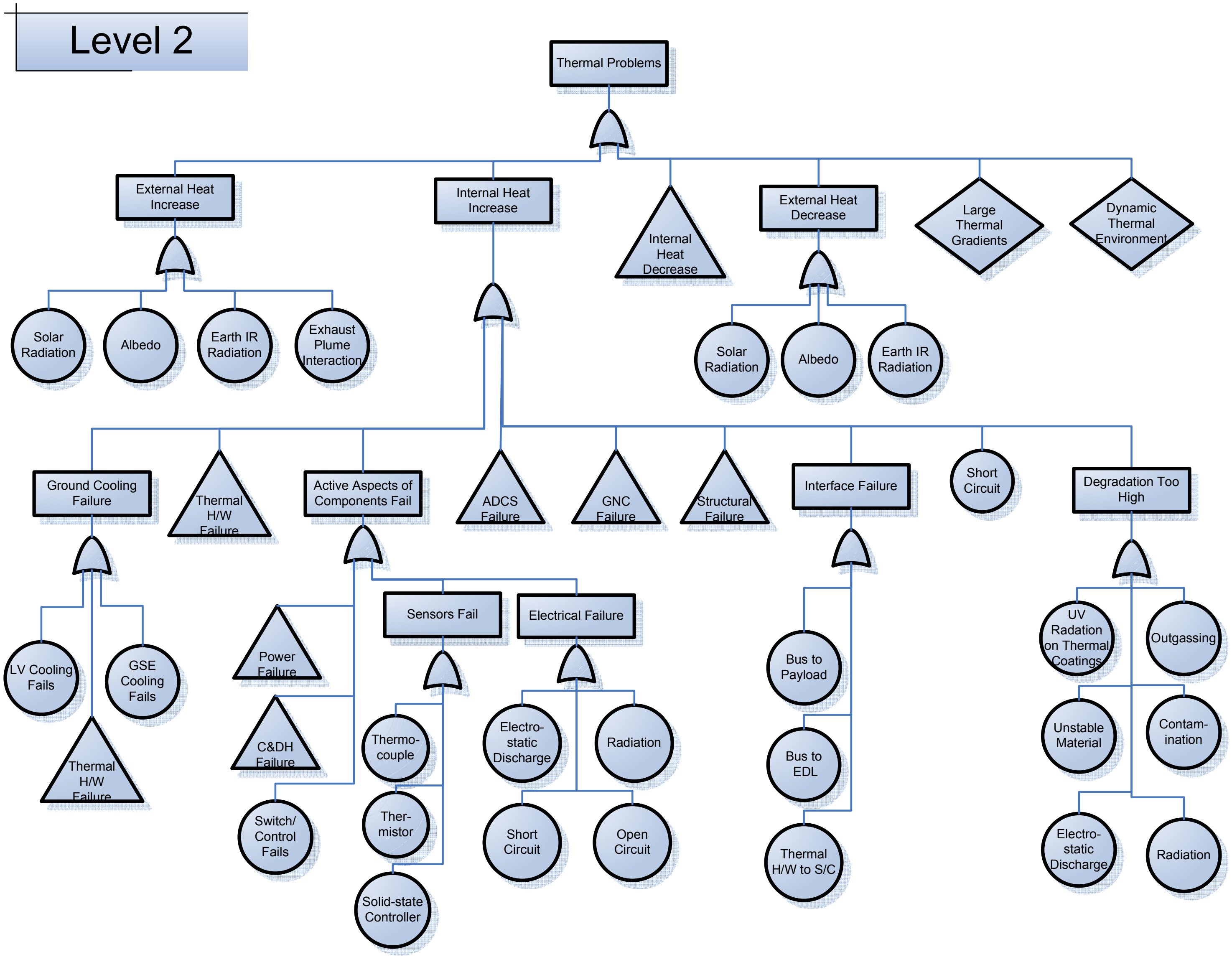


Figure 17. MLD for "Thermal Failure"

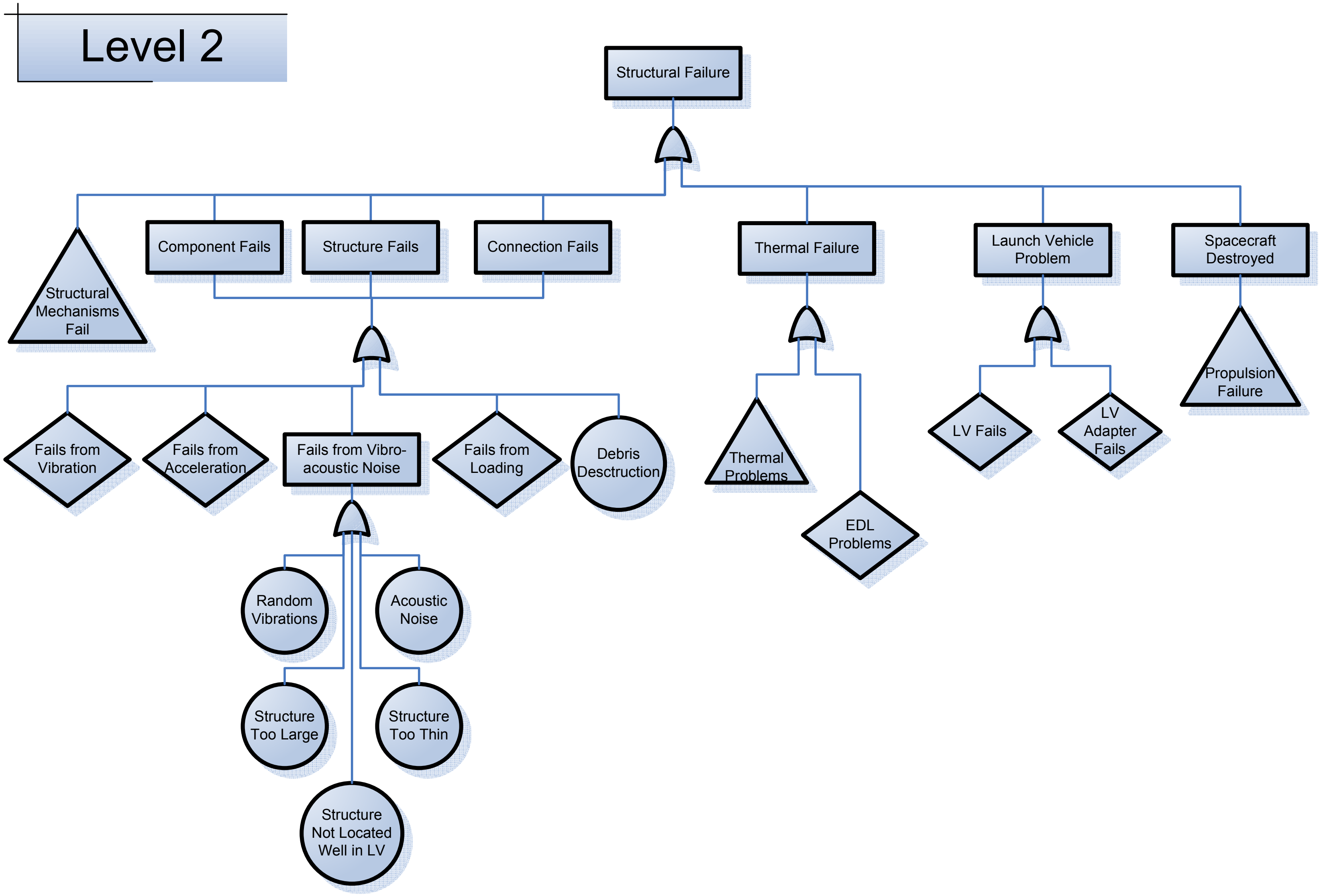


Figure 18. MLD for "Structural Failure"

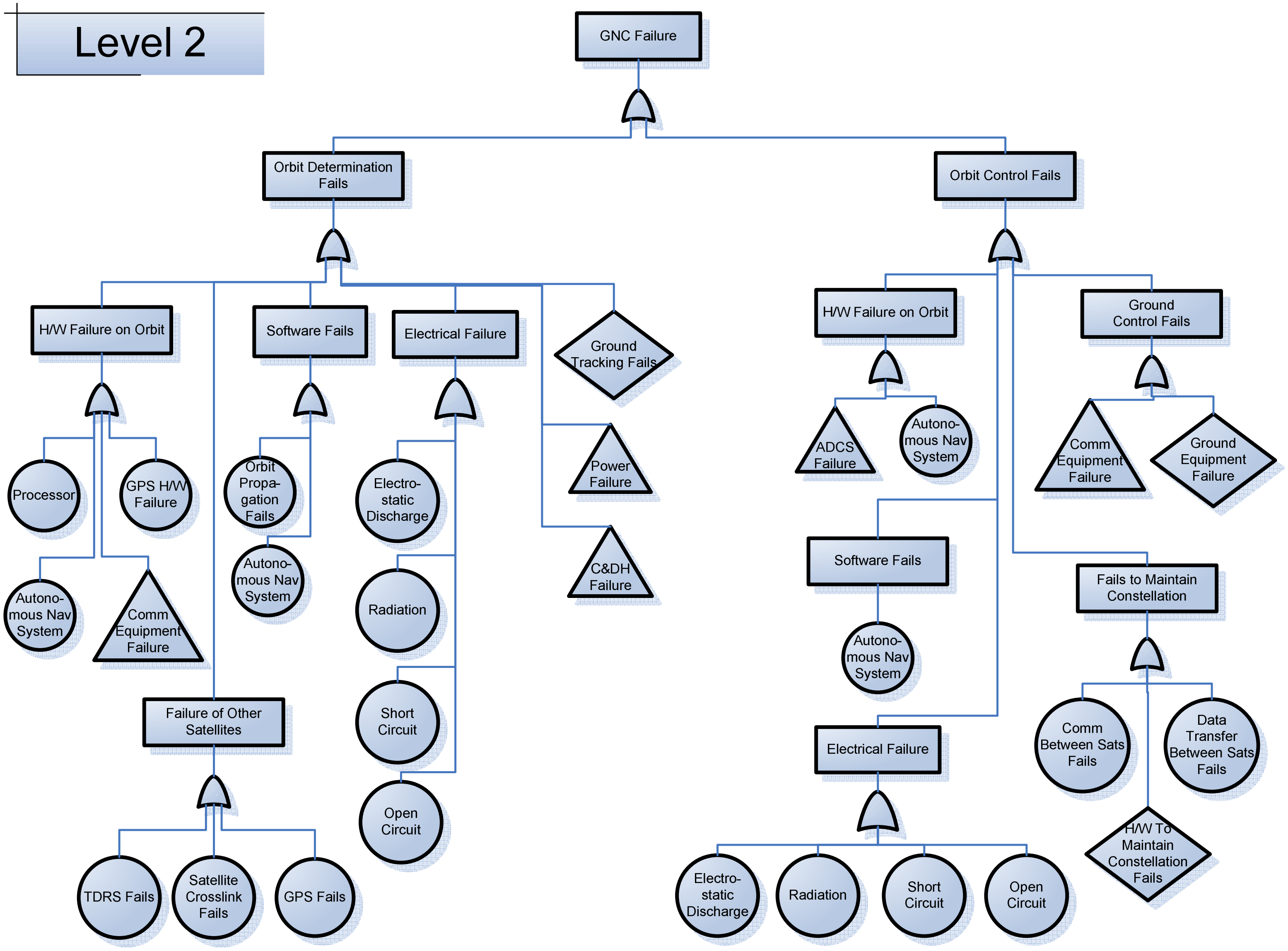


Figure 19. MLD for "GNC Failure"

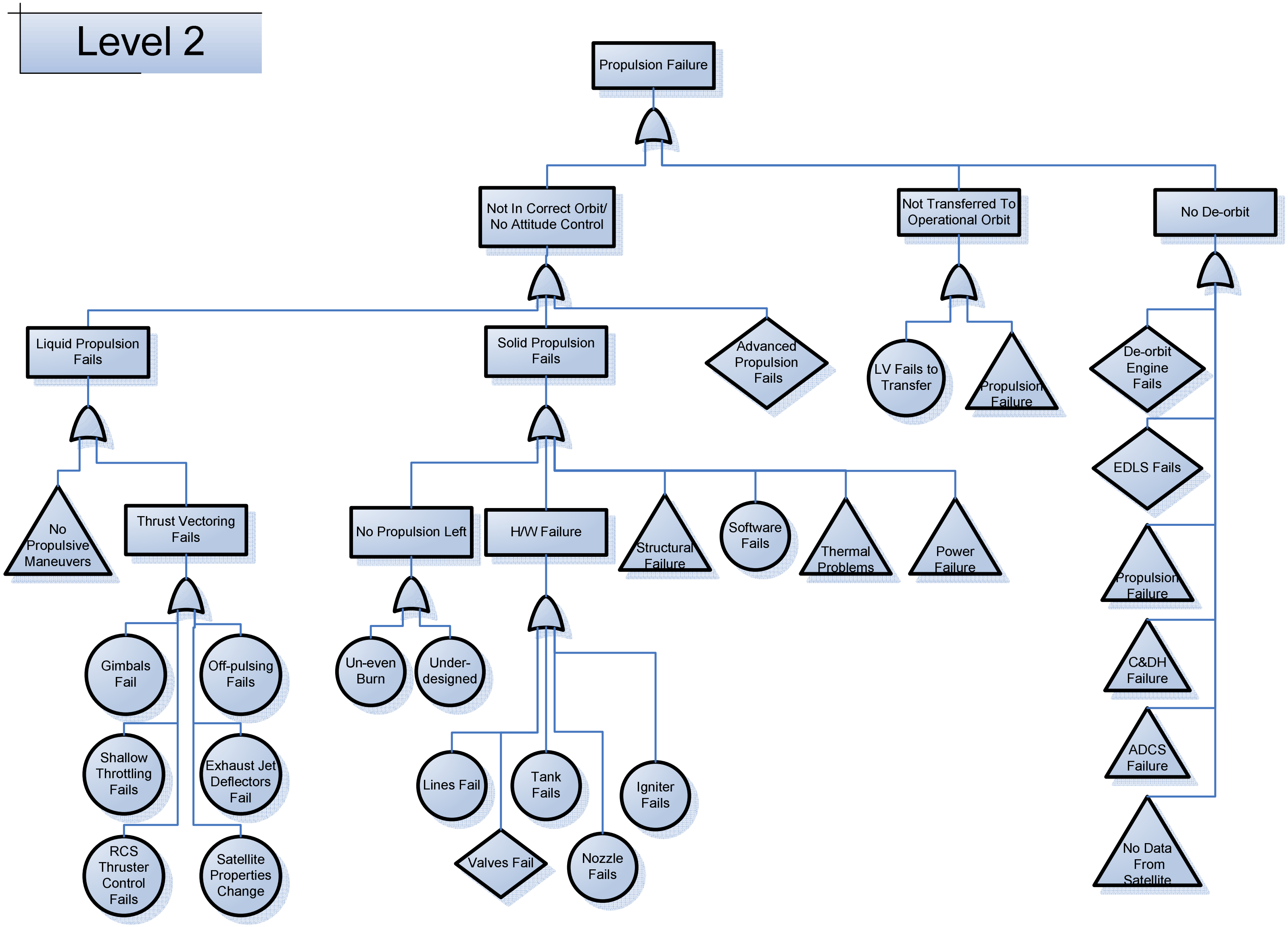


Figure 20. MLD for "Propulsion Failure"

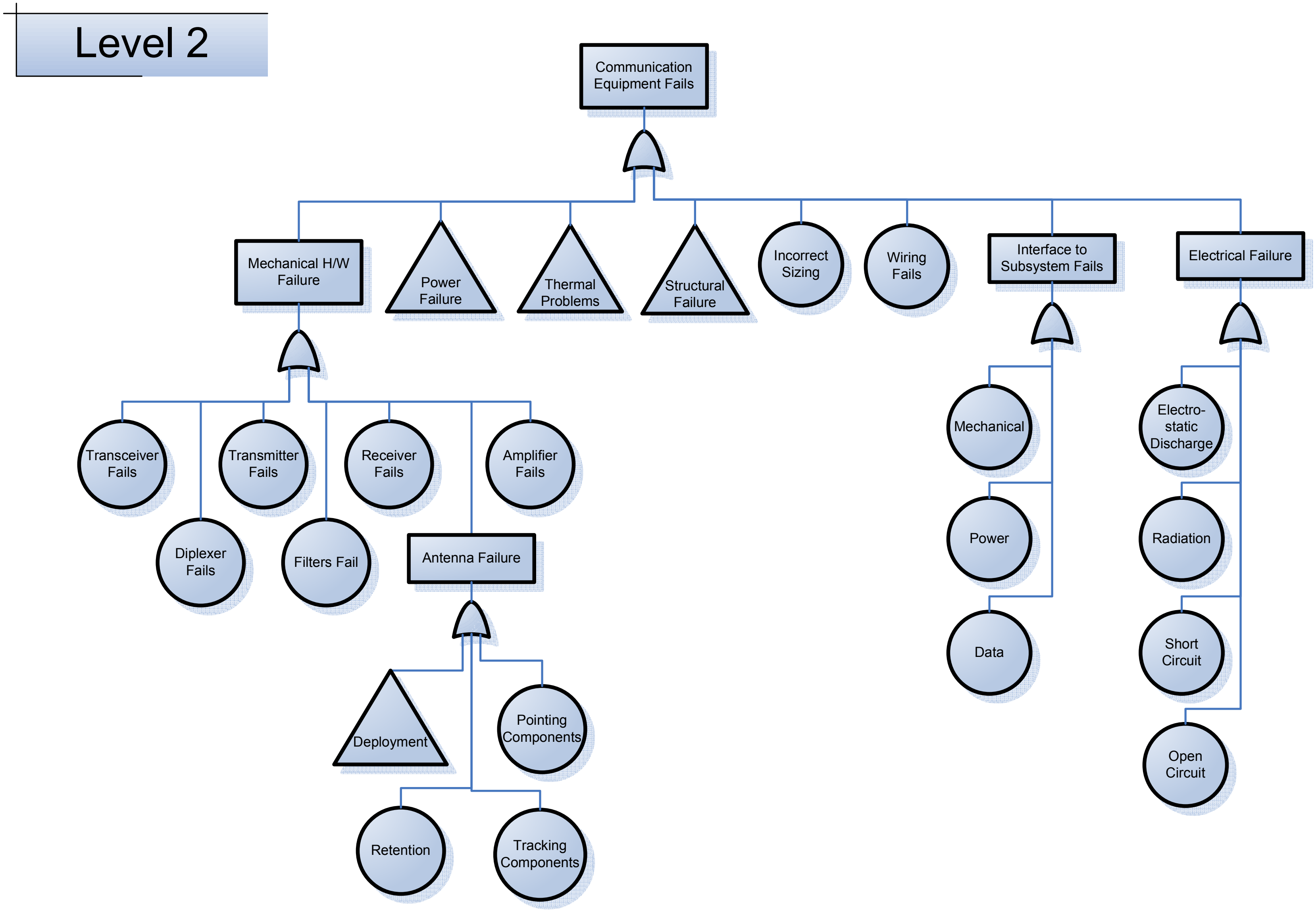


Figure 21. MLD for "Communication Equipment Fails"

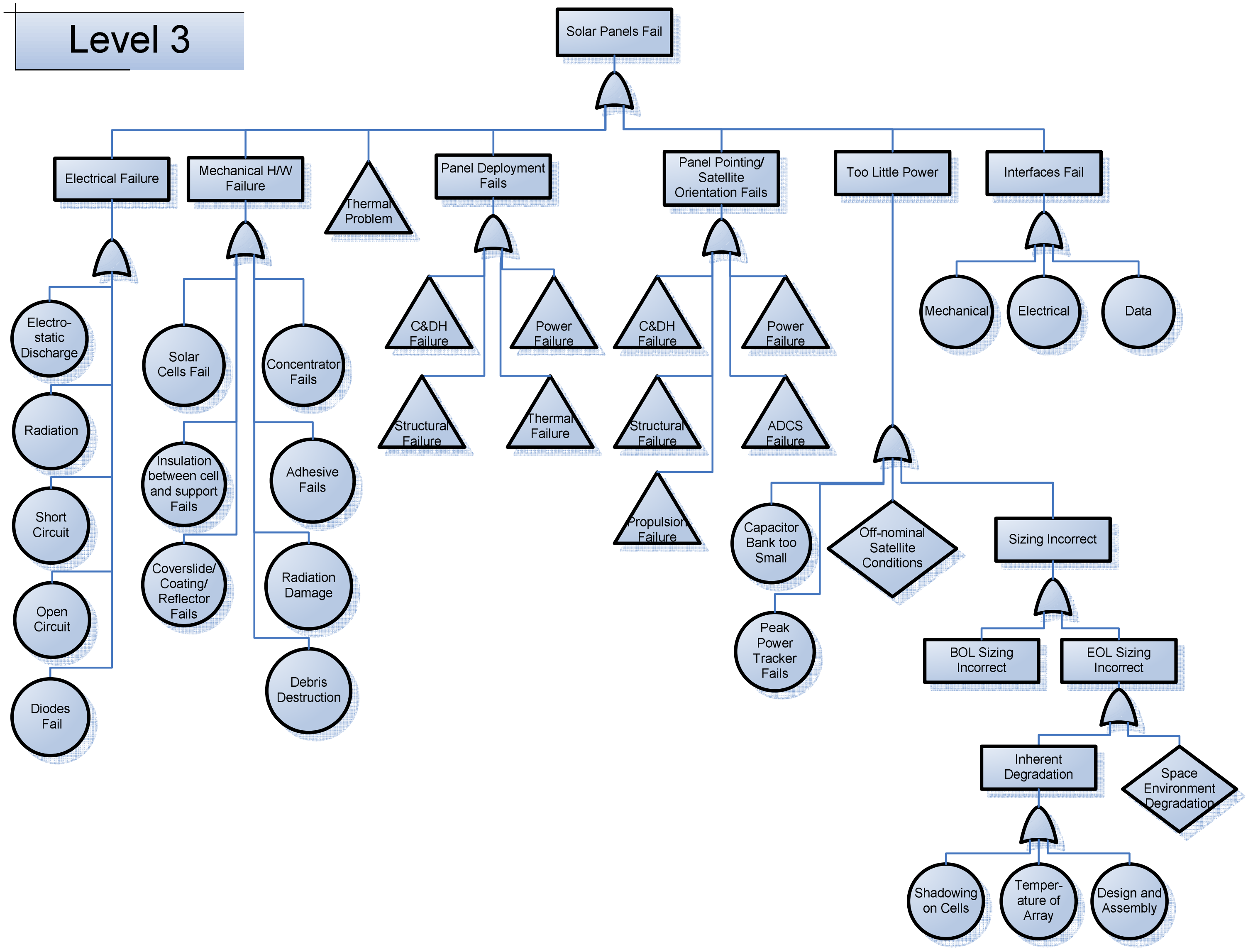


Figure 22. MLD for "Solar Panels Fail"

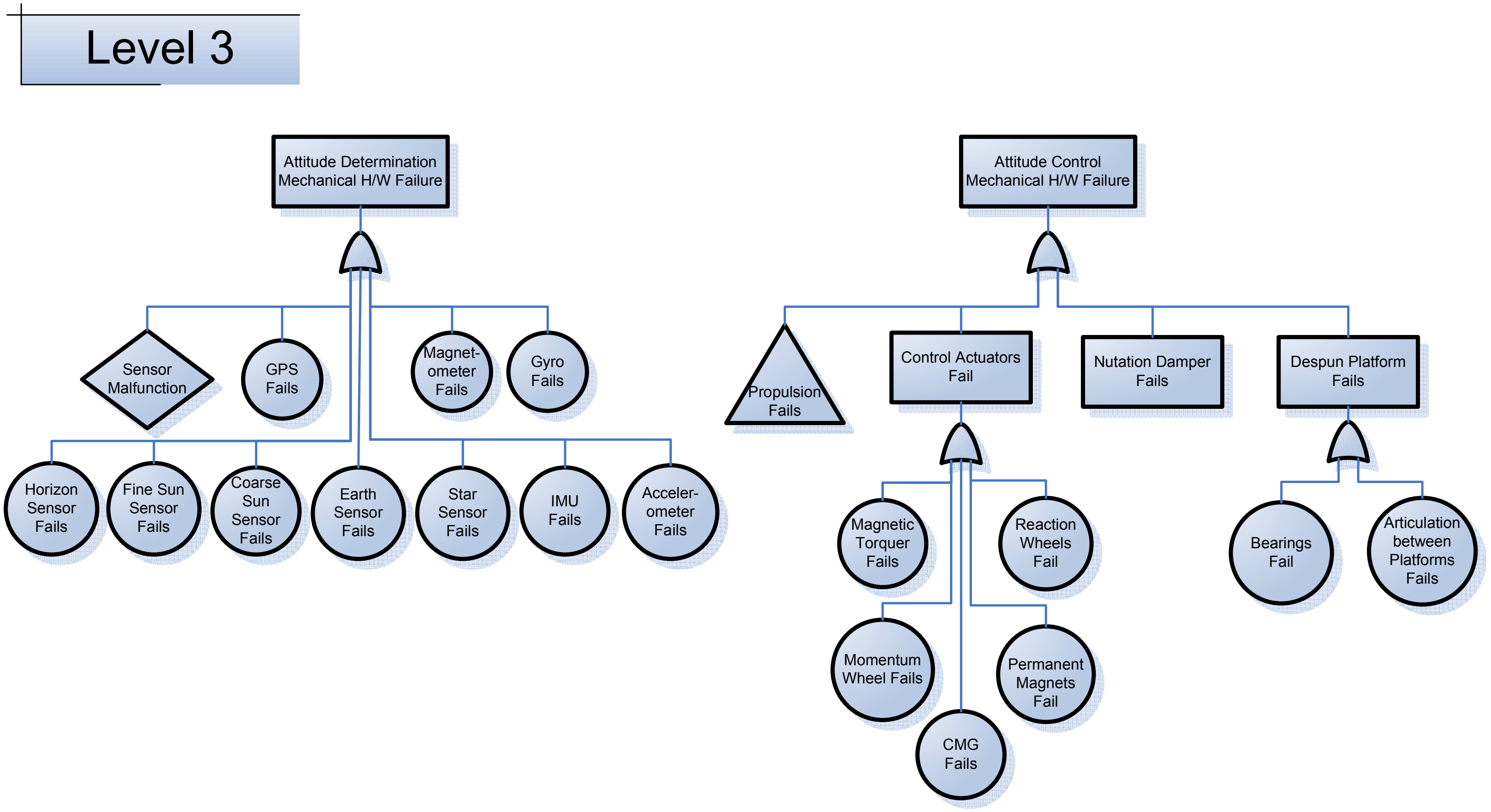


Figure 23. MLD for "ADCS Mechanical Hardware Failure

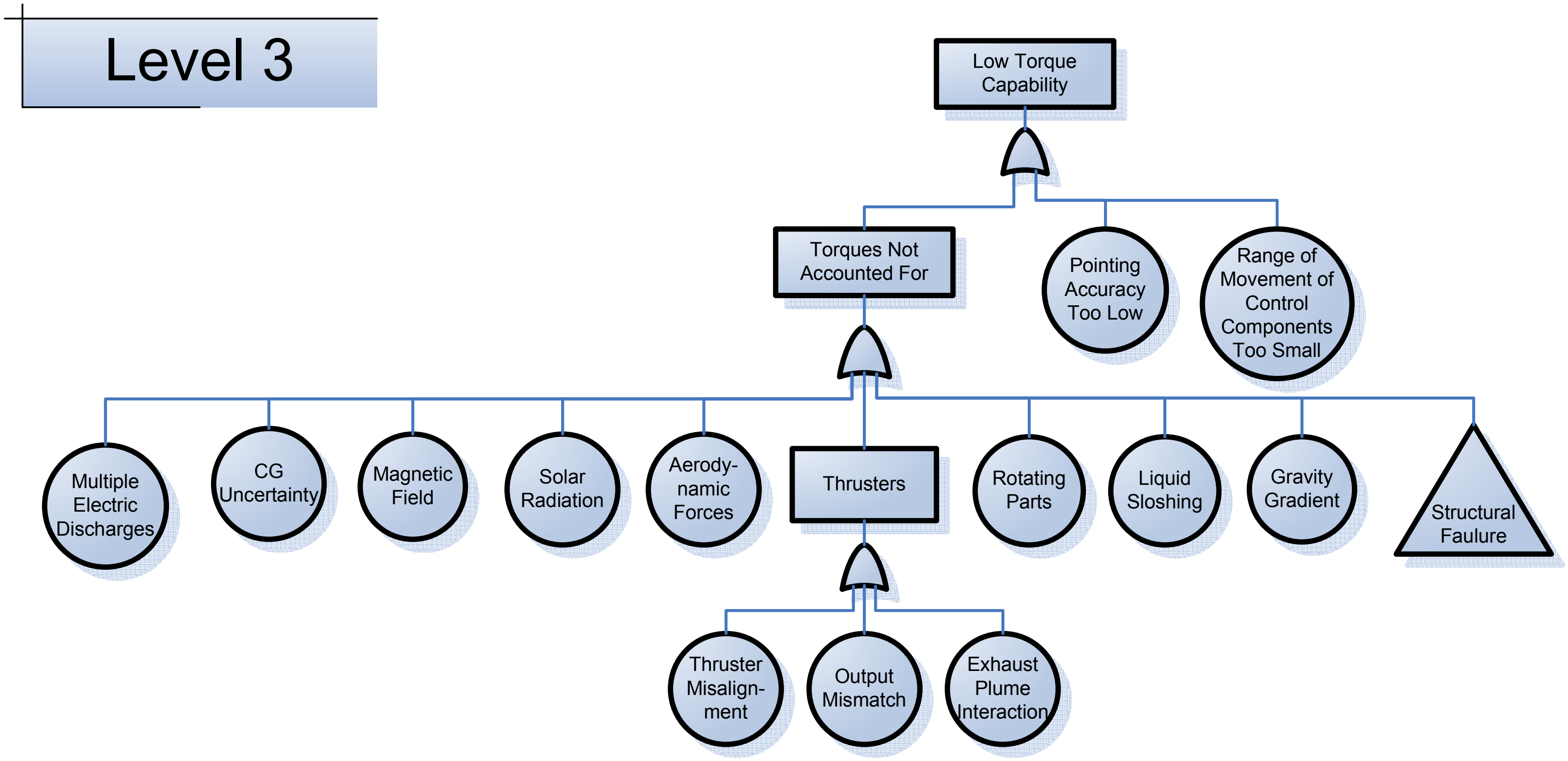


Figure 24. MLD for “Low Torque Capability

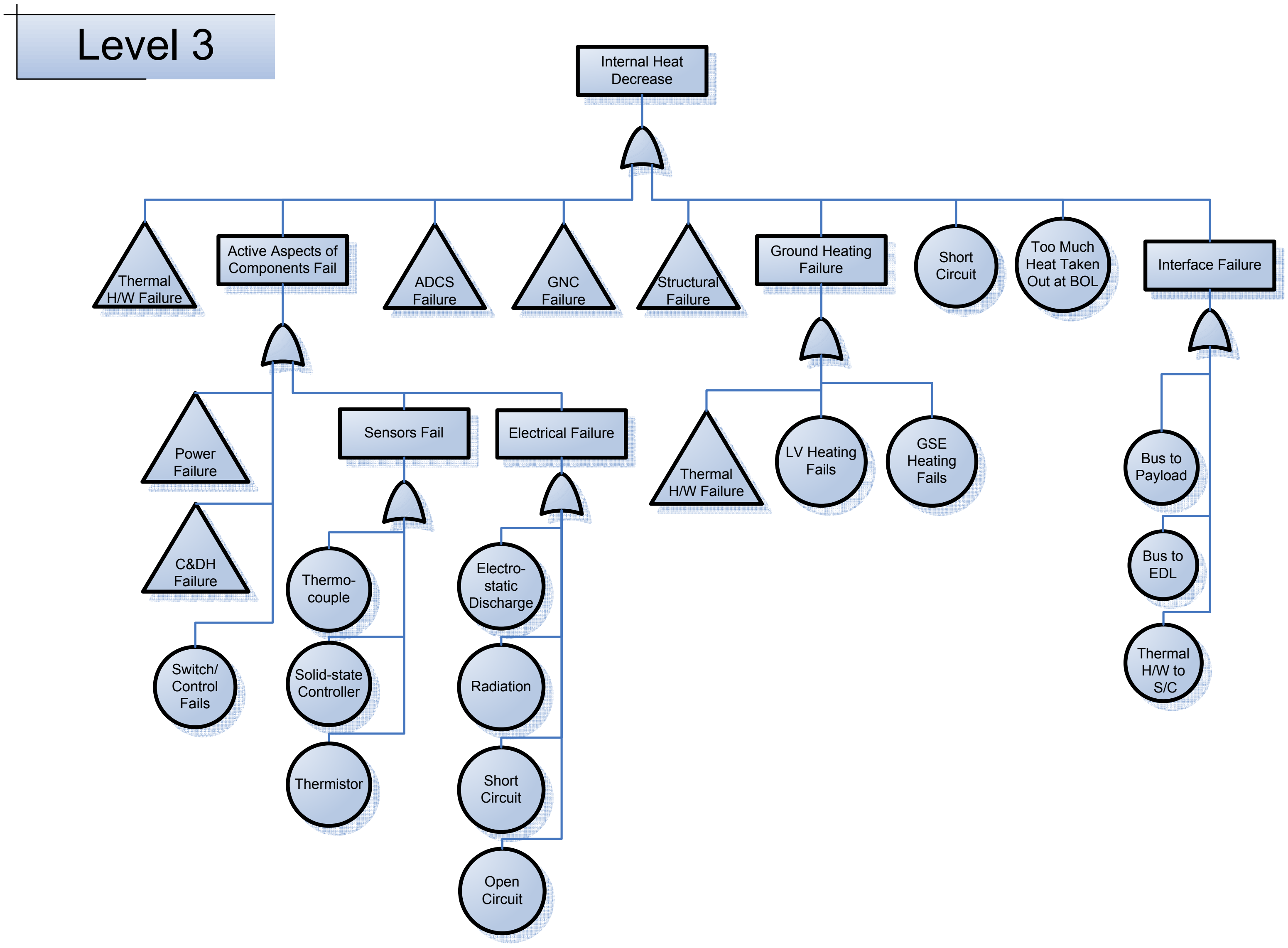


Figure 25. MLD for "Internal Heat Decrease"

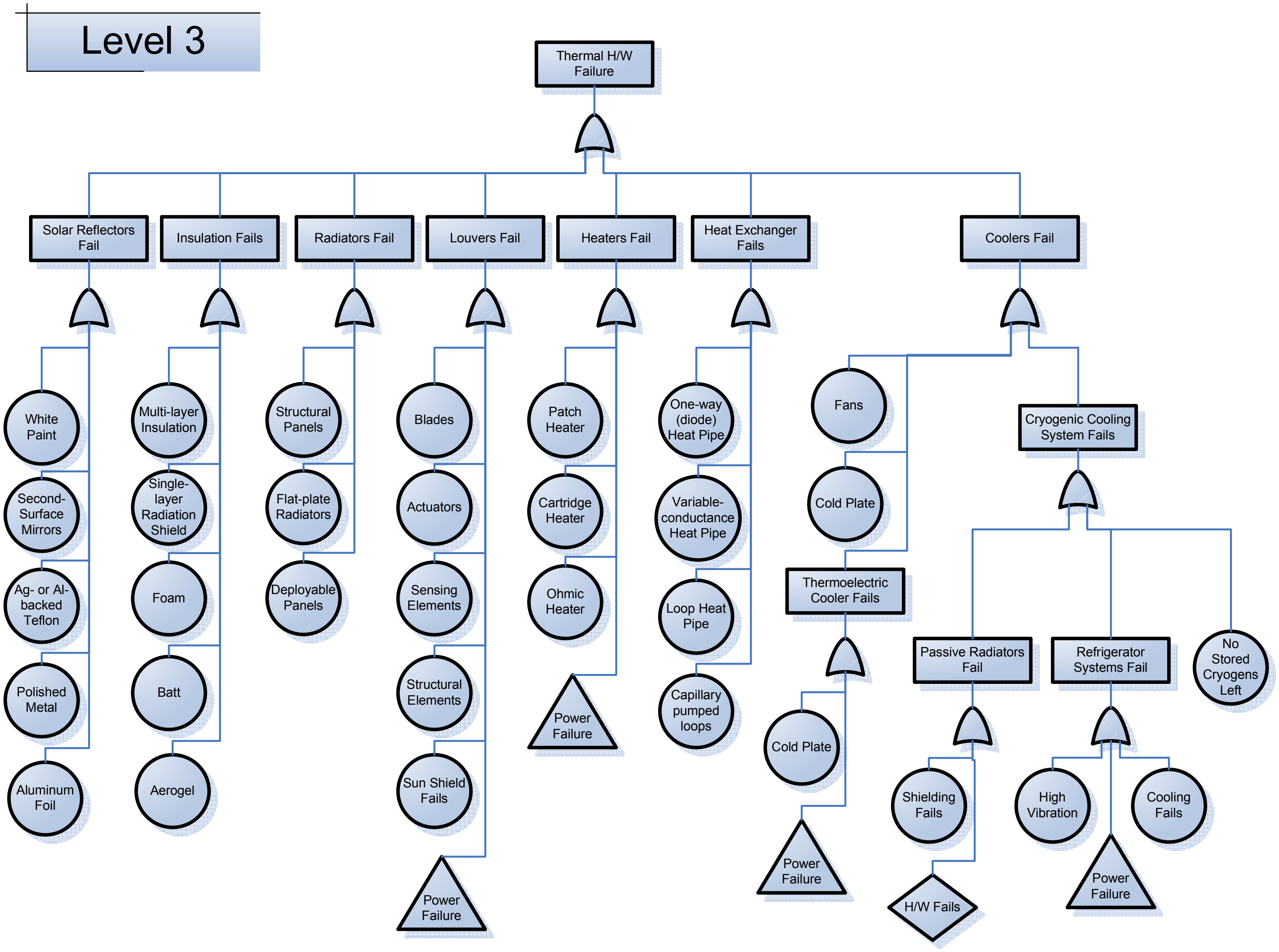


Figure 26. MLD for "Thermal Hardware Failure"

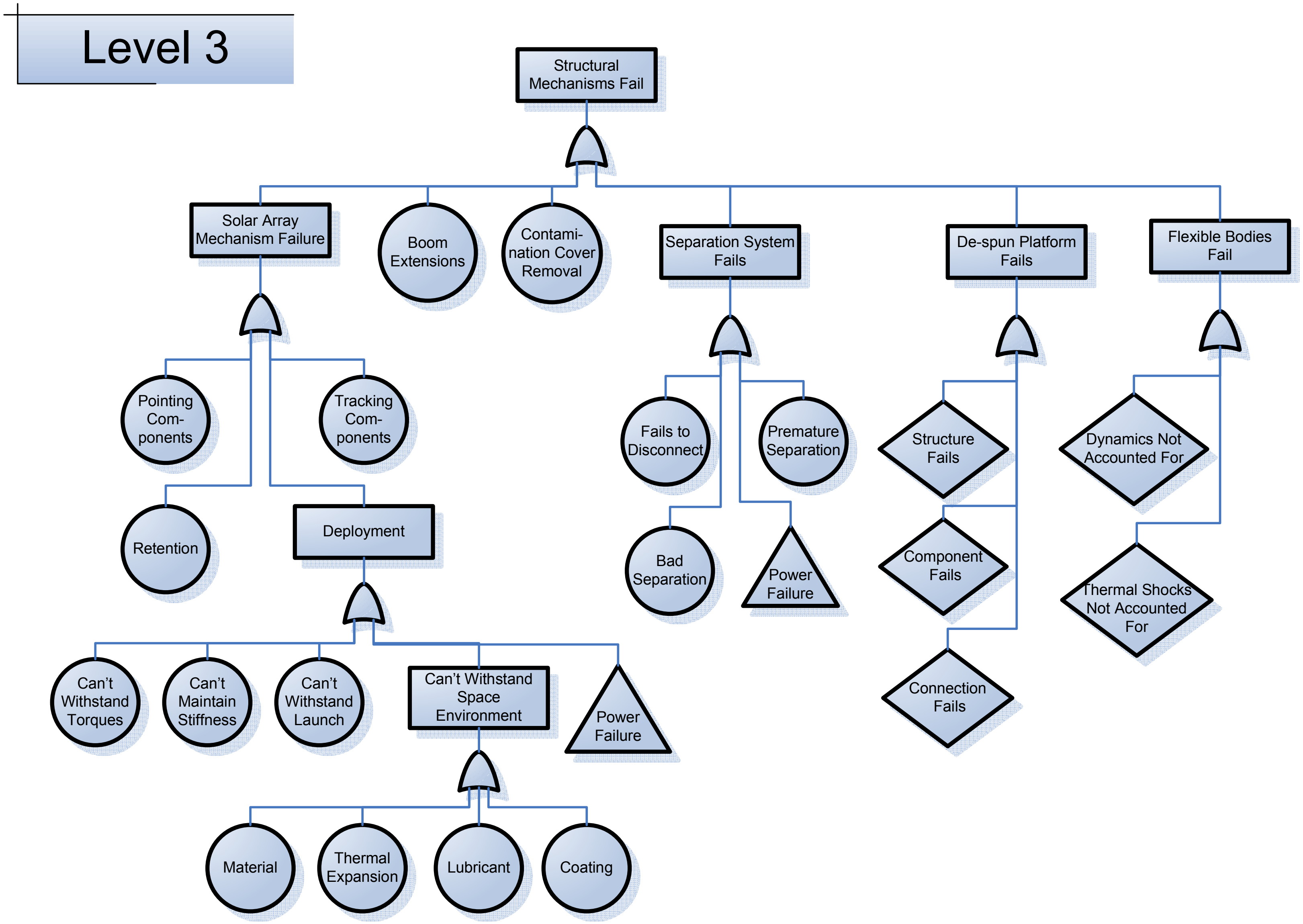


Figure 27. MLD for "Structural Mechanism Failure"

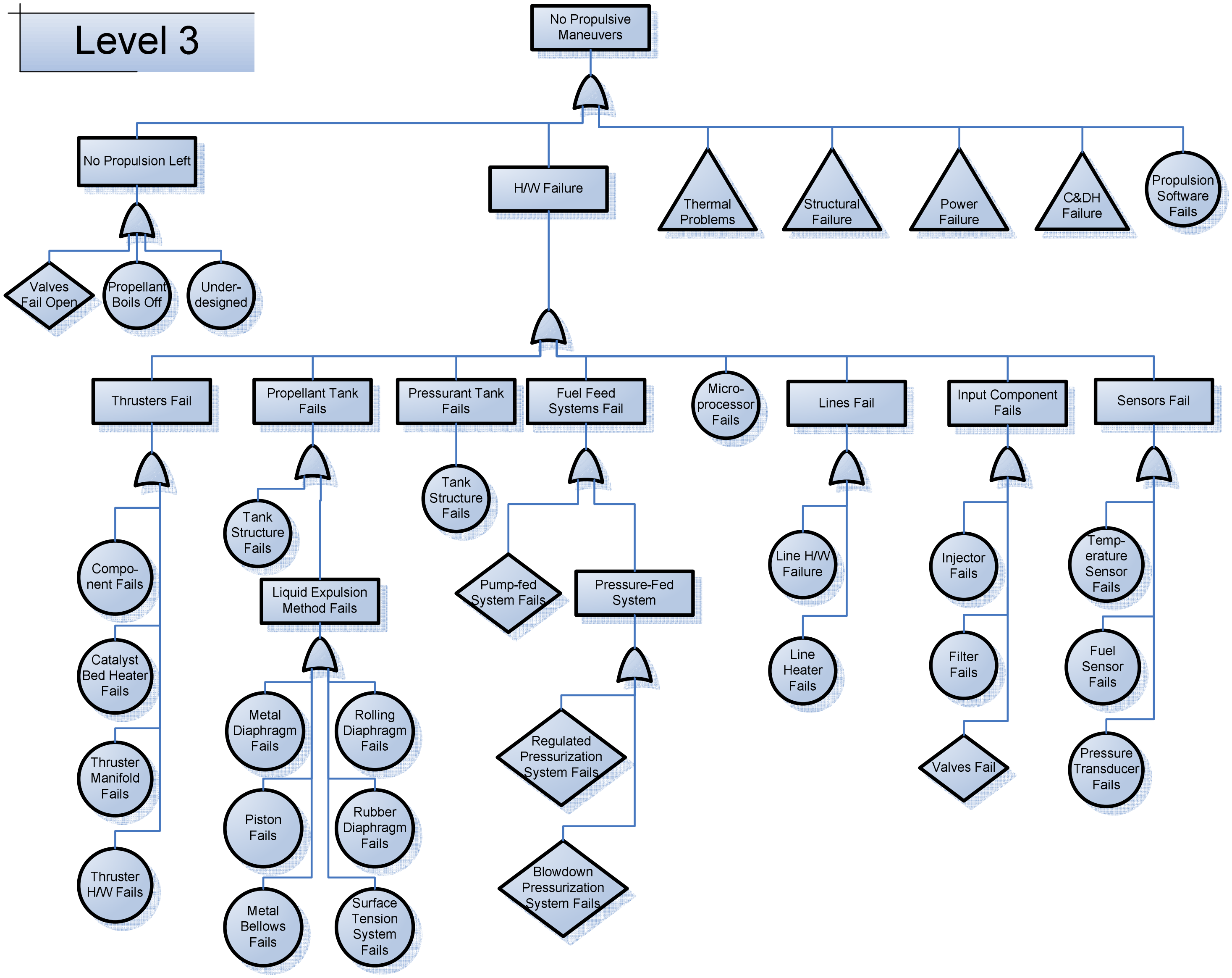


Figure 28. MLD for "No Propulsive Maneuvers"

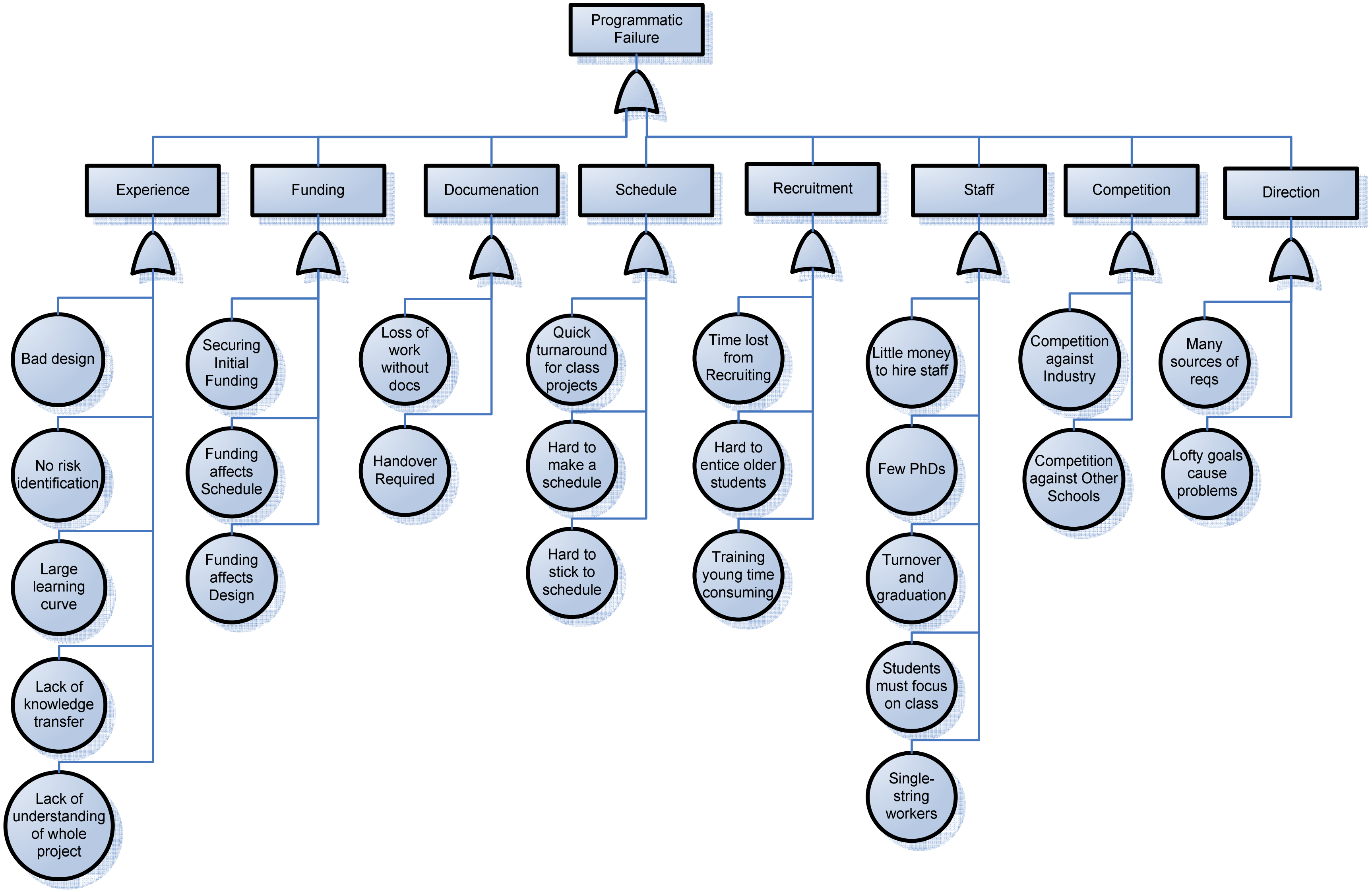

Programmatic Failure
Experience
Funding
Documenation
Schedule
Recruitment
Staff
Competition
Direction
Bad design
No risk identification
Large learning curve
Lack of knowledge transfer
Lack of understanding of whole project
Securing Initial Funding
Funding affects Schedule
Funding affects Design
Loss of work without docs
Handover Required
Quick turnaround for class projects
Hard to make a schedule
Hard to stick to schedule
Time lost from Recruiting
Hard to entice older students
Training young time consuming
Little money to hire staff
Few PhDs
Turnover and graduation
Students must focus on class
Single-string workers
Competition against Industry
Competition against Other Schools
Many sources of reqs
Lofty goals cause problems

Figure 29. MLD for "Programmatic Failures"

References

1. Pickett, Joe (ed.), The American Heritage Dictionary: Fourth Edition, Houghton Mifflin Company, Boston, 2000.
2. Rosenberg, Linda, Theodore Hammer, Albert Gallo, "Continuous Risk Management at NASA," NASA Software Assurance Technology Center, November 2005, <http://satc.gsfc.nasa.gov/support/ASM_FEB99/crm_at_nasa.html>
3. Swartout, Michael, "Twenty (plus) Years of University-Class Spacecraft: A Review of What Was, An Understanding of What Is, And a Look at What Should be Next," USU Conference on Small Satellites, Logan, Utah, August 2006.
4. Small Satellites Homepage, February 16, 2004, Surrey Satellite Technology Limited, March 22, 2007, <http://centaur.sstl.co.uk/SSHP/micro/micro2000.html>
5. Peck, Mason, Cornell University, "Risk Mitigation of Student-run Satellites at Cornell," E-mail message to Elizabeth Deems, December 12 and 14, 2005.
6. Carney, Matt, Utah State University, Personal Conversation, August 2006.
7. Donnio, Jean-Philippe, The Satellite Encyclopedia, March 2, 2007, Tag's Broadcasting Services, March 21, 2007, <http://www.tbs-satellite.com/tse/online/>
8. Satellite News Digest, Updated daily, March 14, 2007, <http://www.sat-index.com/>
9. Mission and Spacecraft Library, The Jet Propulsion Laboratory, March 14, 2007, <http://msl.jpl.nasa.gov/home.html>
10. Airclaims SpaceTrak, Airclaims, March 14, 2007, <http://www.airclaimsportal.com/Home/AirclaimsSpaceTrak.aspx>
11. Wade, Mark, Encyclopedia Astronautica, March 21, 2007, <http://www.astronautix.com/>
12. Krebs, Gunter, Gunter's Space Page, March 21, 2007, <http://www.skyrocket.de/space/>
13. NASA Lessons Learned Database, National Aeronautics and Space Administration, March 15, 2007, <http://search.nasa.gov/offices/oce/llis/home/index.html>
14. Thomsen, Michael, Michael's List of CubeSat Satellite Missions, Updated March 20, 2007, Visited March 21, 2007, <http://mtech.dk/thomsen/space/cubesat.php>

15. Operational OSCAR Satellite Status Summary, April 9, 2006, The Radio Amateur Satellite Corporation, March 21, 2007, <http://ftp.amsat.org/amsat-new/satellites/status.php>

16. Swartout, Michael, University-Class Spacecraft, August 21, 2006, Washington University in St Louis, March 21, 2007, <http://www.me.wustl.edu/faculty/mas/universityclass.htm>

17. Wikipedia, The Free Encyclopedia, March 21, 2007, <www.wikipedia.com>

18. Smith, Billy Jr., The United States Naval Academy Small Satellite Program, The United State Naval Academy, March 20, 2007, <http://web.usna.navy.mil/~brsmith/SmallSat.html>

19. Tsinghua Designed Nanosatellite NA-1 Successfully Launched, April 22, 2004, Tsinghua Univerisity News, March 19, 2007, <http://news.tsinghua.edu.cn/eng__news.php?id=528>

20. Volpe, Kyle, United States Air Force Academy, Personal conversation about FalconSat-3, April 18, 2007.

21. University Nanosatellite Program Participants, July 18, 2006, The University Nanosatellite Program, March 19, 2007, <http://www.vs.afrl.af.mil/UNP/participants.html>

22. DNEPR Launch 2: Satellite Status, Cal Poly CubeSat Program, May 8, 2007, <http://cubesat.atl.calpoly.edu/pages/missions/dnepr-launch-2/satellite-status.php>

23. Bruninga, Robert, United States Naval Academy, "Information on Power System Failures for Small Satellites," E-mail message to Elizabeth Deems, April 24, 2007.

24. Alminde, Lars, "Information on Power System Failures for Small Satellites," E-mail message to Elizabeth Deems, May 9, 2007.

25. Maier, Mark, Aerospace Corporation, "Information on Power System Failures for Small Satellites, E-mail message to Elizabeth Deems, May 8, 2007.

26. Sweeting, Sir Martin, Surrey Satellite Technology Ltd, "USA – MIT – Information from Small Satellite Conference," January 12, 2007.

27. Cheng, P., et al., "Space Systems Engineering Lessons Learned System," Proceedings of the SatMax 2002 Conference: Satellite Performance Workshop, Arlington, Virginia, April 2002, AIAA-2002-1804.

28. Sperber, R., "Hazardous Subsystems," Proceedings of the SatMax 2002 Conference: Satellite Performance Workshop, Arlington, Virginia, April 2002, AIAA-2002-1803.

29. White, J., B. Arnheim, and E. King, "Space Vehicle Life Trends," Proceedings of the 20th Aerospace Testing Seminar, Manhattan Beach, CA, March 2002.

30. Robertson, Brent and Eric Stoneking, "Satellite GN&C Anomaly Trends," 2003 AAS Guidance and Control Conference, Breckenridge, CO, AAS 03-071.

31. Mosher, Todd, et al., "Evaluating Small Satellites: Is the Risk Worth It?," Proceedings of the 13th Annual AIAA/USU Conference on Small Satellites, Logan, UT, August 2006, SSC99-IIA-1.

32. Federal Aviation Administration, "Historical Launch Data," March 7, 2007, http://www.faa.gov/data_statistics/commercial_space_data/historical_launch/

33. Chang, I-Shih. "Space Launch Vehicle Reliability," Crosslink magazine, Vol. 2, Num. 1, <http://www.aero.org/publications/crosslink/winter2001/03.html>

34. Claussen, Carl, Derek Huerta, Kyle Leveque, Jordi Puig-Suari, California Polytechnic State University, San Luis Obispo, Personal Interview, January 18, 2006.

35. Pernicka, Hank, University of Missouri-Rolla, "Questions on Risk Management for Student-run Satellites," E-mail message to Elizabeth Deems, December 12, 2005.

36. Boyd, Cameron, Queensland University of Technology, "Small Satellite Projects and Professors," E-mail message to Elizabeth Deems, February 26 and April 6, 2006.

37. Twiggs, Bob, Stanford University, "Questions on Risk Management for Student-run Satellites," E-mail message to Elizabeth Deems, December 14, 2005.

38. Michael V. Frank, "Step-by-Step Through a PRA ," Safe, Reliable, Affordable Technology Papers, November 22, 2005, <http://www.safetyfactorassociates.com/>

39. Horan, Stephen, New Mexico State University, "Small Satellite Projects and Professors," E-mail message to Elizabeth Deems, December 13, 2005.

40. Miller, Dave, Massachusetts Institute of Technology, Personal Conversation, September, 2006.

41. Bleier, Tom, Quakefinder, Personal Conversation, September, 2006.

42. Zee, Robert, University of Toronto Space Flight Lab. "Questions on Risk Management for Student-run Satellites," E-mail message to Elizabeth Deems, December 15, 2005.

43. Franke, Scott, M. Pilinski, M. Diaz-Aguado, S. Forbes, G. Hunyadi, "The University Nanosat Program from Concept to Flight: A Dual Student Program Perspective on What Works and What Does Not," Proceedings of the 20th Annual AIAA/USU Conference on Small Satellites, Logan, UT, August 2006.

44. Thomas, Pete, Air Force Research Lab, "Risk Management Info," E-mail message to Elizabeth Deems, December 8, 2005.

45. Standard Practice for System Safety (MIL-STD-882E), February 10, 2000, The Systems Safety Society and The Department of Defense, April 20, 2007, <www.system-safety.org/Documents/MIL-STD-882E-Feb05.doc>

46. Gilbert, M G, "A systems management approach to improving performance and reducing risks in research projects and programs," 2002 IEEE Aerospace Conference Proceedings, vol. 7, p. 7-3469, Big Sky, MT, March 2002.

47. Larson, Wiley J., and James R. Wertz, Space Mission Analysis and Design, 3rd edition, El Segundo, CA: Microcosm Press, 1999.

48. Sweeting, Sir Martin, Alex da Silva Curiel, Wei Sun, "The 'Personal Computer' Revolution in Space," Proceedings of the 20th Annual AIAA/USU Conference on Small Satellites, Logan, UT, August 2006.

49. Sims, Eleni, Jeffrey Zdenek, "Department of Defense Space Test Program," Small Payloads in Space, SPIE Vol. 4136, p. 57-63, November 2000.

50. Scarsfield, Liam, "The Cosmos on a Shoestring: Small Spacecraft for Space and Earth Science," RAND Corporation, March 26, 2007, <http://www.rand.org/pubs/monograph_reports/MR864/>

51. McCarthy, Michael, University of Washington Balloon Observations, “Information Regarding Balloon Experiments,” E-mail message to Elizabeth Deems, August 31, 2006 and March 26, 2007.

52. Whalen, William, “Cost of Cube Sats?,” E-mail message to Elizabeth Deems, March 25, 2007.

53. Ellis, Ryan, University of Hawai’i Rainex Balloon Program, “Information Regarding Balloon Experiments,” E-mail message to Elizabeth Deems, August 31, 2006.

54. Nelson, Matthew, Iowa State University Spacecraft Systems and Operations Lab, “Information Regarding Balloon Experiments,” E-mail message to Elizabeth Deems, September 11, 2006 and April 4, 2007.

55. “Annual CanSat Competition,” September 1, 2006, AAS/AIAA/NSGC/USRA, Visited March 26, 2007, <http://www.cansatcompetition.com/>.

56. “SPHERES,” Massachusetts Institute of Technology, September 1, 2006, <http://ssl.mit.edu/spheres/>

57. Sell, Steven, Payload Systems Incorporated, Personal Conversation via Swati Mohan, March 29, 2007.

58. Greenfield, Michael, “The Inherent Values of Probabilistic Risk Assessment,” Second NASA Probabilistic Risk Assessment Workshop, June 19, 2001, http://www.hq.nasa.gov/office/codeq/risk/uva.pdf

59. “Reliability Analysis Quick Subject Guides: Fault Trees,” ReliaSoft Corporation, August 28, 2006, <http://www.weibull.com/basics/fault-tree/index.htm>

60. Stamatelatos, Michael, Probabilistic Risk Assessment: What Is It and Why Is It Worth Performing It?, April 5, 2000, NASA Office of Safety and Mission Assurance, March 28, 2007, <http://www.hq.nasa.gov/office/codeq/qnews/pra.pdf>

61. Crow, Kenneth, Failure Modes and Effects Analysis, DRM Associates, March 28, 2007, <http://www.npd-solutions.com/fmea.html>

62. Wagner, Erika, “More about bad end states of the satellite,” E-mail message to Elizabeth Deems and personal discussions, April 10, 2006.

63. Risk Management Toolkit, April 2, 2003, MITRE Corporation, April 14, 2007, <http://www.mitre.org/work/sepo/toolkits/risk/StandardProcess/definitions/impact.html>