

# Improving the Design Process of the REgolith X-Ray Imaging Spectrometer with Model-Based Systems

Engineering

by

Mark A. Chodas

S.B., Massachusetts Institute of Technology (2012)

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

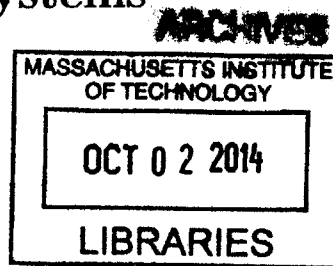
**Master of Science in Aeronautics and Astronautics**

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2014

©Massachusetts Institute of Technology 2014. All rights reserved.



Signature redacted

Author .....  
Department of Aeronautics and Astronautics  
August 21, 2014

Signature redacted

Certified by .....  
Olivier de Weck  
Professor of Aeronautics and Astronautics and Engineering Systems

Certified by .....  
Signature redacted  
Rebecca A. Masterson  
Research Engineer

Signature redacted.

Accepted by .....  
Paulo Lozano  
Chairman, Department Graduate Committee



# **Improving the Design Process of the REgolith X-Ray Imaging Spectrometer with Model-Based Systems Engineering**

by

Mark A. Chodas

Submitted to the Department of Aeronautics and Astronautics  
on August 21, 2014, in partial fulfillment of the  
requirements for the degree of  
**Master of Science in Aeronautics and Astronautics**

## **Abstract**

Traditional systems engineering processes have supported the development of many complex and successful space systems. However, some systems experience significant cost and schedule overruns. Systems engineering capabilities need to be improved to manage the expected increase in complexity of future systems. Model-based systems engineering (MBSE) is a new systems engineering paradigm where system models instead of documents are used to track requirements, describe design, support trade studies and analyses, and track verification and validation activities. The system models can be studied to expose relationships and details that are impossible to find when information is scattered across many documents and analytical models.

This thesis quantifies the advantages of MBSE over traditional systems engineering by comparing the historical development of the REgolith Imaging X-ray Spectrometer (REXIS), a student-built instrument on the OSIRIS-REx asteroid sample return mission, against a hypothetical development timeline that incorporates information from system models. The system models, constructed in SysML, capture the topological information about the system including the interfaces between all parts of the system, the uncertainty associated with each interface, and the path along which the consequences of selected design choices or requirements flow. The latter two types of information are captured with custom extensions to SysML. The models also provide useful statistics about the development process. The REXIS part count increased 104% between SRR and SDR and 163% between SDR and PDR while the interface count increased 93% between SRR and SDR and 174% between SDR and PDR. Evidence from REXIS shows that incorporating information from system models reduces design iteration and makes the design process more efficient.

Thesis Supervisor: Olivier de Weck

Title: Professor of Aeronautics and Astronautics and Engineering Systems

Thesis Supervisor: Rebecca A. Masterson

Title: Research Engineer

## Acknowledgments

This research was performed under grant number NNX13AL57H as part of the NASA Space Technology Research Fellowship Program and under contract NNG12FD70C with the NASA Goddard Space Flight Center (GSFC) as part of the REXIS program. I would like to thank the NSTRF Program and the REXIS program for their support.

I would like to thank the past and present members of the REXIS team - Kevin, Matt, Hal, Niraj, Harrison, Eric, James, David, and Mike - for all of their hard work over the years. It has been and continues to be a pleasure working with you guys. Someday we'll start to know what we're doing.

I would also like to thank all of the people who have mentored me on REXIS and in my research. Thank you to the REXIS leadership - Becky Masterson, Rick Binzel, Josh Grindlay, and Dave Miller - for guiding us through the design of a space instrument and thank you to my research mentors - Olivier de Weck, Art Ferrer, and Darryl Lakins - for providing feedback on my thesis and forcing me to dig deeper into the system models.

Finally, I'd like to thank my family for their love and support. I promise to try to come home more often.

# Contents

<b>List of Figures</b>	<b>9</b>
<b>List of Tables</b>	<b>13</b>
<b>1 Introduction</b>	<b>15</b>
1.1 Background . . . . .	16
1.1.1 Traditional Systems Engineering vs. Model-Based Systems Engineering	17
1.2 Motivation . . . . .	18
1.3 Literature Review . . . . .	20
1.3.1 Benefits of Systems Engineering . . . . .	21
1.3.2 Benefits of Model-Based Systems Engineering over Traditional Systems Engineering . . . . .	22
1.3.3 Applications of Model-Based Systems Engineering . . . . .	23
1.3.4 Change Propagation Research . . . . .	25
1.3.5 Literature Gap . . . . .	26
1.4 Thesis Objectives . . . . .	27
1.5 Thesis Roadmap . . . . .	27
<b>2 Methodology</b>	<b>29</b>
2.1 System Model Description and Contents . . . . .	30
2.1.1 The Systems Modeling Language (SysML) . . . . .	30
2.1.2 SysML Model Description . . . . .	31
2.2 Inspecting the System Models . . . . .	38

2.3	Constructing and Comparing the Design Timelines . . . . .	39
2.4	Limitations . . . . .	39
<b>3</b>	<b>REXIS Design History Overview</b>	<b>41</b>
3.1	REXIS Instrument Overview . . . . .	41
3.2	REXIS Instrument Architecture . . . . .	42
3.2.1	REXIS Requirements . . . . .	43
3.3	REXIS Design History . . . . .	45
3.3.1	REXIS Design at System Requirements Review (SRR) . . . . .	45
3.3.2	REXIS Design at System Definition Review (SDR) . . . . .	46
3.3.3	REXIS Design at Preliminary Design Review (PDR) . . . . .	47
3.3.4	REXIS Design at Critical Design Review (CDR) . . . . .	49
3.4	SysML Models of REXIS at Milestone Reviews . . . . .	50
3.4.1	SysML Representation of SRR Design . . . . .	50
3.4.2	SysML Representation of SDR Design . . . . .	51
3.4.3	SysML Representation of PDR Design . . . . .	52
3.4.4	Design History Statistics . . . . .	52
3.5	Limitations of using REXIS as a Case Study . . . . .	55
<b>4</b>	<b>Improving the Design Process with Interface Uncertainty</b>	<b>57</b>
4.1	Subassembly Interface Uncertainty Trends . . . . .	57
4.2	Interface Uncertainty as a Predictor of Future Subassembly Change . . . . .	59
4.3	Interface Uncertainty as a Predictor of Final Subassembly Size . . . . .	60
4.4	Summary . . . . .	63
<b>5</b>	<b>Improving the Design Process by Tracing Design Consequences</b>	<b>65</b>
5.1	Design Evolution of the REXIS Thermal System . . . . .	66
5.1.1	Historical Design Evolution . . . . .	66
5.1.2	Model-Based Design Evolution of the Thermal System . . . . .	69
5.2	Design Evolution of the REXIS Radiation Cover . . . . .	74
5.2.1	Historical Design Evolution . . . . .	74

5.2.2	Model-Based Design Evolution of the Radiator Cover . . . . .	78
<b>6</b>	<b>Conclusion</b>	<b>83</b>
6.1	Thesis Summary . . . . .	83
6.2	Generalization of Results . . . . .	83
6.3	Contributions and Future Work . . . . .	84
	<b>Bibliography</b>	<b>85</b>
<b>A</b>	<b>REXIS Requirements</b>	<b>89</b>
A.1	REXIS Measurement Requirements . . . . .	89
A.1.1	Level 2 Measurement Requirements . . . . .	89
A.1.2	Level 3 Measurement Requirements . . . . .	90
A.2	REXIS Level 4 Requirements . . . . .	93
A.2.1	Level 4 Detector Requirements . . . . .	93
A.2.2	Level 4 Electronics Requirements . . . . .	96
A.2.3	Level 4 Structures Requirements . . . . .	99
A.2.4	Level 4 Flight Software Requirements . . . . .	101





# List of Figures

1-1	The systems engineering engine from the NASA Systems Engineering Handbook [1]. . . . .	17
1-2	Trend of cost overruns in major weapons systems against time . . . . .	19
1-3	Cost and schedule overruns of selected NASA missions . . . . .	19
1-4	Relationship between successful and failed mission in terms of complexity and cost and schedule overrun . . . . .	20
1-5	Systems engineering effort against cost and schedule overruns . . . . .	22
1-6	Effect of implementing model-based systems engineering on the number of defects in requirements documents . . . . .	23
1-7	Thesis roadmap . . . . .	27
2-1	SysML Diagram types . . . . .	31
2-2	An example block definition diagram (a) and internal block diagram (b) showing the basic SysML topological modeling features. . . . .	33
2-3	Custom interface block abstraction hierarchy . . . . .	35
2-4	Custom association block abstraction hierarchy . . . . .	36
2-5	Example of a Digital Data interface . . . . .	37
2-6	Example of design consequence tracing . . . . .	38
3-1	The REXIS requirements hierarchy . . . . .	44
3-2	The dates of the key milestone reviews for REXIS. . . . .	45
3-3	The REXIS design at SRR as represented in CAD (a) and in a schematic view that shows more of the design details (b). . . . .	46

3-4	The exterior of the REXIS design at SDR. . . . .	47
3-5	A cutaway view of the REXIS design at SDR showing the radiation cover and detectors inside the instrument. . . . .	48
3-6	The REXIS design at PDR. . . . .	49
3-7	The REXIS Spectrometer design at CDR. . . . .	50
3-8	The REXIS SXM design at CDR. . . . .	51
3-9	Number of parts per subassembly at each milestone review . . . . .	53
3-10	Number of ports per subassembly at each milestone review . . . . .	54
3-11	Number of parts per subassembly plotted against the number of ports per subassembly . . . . .	54
3-12	Number of ports per part at each milestone review . . . . .	55
4-1	The percentage of uncertain interfaces in each subassembly at the key milestone reviews. . . . .	58
4-2	The increase in the number of parts and ports for each subassembly between SRR and SDR plotted against the interface uncertainty for each subassembly at SRR. . . . .	60
4-3	The increase in the number of parts and ports for each subassembly between SDR and PDR plotted against the interface uncertainty for each subassembly at SDR. . . . .	60
4-4	The increase in the number of parts and ports for each subassembly between SRR and PDR plotted against the interface uncertainty for each subassembly at SRR. . . . .	61
4-5	The number of parts and ports for each subassembly at SDR plotted against the interface uncertainty for each subassembly at SRR. . . . .	62
4-6	The number of parts and ports for each subassembly at PDR plotted against the interface uncertainty for each subassembly at SDR. . . . .	62
4-7	The number of parts and ports for each subassembly at PDR plotted against the interface uncertainty for each subassembly at SRR. . . . .	62
5-1	The design of the thermal system at SDR . . . . .	67

5-2	The design of the thermal system at PDR . . . . .	68
5-3	The design of the thermal system at CDR . . . . .	69
5-4	Historical development timeline of the REXIS thermal system . . . . .	69
5-5	Design of the REXIS thermal system at SRR as represented in an Internal Block Diagram . . . . .	71
5-6	Design of the REXIS thermal system at SDR as represented in an Internal Block Diagram . . . . .	73
5-7	Model-based development timeline for the REXIS thermal system . . . . .	74
5-8	The design of the Radiation Cover at SDR . . . . .	75
5-9	The design of the Radiation Cover at PDR . . . . .	76
5-10	The design of the Radiation Cover at CDR . . . . .	77
5-11	Historical development timeline for the REXIS Radiation Cover . . . . .	78
5-12	Design of the REXIS Radiation Cover at SDR as represented in an Internal Block Diagram . . . . .	79
5-13	Design of the REXIS Radiation Cover at PDR as represented in an Internal Block Diagram . . . . .	81
5-14	Model-based development timeline for the REXIS Radiation Cover . . . . .	82



# List of Tables

3.1	Summary of the number of Level 3 and Level 4 requirements at each milestone review. . . . .	44
3.2	Summary of the number of parts and ports at each milestone review. . . . .	54



# Chapter 1

## Introduction

Systems engineering is a core discipline of aerospace engineering concerned with the development of an operational system capable of meeting requirements within constraints [1]. Aerospace systems utilize systems engineering to ensure that requirements are properly determined, interfaces are defined, and the system is verified and validated before operation. The systems engineering processes defined in the NASA Systems Engineering Handbook [1] and the INCOSE Systems Engineering Handbook [2] have enabled the successful development and deployment of extremely complex air and space systems. Yet, cost and schedule overruns are becoming more common in the aerospace industry, hinting that system complexity is beginning to overwhelm the existing systems engineering processes [3].

A new systems engineering paradigm called model-based systems engineering (MBSE) offers a way to expand systems engineering capabilities so that complexity can be managed more effectively and systems can be delivered on time and on budget. MBSE is the formalized application of modeling to support system requirements, design, analysis, verification and validation, beginning in the conceptual design phase and continuing throughout development and later life cycle phases [4]. In contrast to traditional, document-based methods, MBSE houses system information in a descriptive model that becomes the primary artifact of the systems engineering process. MBSE presents several potential advantages over traditional systems engineering including improved communication and improved ability to manage complex systems [4].

This thesis will investigate whether applying model-based systems engineering in the

design process makes the design process more efficient. Building on change prediction approaches, models at the key points in the design process of a previously designed system are created and inspected to extract information that could have assisted in the development process. Using that information, a hypothetical design timeline is constructed and compared with the historical design timeline to determine if applying MBSE to the design process results in a more efficient design process. Case studies from the development of the REgolith Imaging X-ray Spectrometer (REXIS), a student-built instrument on NASA's OSIRIS-REx asteroid sample return mission, are presented. REXIS is a coded-aperture imaging X-ray spectrometer that will categorize the asteroid among the known meteorite groups based on elemental ratios and map the elemental distribution in the asteroid regolith. REXIS is being developed by the MIT Space Systems Laboratory with partners including the MIT Earth, Atmospheric, and Planetary Sciences Department and the Harvard College Observatory. This chapter provides the background, motivation, literature review, and thesis objective for this research.

## 1.1 Background

This section provides background on systems engineering and contrasts traditional and model-based system engineering. System engineering is a methodical, disciplined approach for the design, realization, technical management, operations, and retirement of a system. Systems engineering seeks a balanced design in the face of multiple, opposing interests and constraints. A systems engineer develops a system architecture and requirements, evaluates design tradeoffs, manages risks, defines and assesses interfaces, and oversees verification and validation activities. The NASA Systems Engineering Handbook presents a systems engineering engine, shown in Figure 1-1, that describes the responsibilities of the systems engineer at a given level in the design [1]. As the system is being designed, the systems engineer decomposes higher level requirements into lower level requirements and flows those requirements to lower level parts of the system. As the system is being realized, the systems engineer oversees the verification and validation process and passes the results from lower levels up to higher levels of the system. The systems engineering engine is recursive in that



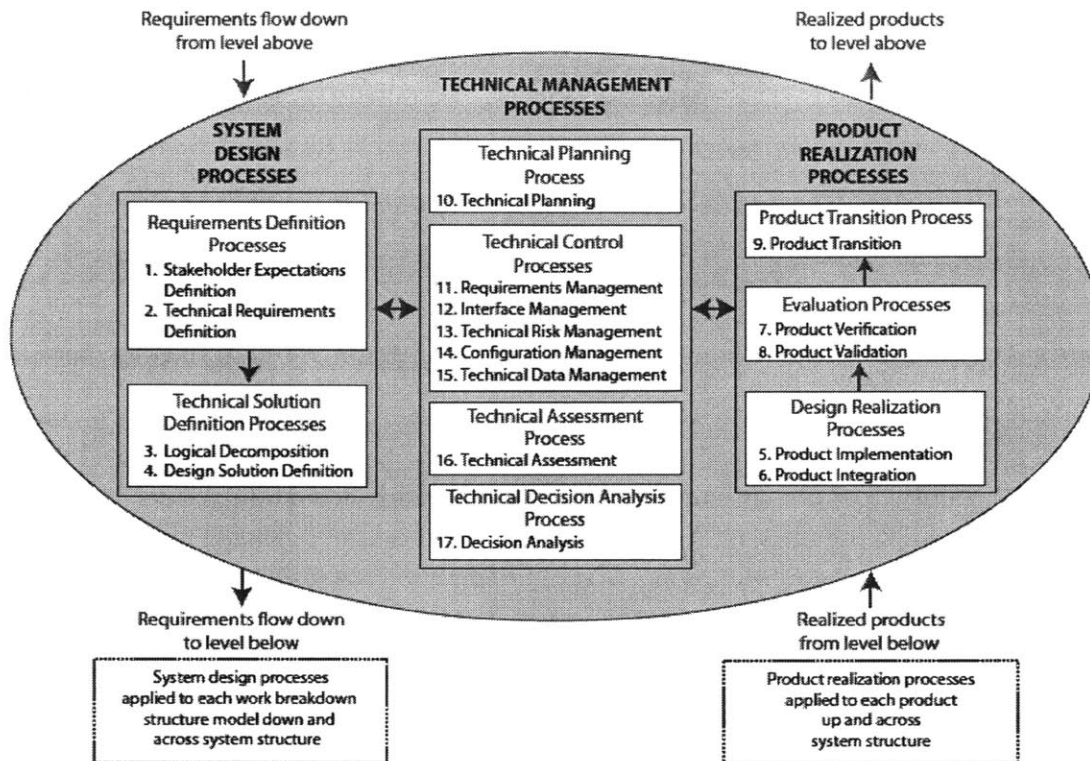


Figure 1-1: The systems engineering engine from the NASA Systems Engineering Handbook [1].

it is applied to higher-level products of the engine during design and to lower level products of the engine during implementation.

### 1.1.1 Traditional Systems Engineering vs. Model-Based Systems Engineering

The primary difference between traditional systems engineering and model-based systems engineering is the use of a descriptive model to capture information about the system, instead of capturing system information in documents. In traditional systems engineering, requirements are captured in requirements documents, verification information is captured in verification plans, and the link between requirements and their verification activities is captured in a requirements verification matrix. Behavioral information is kept in functional flow block diagrams, risks are kept in a risk matrix, and interface information is captured in Interface Control Documents (ICDs). The NASA Systems Engineering Handbook lists 44

different types of plans that describe and capture information about the system [1]. Keeping the information in the different documents consistent and up to date is an overwhelming task. Model-based systems engineering seeks to overcome this hurdle by centralizing all information about the system in one descriptive model. Potential benefits of using model-based systems engineering include [4]:

- Improved communications among the development team by eliminating inconsistent information
- Increased ability to manage complex systems by using a system model to gain additional insight into the system under development
- Improved quality by evaluating the system model for consistency, correctness, and completeness
- Enhanced knowledge capture and reuse of information through standardized modeling techniques

## 1.2 Motivation

Cost and schedule overruns are common in today's aerospace systems. Figure 1-2 is from an analysis of weapons systems acquired by the Department of Defense over the past twenty years. It shows the average cost overrun for each year between 1993 and 2007. From the slope of the best fit line, the average cost overrun is increasing by about 1.9 percentage points per year [3]. Similarly, NASA spacecraft commonly experience cost and schedule overruns. Figure 1-3 is from a study of forty NASA missions from the Mars Surveyor and Exploration, Discovery, Medium and Small Explorer, and other programs [5]. It shows the ratio of actual schedule to nominal schedule on the vertical axis plotted against the ratio of actual cost to nominal cost on the horizontal axis. Note that few missions meet either their cost or schedule targets. Most missions experience cost and schedule overruns, up to double the nominal schedule or cost. The average cost overrun is 27% and the average schedule overrun is 22%.

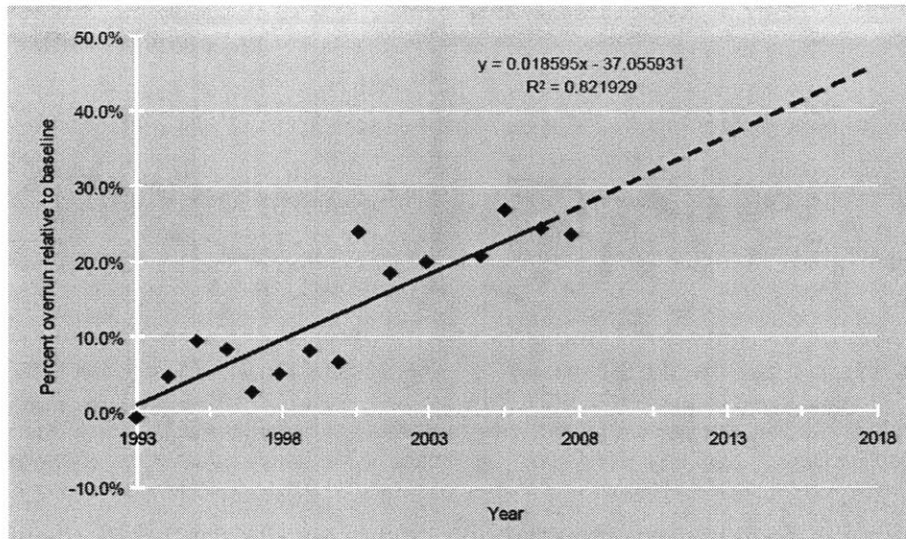


Figure 1-2: Cost overruns in major weapons systems have been getting progressively worse over the past two decades. The slope of the best fit line indicates that the average cost overrun increases by about 1.9 percentage points per year [3].

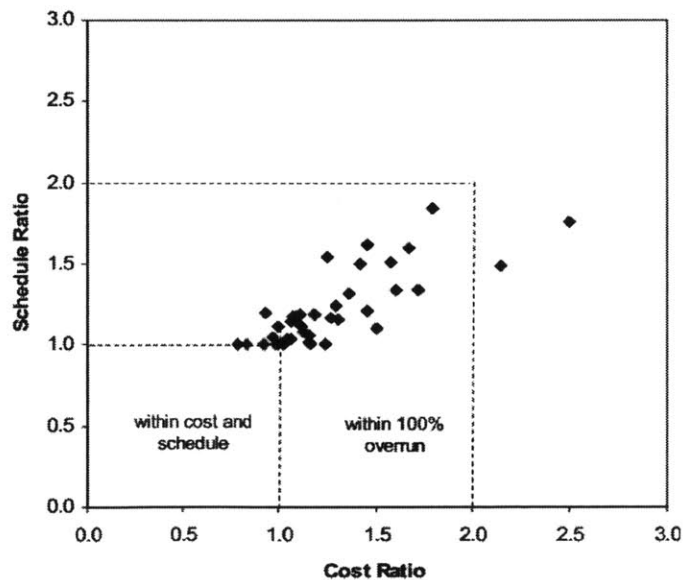


Figure 1-3: A plot of the ratio of actual cost to initial cost estimate on the horizontal axis and actual schedule to initial schedule estimate on the vertical axis for selected NASA missions between 1992 and 2007. The average cost overrun is 27% and the average schedule overrun is 22% with cost and schedule overruns being correlated. [5]

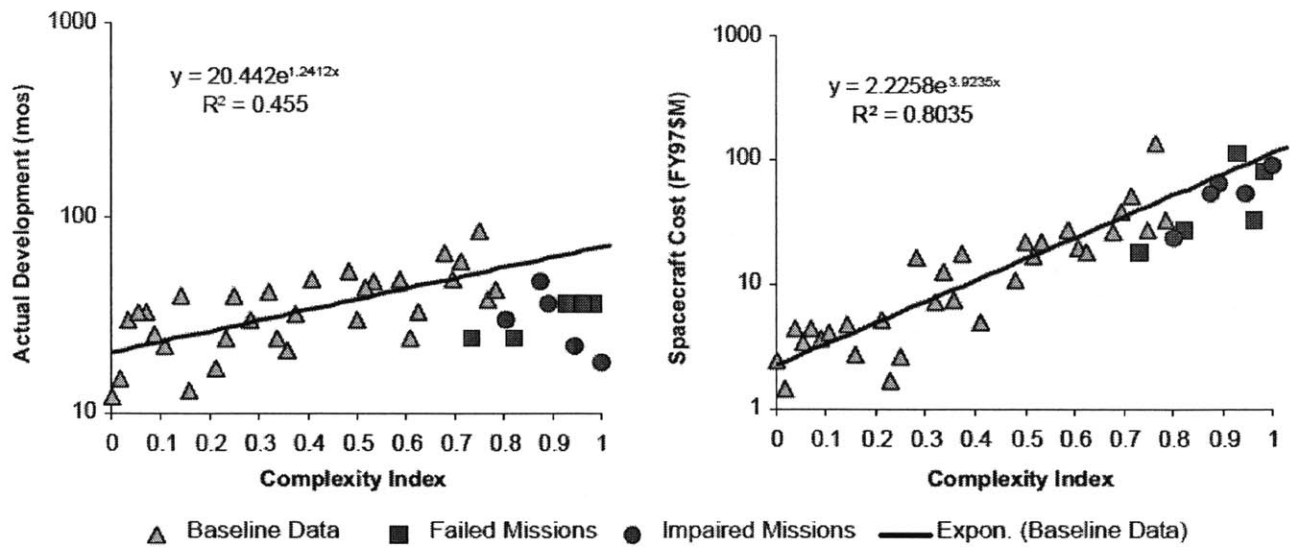


Figure 1-4: Plots of how failed and impaired missions relate to a sample of all space missions in terms of cost, schedule, and complexity. Failed and impaired missions tend to be more complex than average, yet have shorter schedules and tighter budgets than typical of project of their complexity [6].

Authors have hypothesized that system complexity and the struggle to manage that complexity are driving cost and schedule overruns and mission success. In a study of weapons acquisition programs, the technical complexity of today’s systems is shown to be responsible for about one third of the average cost overrun [3]. Additional evidence is found in a study of NASA’s “Faster, Better, Cheaper” missions that examined how complexity, schedule, and cost interacted to produce either successful or unsuccessful missions. As shown in Figure 1-4, failed or impaired mission tend to be more complex and have shorter schedules and smaller budgets than the average space mission [6]. From the available evidence, the complexity of a system is a major factor in determining cost and schedule overruns and mission success.

### 1.3 Literature Review

This section reviews the literature on the benefits of systems engineering, the evidence that model-based engineering is superior to traditional systems engineering, and the current applications of model-based systems engineering. Also, change propagation and prediction research that has looked into ways to improve the design process is discussed. The section

concludes with the gap in the literature that this thesis will help to fill.

### 1.3.1 Benefits of Systems Engineering

There is evidence in the literature that systems engineering can reduce system cost and schedule overruns. Figure 1-5 presents data from an INCOSE survey that shows that increased systems engineering effort decreased cost and schedule overruns [7]. Systems engineering effort is defined as the product of systems engineering quality and the fraction of total cost used for systems engineering. Systems engineering quality is a subjective evaluation of the quality of the systems engineering on each project provided by the respondents. The figure shows the systems engineering effort on the horizontal axis with the cost or schedule overrun of each project on the vertical axis. The solid best fit line and the dashed 90% confidence lines are shown on the figure. The optimal systems engineering effort, where cost and schedule overruns are minimized, is around 15%. Adding more systems engineering effort beyond 15% does not result in any further reduction in cost or schedule overrun.

A survey of government contractors within the Systems Engineering Division of the National Defense Industrial Association showed that projects with better systems engineering delivered higher performance [8]. The survey measured the systems engineering effectiveness by asking respondents if their project had performed tasks from a list of systems engineering work products from the Capability Maturity Model Integration®(CMMI) Systems Engineering/Software Engineering model. Project performance was measured using a number of factors including earned value data, percent of requirements satisfied, changes in budget or schedule, and perception of customer satisfaction. The study also gathered data on project difficulty. The data showed that projects that applied systems engineering best practices were more successful than projects that did not. Additionally, the improvement due to applying systems engineering best practices was more pronounced in more challenging projects than in less challenging projects, suggesting that systems engineering becomes more important as projects become more complex.

Within the aerospace context, a study on six aerospace projects again found that increasing system engineering quality reduced cost and schedule overruns [9]. The study defined the quality of systems engineering as the fraction of systems engineering tasks performed

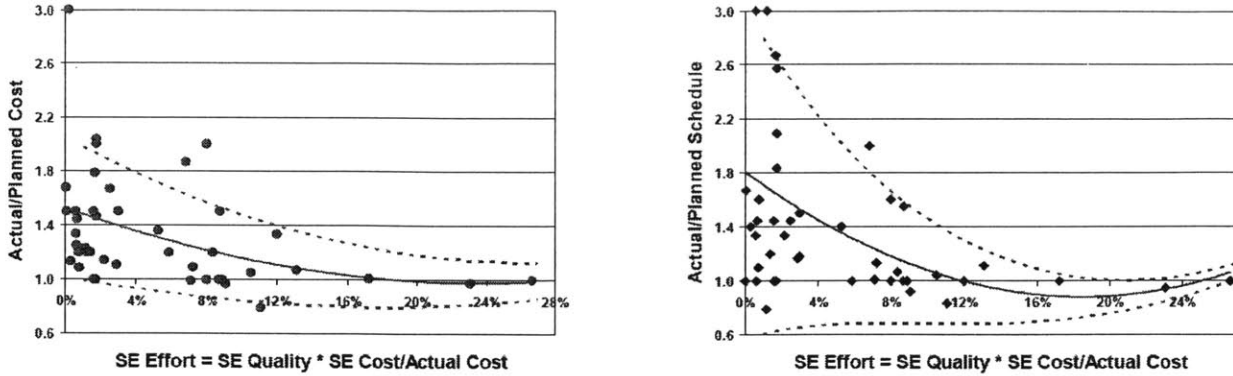


Figure 1-5: Plots showing that increased systems engineering effort can decrease cost overruns and schedule overruns. Systems engineering effort is defined as the product of systems engineering quality, a subjective evaluation of the quality of the systems engineering on each project provided by the respondents, and the fraction of total cost used for systems engineering. The figure shows the systems engineering effort on the horizontal axis with the cost or schedule overrun of each project on the vertical axis. The solid best fit line and the dashed 90% confidence lines are shown on the figure. The optimal systems engineering effort is around 15%, beyond which the benefits of systems engineering do not increase [7].

as part of each project divided by the total number of systems engineering tasks on a list compiled by the authors. The level of success for each project was measured using a scoring system that evaluated the performance of each project with respect to cost, schedule, and technical quality. The study found that projects that completed a higher fraction of systems engineering tasks performed better with regards to both cost and schedule.

### 1.3.2 Benefits of Model-Based Systems Engineering over Traditional Systems Engineering

Currently, there aren't many papers with quantitative evidence that introducing model-based systems engineering has benefits over traditional systems engineering processes. One Raytheon study of errors in requirements specifications found that errors were reduced from 1.05 defects per shall statement to 0.33 defects per shall statement when model-based systems engineering was introduced to the requirements specification writing process [10]. That data is shown in Figure 1-6. Data that explores the advantages and disadvantages of model-based systems engineering versus will benefit the field.

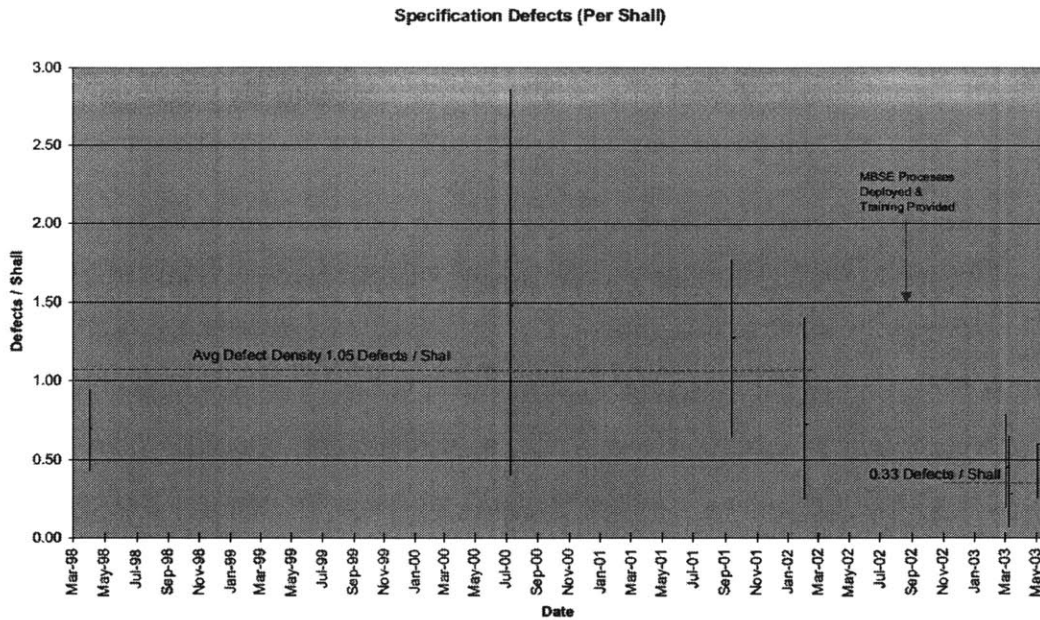


Figure 1-6: The number of defects per shall statement within a requirements specification on the vertical axis against time on the horizontal axis. MBSE processes were introduced in September 2002, leading to a 68% percent reduction in specification errors per shall statement.[10]

### 1.3.3 Applications of Model-Based Systems Engineering

In the literature, most applications of model-based systems engineering fall into three general categories: requirements engineering, system description, and integration with external analysis tools. As design can be thought of as a series of decisions that move a system from concept to operations, the general research focus is on using model-based systems engineering to improve decision making in the design process [11, 12]. The following sections will give an overview of the work done in each area.

#### Requirements Engineering

The support for traceability among requirements and from requirements to design in model-based systems engineering is an improvement over the current document-based methods of capturing traceability. Bernard explored the capability of system models to directly model requirements as constraints on properties of the model instead of as text statements. Integrating requirements into the model improves clarity and simplifies verification [13]. Soares uses system models to enhance the description of text-based requirements, including catego-

ricing and grouping requirements [14]. Russell presents a methodology that uses requirements and verification traceability to design to support decision making during design [15].

## **System Description**

Other papers have shown that model-based systems engineering can capture the structure and behavior of complex systems. A study of NASA's space communications networks found that the integration of several different communications networks was significantly improved by capturing the details of each system in a system model as opposed to in documents. In particular, the ability of model-based systems engineering to capture the complexity of the system and allow analyses of different architectural options was an improvement over document-based practices [16]. A team at NASA's Jet Propulsion Laboratory (JPL) used model-based systems engineering to assist in the design of the Europa Clipper. Model-based systems engineering was able to describe the system under development and capture many of the traditional products used in mission concept development including mission scenarios, the master equipment list, the powered equipment list, and cost models. The team attributed their ability to evaluate three different architectural concepts within a time normally sufficient for evaluating only one or two different options to the application of model-based systems engineering [17, 18]. Other authors have abstracted the description of a single spacecraft to a description of a spacecraft in general. The INCOSE Space Systems Working Group has developed a general architectural framework for CubeSats. The framework can be used by any team that desires to build a CubeSat, saving development time that would have otherwise be spent on architecture generation [19].

## **Integration with Analysis Tools**

Once system information is captured in a descriptive model, it can be extracted from that model to be used in external analysis engines, and the results returned to the model. Previous work has focused on integrating advanced dynamics analysis capabilities within an MBSE context. In one paper, the authors customized SysML to support modeling of mechatronic systems with complex discrete and continuous dynamics [20]. The system behavior was initially described in SysML and then, using a triple-graph grammar based transforma-



tion, input into an external analysis tool that simulated the dynamics of the system. The transformation is bi-directional so that the output of the model can be transferred back into the SysML model. Using this scheme, trade studies that rely on information on the dynamic behavior of the system can be performed in SysML with seamless analysis of alternative designs using external tools. Other authors have integrated model-based systems engineering tools with analysis and optimization environments that already support the integration of a wide range of CAD, analysis, and optimization tools. Both Lockheed Martin and JPL have explored the integration of SysML with Phoenix ModelCenter®. Through ModelCenter, information in the system model can be used in external tools such as Satellite Tool Kit® and MATLAB® to support trade studies and verification [21, 22].

### 1.3.4 Change Propagation Research

The field of change propagation utilizes models to improve the designer's ability to predict and manage change in a system. Change propagation research focuses on the system interfaces as those are the pathways through which changes can propagate through the design. Giffin et al. studied a large change database and found that some parts of the system acted as absorbers of change while others acted as multipliers of change [23]. However, they did not apply their approach to a system under development.

While most change propagation research is focused on analyzing changes to a fully developed and operational system, some authors are looking into using change prediction to improve the design process [24]. Fricke et al. enumerated the methods for coping with change: prevention, front-loading, effectiveness, efficiency, and learning [25]. The prevention strategy aims to stop design changes that can be avoided, and thereby prevent unimportant changes from causing design churn. The front-loading strategy aims to detect potential changes as early as possible so that the changes to the system don't undo previously made design decisions. The effectiveness strategy limits changes to only those that pass a cost-benefit analysis. Changes that are not required but make the design better are not permitted. The efficiency strategy aims to process changes in an efficient manner so that they do not overwhelm the design process. Finally, the learning strategy continuously studies previously made changes so that those changes can be initially incorporated into future designs.

Methods of finding areas of the design that are likely to change have been studied. Clarkson, Simons, and Eckert presented an approach for change prediction that applied likelihoods and consequences to all potential change propagation paths that were identified using a Design Structure Matrix (DSM) [26]. Therefore, the chance that change in one subsystem would propagate to a different subsystem could be calculated. Another study by Morkos and Summers used DSMs to trace the impact of requirements changes on the design [27].

### 1.3.5 Literature Gap

There are two gaps in the current literature: applying model-based systems engineering to change prediction to improve the design process and providing evidence that model-based systems engineering is superior to traditional systems engineering. Building on previous change prediction work, the topological information captured in system models can be used to improve design decisions. In most model-based systems engineering work, the traceability from requirements to other requirements or the design has been explored, but the interfaces between parts of a system have received less attention. In traditional systems engineering, this information is spread among CAD, electrical schematics, and analytical models making it difficult to conduct a rigorous examination of all of the system interfaces. Therefore, extracting information to support design decisions from topological system models is an area where model-based systems engineering is expected to have a significant advantage over traditional systems engineering. In order to have improved the system design process, the extracted information needs to uncover issues with the design that were not known at the time or steer the design in the correct direction before the design historically moved in that direction. In other words, it is a change prediction problem. This thesis will contribute to both literature gaps by extracting decision support information from system model topological information and showing that using this additional information could have made the REXIS design process more efficient.

## 1.4 Thesis Objectives

This thesis hypothesizes that applying model-based systems engineering to the REXIS design process would have resulted in a savings of time and effort. This hypothesis will be tested by examining system models of the historical REXIS design at the key milestone reviews and extracting information about the design that was not known at the time as the models were unavailable. The system models contain the interface information as it was known at the time, information about the uncertainty associated with each interface, and the interfaces along which the consequences of selected design decisions and requirements flow. Using this information, an alternative design timeline is constructed that reflects how the REXIS design process might have gone if the system models had been available. The hypothetical model-based design timeline is compared against the historical design timeline to determine how much faster the model-based timeline might have been. If the model-based timeline is faster than the historical timeline, then the hypothesis will be confirmed.

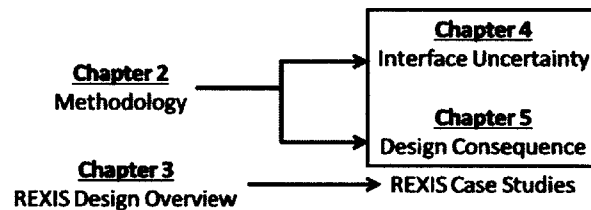


Figure 1-7: Thesis roadmap

## 1.5 Thesis Roadmap

This thesis is organized into five chapters. Figure 1-7 shows the thesis roadmap. Chapter 1 is the thesis introduction, including background, motivation, literature review, and thesis objectives. Chapter 2 explains the methodology used to accomplish the thesis objectives. Chapter 3 describes the REXIS design evolution to provide context for the case studies based upon the REXIS experiences. Chapter 3 also presents statistics from the SysML models of the REXIS design at each milestone review. Chapter 4 presents the investigation into the ability of interface uncertainty to predict future design changes. Chapter 5 presents the investigation into how design consequence tracing can improve the design process using two

cases from REXIS. Chapter 6 concludes with a summary, evidence that the methodology presented in Chapter 2 can be generalized to other systems, the contribution to the literature, and future work.

# Chapter 2

## Methodology

This chapter will present the methodology used to quantify the effect of applying of model-based systems engineering to the design process. A system whose design has been completed using traditional systems engineering is modeled in the Systems Modeling Language (SysML) at important points in the design process. The SysML models contain all the topological information about the design at that point in time. Each model is inspected to extract information about the system that is inaccessible with traditional systems engineering methods. A hypothetical design timeline is constructed that includes the information extracted from the system models. To quantify the design process improvements, the historical system design timeline is compared against the hypothetical, model-based design timeline. If the alternative timeline arrives at the final design before the historical design timeline does, then it can be concluded that model-based systems engineering makes the design process more efficient.

The topological information captured in the model includes the interfaces between all parts of the system, the uncertainty associated with each interface, and the path along which the consequences of selected design choices or requirements flow. The latter two types of information are captured using custom extensions to SysML. The uncertainty associated with each interface highlights immature areas of the design that need to undergo changes to arrive at a complete design. Tracing design consequences improves design decision by making the full consequences of each decision option clear to the design team. Both of these extensions have the potential to make the design process more efficient.

The following sections describe how the system model is constructed and what information it contains, how it is inspected, and how an alternative timeline is constructed and compared with the traditional design timeline. Finally, the chapter concludes with a discussion of the limitations of this methodology.

## 2.1 System Model Description and Contents

This section introduces the Systems Modeling Language (SysML) and describes the information that is contained in the system models.

### 2.1.1 The Systems Modeling Language (SysML)

The System Modeling Language (SysML) is a general purpose visual modeling language capable of modeling requirements, design, and behavior that is managed by the Object Management Group (OMG) [28]. SysML is closely related to the Unified Modeling Language (UML), reusing many of its characteristics and adding some new features [29]. There are nine types of SysML diagrams that are divided into three different categories as shown in Figure 2-1. Four diagrams are directly inherited from UML, three have been modified from UML, and two diagrams are unique to SysML. Four diagrams are behavioral diagrams meaning that they capture the behavior of the system, four diagrams are structural diagrams meaning that they capture the parts of the system and how those parts are connected, and one diagram, the requirements diagram, is neither a behavioral diagram nor a structural diagram. Requirements diagrams are used to capture requirements and describe how those requirements relate to other aspects of the system including the part of the system that must satisfy a requirement and the verification activity for a requirement. The diagrams in the orange box (Block Definition Diagram (BDD) and Internal Block Diagram(IBD)) are used to capture interface information in this methodology.

There are four types of behavioral diagrams: activity diagrams, sequence diagrams, state machine diagrams, and use case diagrams. Activity diagrams are used to model the behavior of a system in terms of a flow of actions. Sequence diagrams are used to model the interactions between different system elements. State machine diagrams are used to model

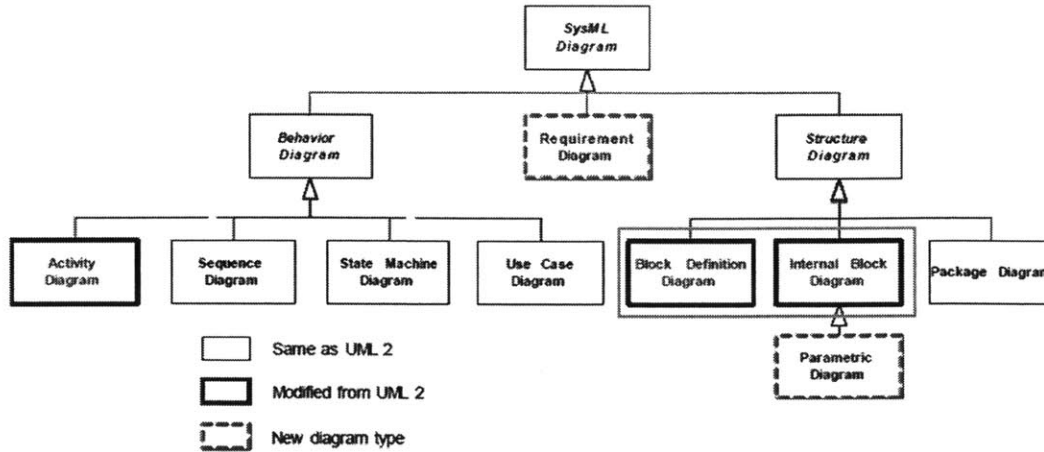


Figure 2-1: The types of SysML diagrams [28]. The diagrams in the orange box are used to capture interface information in this methodology.

state-dependent behavior, including transitions between states. Use case diagrams model the relationships between the system and external actors.

There are four types of structure diagrams: block definition diagrams (BDDs), internal block diagrams (IBDs), package diagrams, and parametric diagrams. Blocks are the fundamental unit used to represent elements of a system. Block definition diagrams describe all the parts of the system and how parts come together in assemblies. Internal block diagrams describe the interfaces between parts of a system.

A port is a SysML element used to describe how a block interacts with other blocks. Ports are connected with Connectors in Internal Block Diagrams to describe an interface. Package diagrams capture the organization of the model. They are not used to describe the system that is being modeled. Parametric diagrams add more detail to internal block diagrams as they capture the quantitative relationships between parts of a system.

### 2.1.2 SysML Model Description

The system models use the built-in SysML capabilities (BDDs, IBDs, ports, connectors, etc.) to capture topological information, but extend SysML in two ways in order to capture additional information about the uncertainty of each interface and information about the effect of each design decision on the system. The basic interface modeling capabilities of SysML are sufficient for capturing interface information, but provide little metadata about

the interfaces that can assist with decision making. The uncertainty of an interface informs where future work must be directed. In other words, the uncertainty is a placeholder for a future change. Design areas with high interface uncertainty are expected to be areas where lots of design detail will be added in the future. Therefore, early identification of areas of high design uncertainty has the same benefits as front-loading change implementation: the rework associated with accommodating the change is minimized [25]. Tracing the path along which the consequences of design choices or requirements flow can improve design decisions by showing where the design has been excessively driven by a decision or by showing where alternative design choices exist. If the choices and requirements that heavily drive the design are identified early, the necessity of those constraints can be analyzed and some constraints may be removed. The following sections will explain how topological information is captured in SysML using a combination of the built-in interface modeling capabilities and custom extensions. Examples will be provided for a generic system.

### **Basic SysML Interface Syntax**

The basic SysML syntax for capturing interfaces forms the foundation of the system models. A BDD and IBD for an example system are shown in Figure 2-2. The BDD shows that the system is composed of one part and one assembly. The assembly can be further broken down into two parts. The IBD shows the interfaces of each part with the other two parts. The boxes on the edge of each part are called Ports and the lines connecting them are called Connectors. The ports are labeled with the string <proxy>. Labeling model elements like this is called stereotyping. This stereotype means that those ports are proxy ports. Proxy ports are SysML symbols that are used to show access to the internal features of the block to which they are attached. They do not represent parts of the system. Connectors represent an interface between two parts or ports. Note that the connectors pass through the assembly boundary without any interaction because the part is interacting with the parts within the assembly, not the entire assembly.



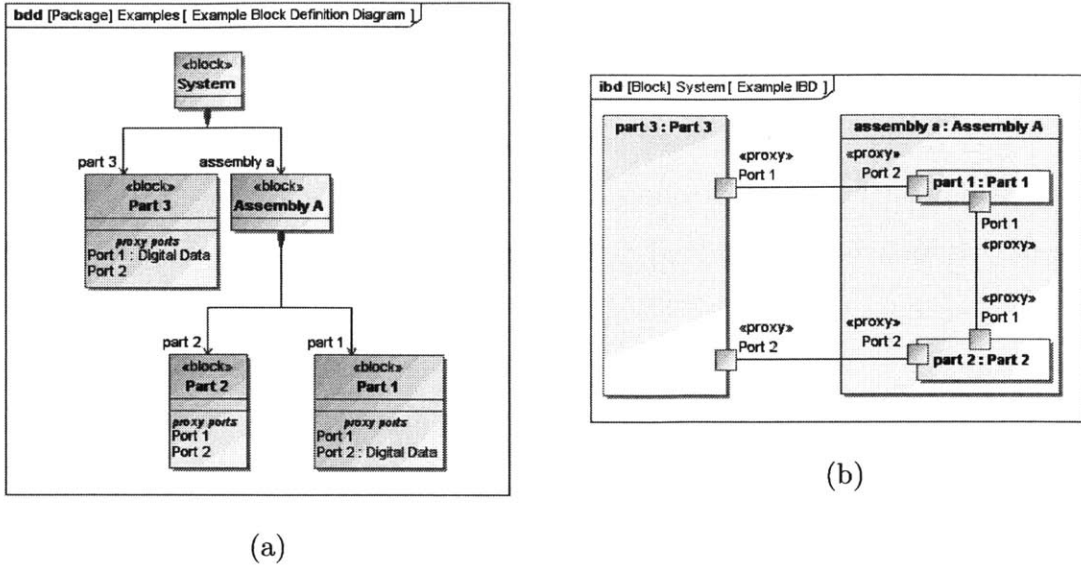


Figure 2-2: An example block definition diagram (a) and internal block diagram (b) showing the basic SysML topological modeling features.

### Capturing Interface Types and Uncertainty

Custom elements were added to the SysML models to enable the type and uncertainty of each interface to be captured. These additional elements facilitate the quantification of the potential for change in certain design areas. The addition of constraints on the range of values of an element in SysML is called typing that element. A fully defined interface is represented by two ports connected by a connector with the ports typed by interface blocks and the connector typed by an association block of the same kind as the interface blocks. Interface blocks are special blocks that are used to type ports, and association blocks are special blocks that are used to type connectors. Custom interface and association blocks were created to enable the typing of each proxy port and connector with custom interface types that capture the uncertainty of each interface.

The interface types were created as needed and grouped into an abstraction hierarchy according to the groups established in Pimmler and Eppinger 1994: Spatial, Information, Material, and Energy [30]. Examples of the abstraction hierarchy for interface and association blocks from the REXIS case study are shown Figure 2-3 and Figure 2-4, respectively. Figure 2-3 shows the abstraction hierarchy for interface blocks with the most generic interface type at the top and the most specific interface types at the bottom. Figure 2-4

shows the abstraction hierarchy for association blocks with the most abstract interface type at the center and the most specific interface types towards the outside. Each association block is tied to the most generic type of interface block to define that those association blocks can type any connector that connects two ports of any interface type on Figure 2-3. The lines with white arrows on the end define the different levels of abstraction and point in the direction of more abstract interface types. Interface types that are not one of the most specific types have uncertainty associated with them. For example, early in the design process, an interface between two parts might be known to exist but no details about the interface have yet been established. Therefore, the interface would be typed with Interface, the most generic type of interface. As the design begins to be established, the knowledge about the interface may increase and it can be established that the interface is an exchange of information. Therefore, the interface is now typed with Information. When the design has matured even more, the interface may be defined as a digital data interface and typed with Digital Data. An example of a Digital Data interface is shown in the context of the example system in Figure 2-5. As shown by the blue ovals, both ports and the connector are typed with Digital Data.

### **Capturing Design Consequences**

The second extension to SysML captures how requirements and design decisions shape the topology of a system. Frequently, the imposition of a requirement on a system causes that system to gain or lose elements or properties of elements to change in order to meet that requirement. For example, a requirement specifying that a car will have cruise control means that the car must contain the sensors and actuators necessary to sense and control its speed. Without the requirement for cruise control, those sensors and actuators do not need to be part of the car. Alternatively, making a design choice, such as choosing a component to fulfill a certain role, may cause the system topology to change as the component is accommodated in the system. Those changes flow through a system along interfaces as components are added or removed. For example, the choice of a radioisotope thermoelectric generator (RTG) to generate power on a spacecraft instead of solar panels necessitates thermal system design changes to accommodate the large amount of heat that will be produced by the RTG. In

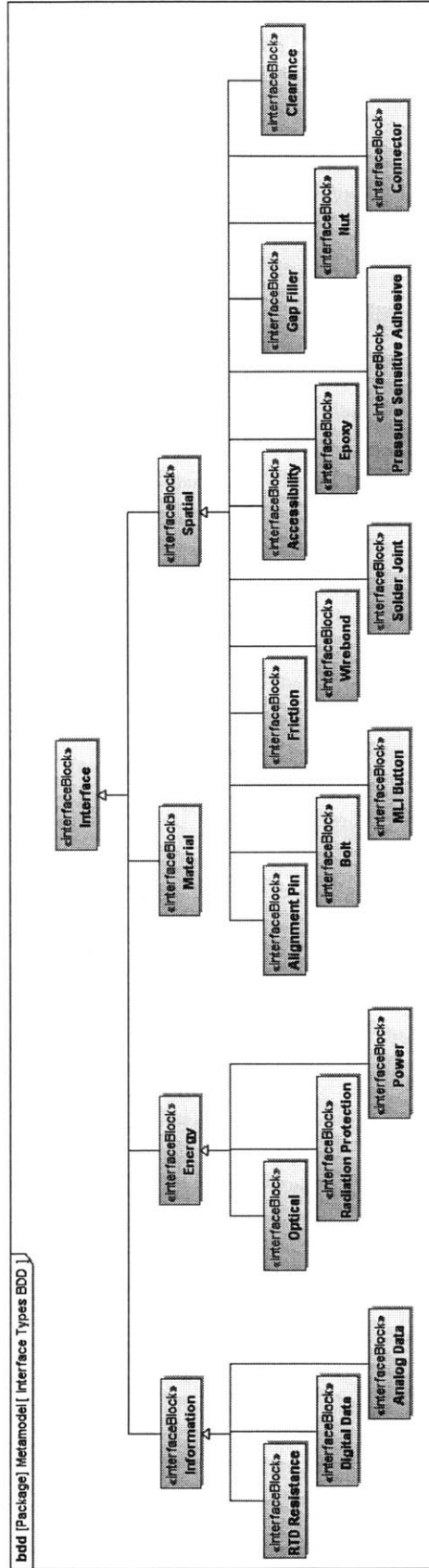


Figure 2-3: The abstraction hierarchy of custom interface blocks created to type each proxy port. The lines with white arrows on the end point in the direction of more abstract interface types.

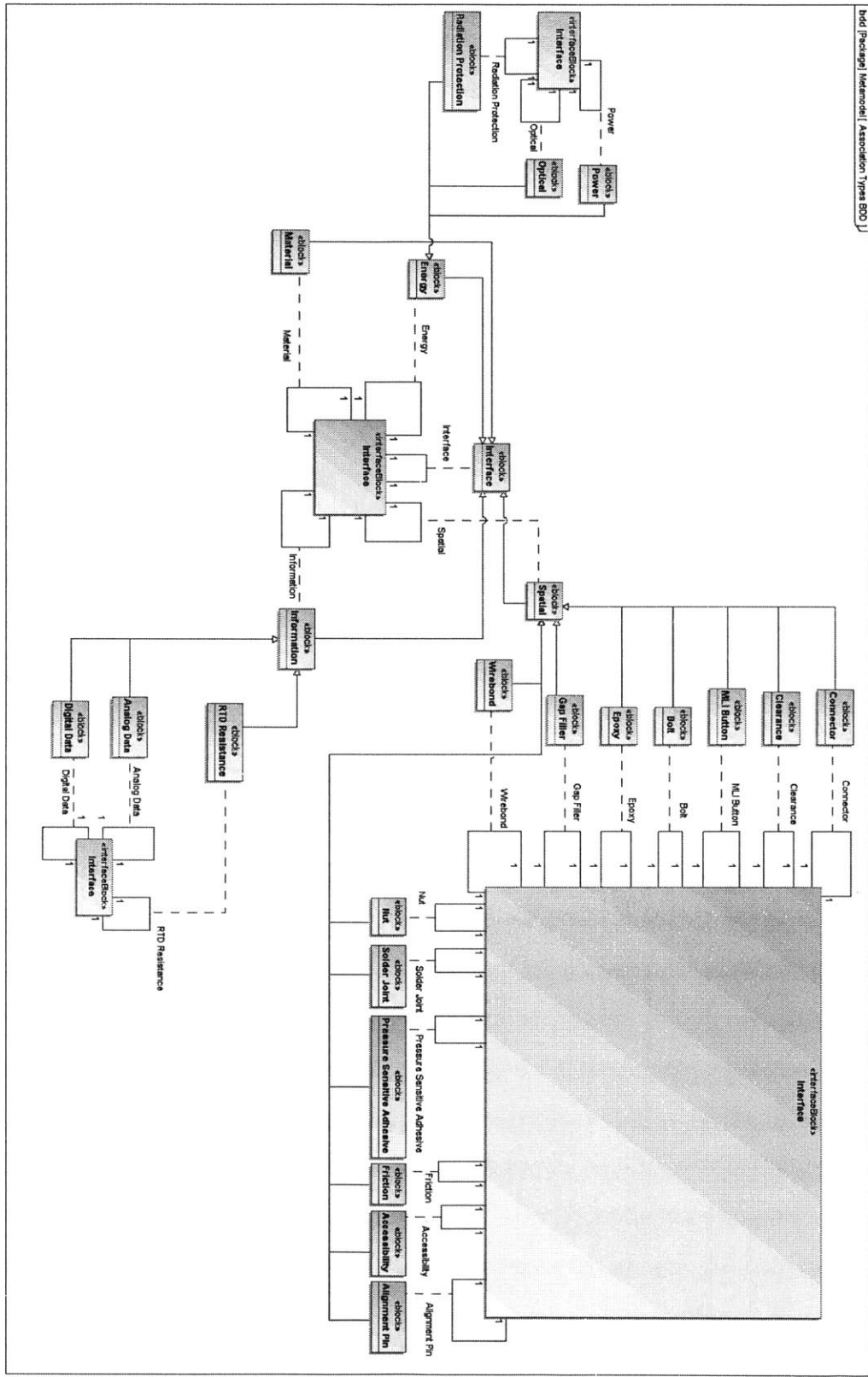


Figure 2-4: The abstraction hierarchy of custom association blocks created to type each connector. The lines with white arrows on the end point in the direction of more abstract interface types.

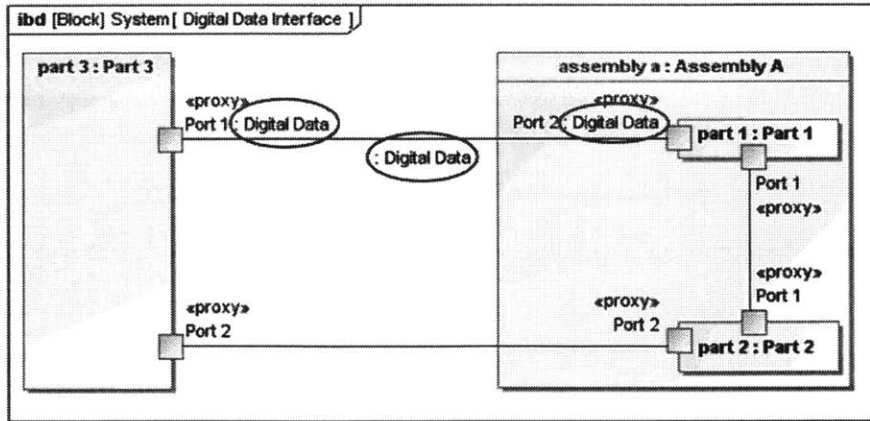


Figure 2-5: An example of a Digital Data interface. The addition of interface types is shown with the blue ovals.

other cases, tracing the design consequences reveals that other possible design solutions exist. For example, assume that the consequence of incorporating an RTG into the design was that thermal isolation was introduced to isolate the RTG from the spacecraft structure and a dedicated radiator was introduced to reject the heat generated by the RTG. By tracing the design consequences from the RTG to the thermal isolation, the possibility of instead thermally coupling the RTG to the structure and using the structure as a radiator may be discovered. Design decisions can be improved by tracing the full ramifications of each design feature and requirement. Choices that drive the system too heavily can be avoided.

In order to trace the ripples of a decision or requirement through a system, a custom stereotype for a Dependency was created. Dependencies are built-in SysML features that symbolize a directional relationship between two model elements. The «designConsequence» stereotype defines the Dependency to which it is applied as tracing the consequences of a design choice or requirement. An example of this extension is shown in Figure 2-6. The added Dependencies are marked with the blue ovals. Because of the choice of Part 3 to fulfill some role, a Digital Data interface needed to be created to allow exchange of information with Part 1.

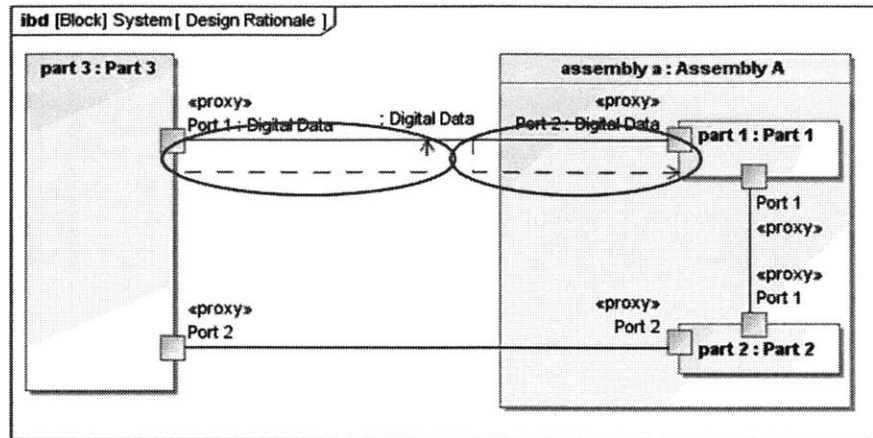


Figure 2-6: An example of Dependencies tracing the consequences of a design choice of Part 3. The added Dependencies are marked with the blue ovals.

## 2.2 Inspecting the System Models

The inspection of the system models consists of quantifying the level of uncertainty for each subassembly and tracing the design ramifications of driving system design choices and requirements. The uncertainty level for each subassembly is quantified by taking the fraction of ports that are of type Interface, Spatial, Information, Material, or Energy (At the first or second level of abstraction) out of the total number of ports associated with parts within that subassembly. Any port with one of these types has not been fully defined and therefore will be changed in the future. Subassemblies with a high fraction of uncertain ports are expected to experience a large increase in the number of parts or ports while subassemblies with a low fraction of uncertain ports are expected to have a nearly frozen design. Because the completeness of each subassembly was not known during the design process, any information gleaned using this method could be used to focus work on subassemblies with higher uncertainty and therefore improve the design process.

The impact of certain requirements or design choices is qualitatively determined by inspecting how their consequence paths flow through the system. If the consequence path for a design choice or requirement goes through many components of the system, then it has a large impact on the design of the system. Additionally, if consequence paths intersect and alter the consequence paths of other requirements or design choices, then that consequence path can be judged to have a large impact on the system. The information about which

design choices or requirements drive the system the most can be used to push back on those requirements or design choices to avoid over constraining the design.

## **2.3 Constructing and Comparing the Design Timelines**

The hypothetical, model-based timeline is constructed by altering the historical timeline using the information extracted from the system models about how the design can be improved. The impact of the information extracted from the system models on the design timeline is quantified by comparing the point in the design process at which the model was created against the point in the design process where that information was known historically.

The model-based and historical design timelines are compared using difference in time between the point at which the final design was developed in the model based timeline and when final design was developed in the historical timeline. If the model-based design timeline arrives at the final design before the historical timeline, then it can be concluded that applying model-based systems engineering improves the design process.

## **2.4 Limitations**

The biggest limitation of this methodology is the issue of separating hindsight from insight. When looking back at previously designed system, it is easy to see the features that would later be changed as the design improved. An effort must be made to separate the information extracted from the model from the information discovered as part of the design process. However, it is impossible to definitely say that a certain piece of information could not have been known by someone who was not involved in the system design. This challenge is analogous to that of differentiating between an immature design and an abstract design. In both cases, the design is not complete but in an immature design, the missing information is not known whereas in an abstract design, the missing information is hidden. Even if the system models are intended to be strict representation of what was known at the time, the missing pieces of information are known to the modeler. The risk is that the conclusions drawn by studying models that are potentially abstract instead of immature are invalid when

applied to a system whose final design is not known. An effort has been made in this thesis to avoid using knowledge of the future design to find issues in past designs. One area of future work would be for an individual not associated with the development of the modeled system to apply this methodology in order to avoid this limitation.



# Chapter 3

## REXIS Design History Overview

This chapter gives an overview of the design history of the REXIS instrument, from initial conception at System Requirements Review (SRR) through Critical Design Review (CDR). The design history is illustrated using snapshots taken at the major milestone reviews (SRR, System Definition Review (SDR), Preliminary Design Review (PDR), and CDR). This chapter first details each design snapshots with traditional tools (CAD, block diagrams, etc.) and then gives an overview of the SRR, SDR, and PDR designs as represented in SysML. The CDR design was not modeled in SysML as it serves as the design history timeline is only built up to CDR, not beyond, so insights gained at CDR do not affect the timeline. Finally, the chapter concludes with a discussion of the limitations of using the REXIS design history as a case study.

### 3.1 REXIS Instrument Overview

The REgolith Imaging X-ray Spectrometer (REXIS) is a risk class D, student-build instrument flying on NASA's OSIRIS-REx mission. OSIRIS-REx will launch in September 2016 on a mission to return a sample from the asteroid 101955 Bennu. The mission is led by the University of Arizona and managed by NASA Goddard Space Flight Center. The spacecraft is built by Lockheed Space Systems Company and includes payloads from the University of Arizona, Arizona State University, NASA Goddard Space Flight Center, and the Canadian Space Agency [31].

REXIS is a coded aperture imaging X-ray spectrometer designed to observe Bennu in the 0.5-7.5 keV soft X-ray band. REXIS will classify Bennu among the known meteorite groups by measuring the elemental ratios Mg/Si, Fe/Si, and S/Si within 20% accuracy. Additionally, REXIS will map the distribution of Iron in the regolith on 50 meter scales. REXIS detects photons from the asteroid using a 2x2 array of charge coupled devices (CCDs). Because the asteroid spectrum is generated through X-ray fluorescence caused by impinging solar X-rays, REXIS must measure the solar X-ray spectrum during observations to properly calibrate the observed asteroid spectrum. Therefore, REXIS contains a separate, Sun-facing X-ray spectrometer called the Solar X-ray Monitor (SXM). REXIS produces images using the coded aperture imaging technique. Photons from the asteroid first pass through a coded aperture mask before being detected by the CCDs. The coded aperture mask is a random pattern of closed and open holes that casts a shadow on the detector array. By deconvolving the shifted and super-imposed shadows on the detector array, a sky image can be reconstructed. The sky images are then co-added as OSIRIS-REx orbits the asteroid to build up a global map. The REXIS science products will complement and enhance the science produced by other instruments on OSIRIS-REx. REXIS is the first planetary wide-field imaging X-ray spectrometer, building on the heritage of past wide-field X-ray spectrometers like the X-ray/Gamma-ray Spectrometer (XGRS) on the Near Earth Asteroid Rendezvous (NEAR) mission and astrophysical coded aperture imaging systems like the Burst Alert Telescope (BAT) on the Swift Midex mission [32, 33].

## **3.2 REXIS Instrument Architecture**

REXIS is composed of two subassemblies: the spectrometer containing the CCDs, coded aperture mask, and electronics boards, and the Solar X-ray Monitor (SXM). Refer to Figure 3-7 and Figure 3-8 for a visual aid to the instrument architecture. Within the spectrometer, the four CCID-41 detectors are housed within the Detector Assembly Mount (DAM). The DAM is attached to the Tower assembly, composed of the Detector Assembly Support Structure (DASS) and the four Tower panels. On top of the truss sits the coded aperture mask. A radiation cover, in combination with the protection provided by the Tower and

DAM, shields the detectors from excessive radiation damage during cruise to Bennu. It opens once OSIRIS-REx is in orbit around the asteroid to permit the collection of data. The Electronics Box houses three electronics boards: the Main Electronics Board (MEB), and the Detector Electronics boards: the Video Board and the Interface Board. The Main Electronics Board contains the main processing unit, the power management, and the communication and power interfaces with the spacecraft. The Video Board digitizes the raw CCD data, while the Interface Board sends the digitized CCD data to the MEB and drives the CCDs. Thermal isolation between the Electronics Box and the detectors, in combination with a thermal strap that removes heat from the detector array using a radiator mounted to the truss, supports a detector temperature of below -60 °C. The SXM is composed of a Silicon Drift Diode (SDD) mounted to a small preamp board. The SDD package contains a thermoelectric cooler to cool the detector below the required operating temperature of -60 °C. The SXM is mounted on a bracket that attaches it to the spacecraft. Because the SXM is on a different face of the spacecraft, it is connected to the Electronics Box with a harness.

### **3.2.1 REXIS Requirements**

The design of REXIS is subject to many different sets of requirements, as shown in Figure 3-1. The Level 1 requirements for REXIS come from NASA and Goddard Space Flight Center documents. Those requirements flow down into the Level 2 requirements consisting of the OSIRIS-REx specific mission assurance, performance, and environmental requirements. The REXIS measurement requirements do not flow directly from the OSIRIS-REx Level 1 science requirements but instead are defined by the REXIS team in a manner that enhances the scientific return of the OSIRIS-REx mission. At Level 3, the project-specific documents are flowed to REXIS in the form of the REXIS Mission Assurance Implementation Plan (MAIP), the functional and performance requirements, and the interface requirements to the spacecraft. Finally, the Level 3 requirements are flowed to the subsystems at Level 4. Just like the design, the requirements evolved over the lifecycle. Table 3.1 shows the number of Level 3 and Level 4 at each milestone review (within the orange box on Figure 3-1). The general pattern is that the number of Level 3 requirements stayed constant while

Table 3.1: Summary of the number of Level 3 and Level 4 requirements at each milestone review.

Milestone Review	Number of Level 3 Requirements	Number of Level 4 Requirements	Total Number of Requirements
SRR	44	114	158
SDR	36	68	104
PDR	38	185	223
CDR	37	248	285

the number of Level 4 requirements increased. The decrease between SRR and SDR is due to a reorganization of the requirements that removed the interface and environmental requirements from the scope of the requirements described in the table.

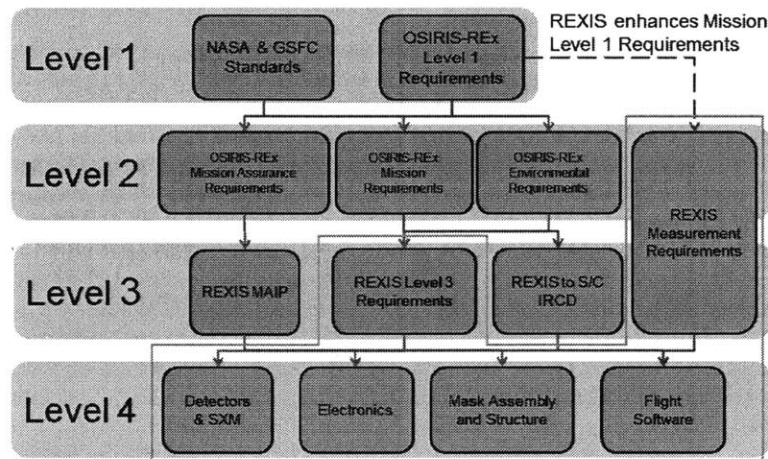


Figure 3-1: The REXIS requirements hierarchy. The number of Level 3 and Level 4 requirements that fall within the orange box are tabulated in Table 3.1.

A small handful of requirements drive most of the REXIS design. The spectral resolution of the CCDs must be fine enough to allow the elemental fluorescence lines to be separated in the asteroid spectrum. The two primary contributors to the CCD spectral resolution are the CCD temperature and the damage that radiation has done to the CCDs. The operating temperature requirement drives the mechanical configuration of the instrument. The warm Electronics Box at the bottom of the instrument must be isolated from the cold detectors inside the Tower. To develop the necessary thermal gradient, thermal isolation exists between the Electronics Box and the DAM and the heat from the DAM is rejected by the Radiator via a thermal strap. These mechanical features are all driven by the thermal requirements. The structure is designed to survive, with appropriate factors of safety, the predicted vibra-

tion environment. The radiation protection requirement drives the need for the Radiation Cover. Because the Radiation Cover must stay closed as long into the mission as possible to provide the maximum amount of protection, the mechanism must actuate successfully after surviving for years in the interplanetary environment. This challenging requirement drives the actuation mechanism, hinge, and cover design. To track the degradation of the CCDs due to radiation damage both while the cover is closed and after it is open, several  $^{55}\text{Fe}$  calibration sources illuminate small parts of the detector.

### 3.3 REXIS Design History

This section describes the historical REXIS design timeline between SRR and CDR. SRR took place on January 23, 2012. SDR took place on April 24, 2012. PDR took place on January 31, 2013. CDR took place from February 18-20, 2014. Figure 3-2 shows a timeline of the key milestone reviews. Starting in the fall of 2011 through the spring of 2012, including the preparations for SRR and SDR, the REXIS team was primarily composed of undergraduate students with graduate and faculty mentors. REXIS development took place as part of the aerospace engineering capstone class. Starting in the summer of 2012, the REXIS team became a mostly graduate student team with faculty mentors and undergraduate volunteers and REXIS development was divorced from classes.

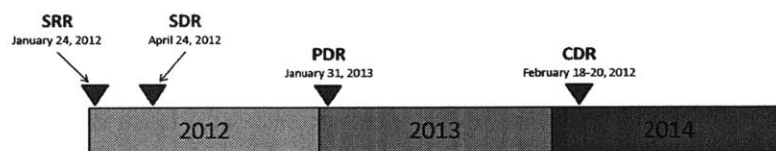


Figure 3-2: The dates of the key milestone reviews for REXIS.

#### 3.3.1 REXIS Design at System Requirements Review (SRR)

The initial design, presented at SDR on January 23, 2012, is shown in Figure 3-3. REXIS was a truss mounted to a simple electronics box. The truss separated the square coded aperture mask from the detectors that were mounted in a detector assembly. The notional Optics Bench and a layer of thermal isolation separated the detector array from the Electronics

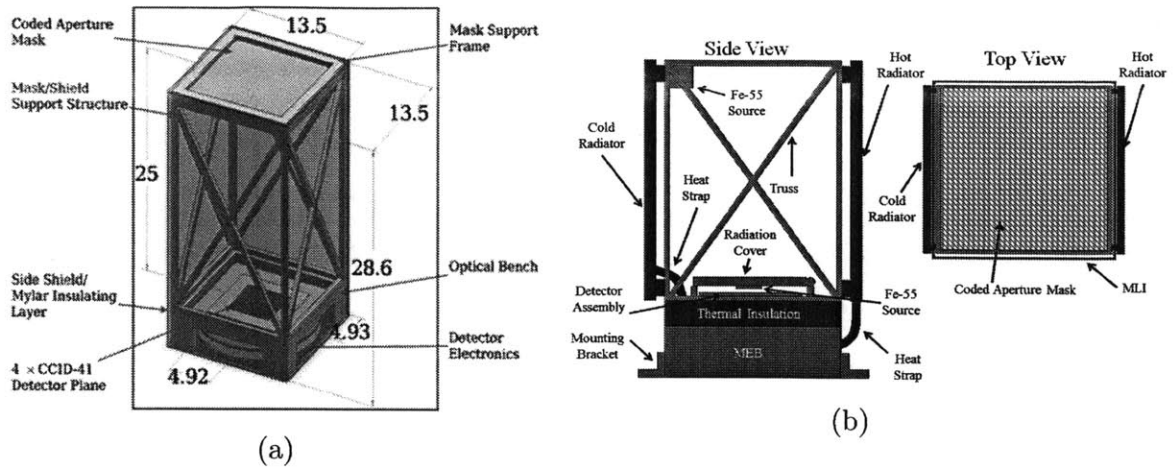


Figure 3-3: The REXIS design at SRR as represented in CAD (a) and in a schematic view that shows more of the design details (b).

Box holding the boards. The thermal design included two thermal straps attached to two radiators. One thermal strap runs from the Optical Bench to a radiator to cool the detectors and the other runs from the Electronics Box to a radiator to reject heat from the electronics boards. A thin sheet of tungsten was attached to the interior of the aluminum truss to block off-axis X-rays from being detected by the CCDs. The detector assembly along with a radiation cover mounted just above the CCDs protected the detectors from radiation damage during cruise. Two  $^{55}\text{Fe}$  calibration sources, one mounted underneath the radiator cover and one mounted high in the truss supported calibration of the detectors both in cruise with the radiation cover closed and while operating with the radiation cover open. Finally, a wrap of MLI around the truss insulated the colder inner surfaces from the external thermal environment. The SXM was proposed as an addition to the design but had not officially been incorporated at this stage.

### 3.3.2 REXIS Design at System Definition Review (SDR)

The REXIS design, shown in Figure 3-4 and Figure 3-5, evolved slightly for SDR three months later. The basic structure of a Tower mounted on the Electronics Box separated by thermal isolation remained the same. The Electronics Box design matured and the spacecraft interface was defined for the first time. The CCD array remained the same but the Detector Assembly Mount (DAM) was introduced to attach the CCDs to the Detector Assembly

Support Structure (DASS), the new name for the Optical Bench. The Radiation Cover was designed and integrated into structure. The cover and hinge sat atop a box that went around the DAM to shield the detectors from radiation in all directions. Due to the confined space within the Tower, the Radiation Cover actuation mechanism was chosen to be an electroresistive break wire. The DAM held two  $^{55}\text{Fe}$  calibration sources and the calibration source on the cover was removed. The thermal design changed slightly with the removal of the cold radiator. Instead, heat from the detectors was rejected via the Tower and side shield, which were left open to the environment on two sides. A Sun Shield was added on the sun-facing side to limit the solar heat input to the cold truss. The SXM was moved to a location on the Sun Shield. On the Main Electronics Board, a Virtex 5 FPGA was chosen to be the central processing unit with volatile and non-volatile memory and the power tree was developed. Finally, the coded aperture mask pattern was changed to a circular pattern to more closely match the expected view of Bennu.

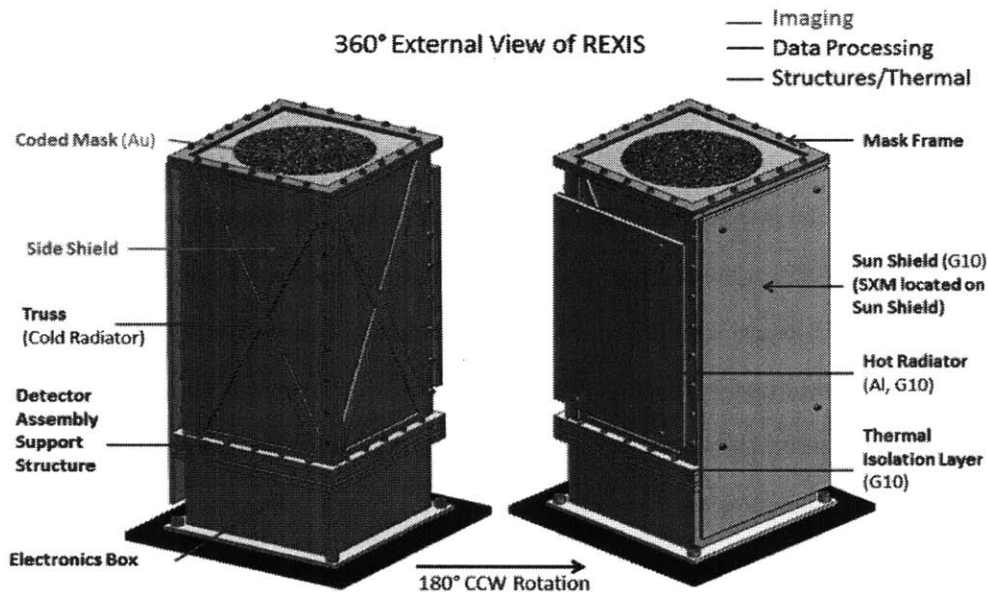


Figure 3-4: The exterior of the REXIS design at SDR.

### 3.3.3 REXIS Design at Preliminary Design Review (PDR)

The REXIS design at PDR, shown in Figure 3-6, matured significantly in the nine months between SDR and PDR. Additional necessary parts were identified and the level of detail for

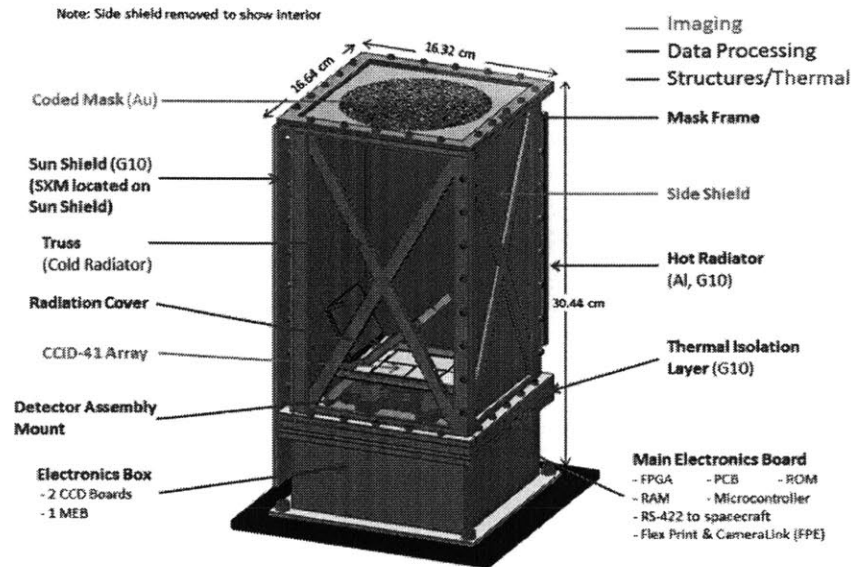


Figure 3-5: A cutaway view of the REXIS design at SDR showing the radiation cover and detectors inside the instrument.

all parts was improved. The biggest change was the removal of the Hot Radiator with the Electronics Box heat now dumped to the spacecraft deck. This change was made possible by the removal of the requirement to be thermally isolated from the spacecraft. The Thermal Isolation Layer was changed from a plate of G10 to five Torlon 5030 standoffs to increase thermal performance. The Tower was no longer used as the Cold Radiator with a dedicated Radiator added back to the design. A graphite thermal strap moved heat from the DASS to the Radiator. The Sun Shield was removed and MLI instead used to isolate the cold Tower structure from the environment. The Side Shields were changed from a separate sheet of tungsten to an aluminum skin integrated into each Tower panel for manufacturing purposes. The DAM design was changed to accommodate four  $^{55}\text{Fe}$  sources collimated to hit specific parts of each CCD and a calibration source was added to the Radiation Cover to facilitate calibration across the entire detector array when the cover was closed. The Radiation Cover actuation mechanism was changed to a TiNi FD04 Frangibolt. Because the cover must be reset manually whenever it is opened, a hole with a non-structural cover was added to a Tower panel to provide access. To connect the CCDs to the Detector Electronics inside the Electronics Box, holes were added to the base of two truss panels to allow flexprints to run out of the Truss to the Electronics Box. The design of the SXM significantly matured to



include an Amptek Silicon Drift Diode (SDD) mounted on a Preamp board, all protected by a housing. The SXM was removed from the old Sun Shield and moved back to a mounting location on a sun-facing spacecraft face, with a harness connecting the SXM to the Main Electronics Board.

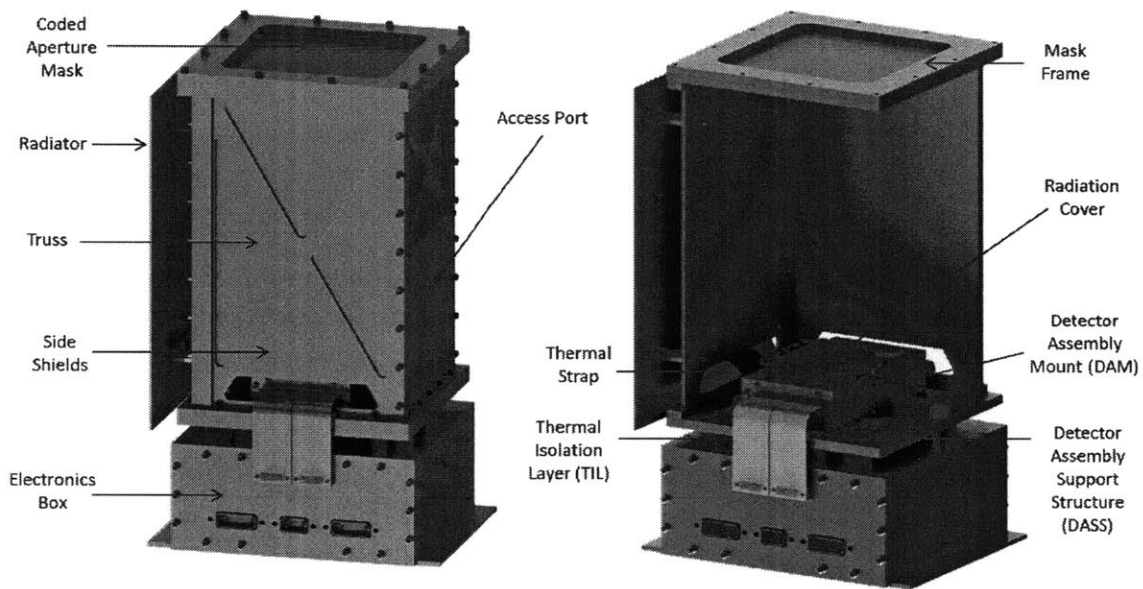


Figure 3-6: The REXIS design at PDR

### 3.3.4 REXIS Design at Critical Design Review (CDR)

Again, the REXIS design underwent significant changes in the thirteen months between PDR and CDR, as shown in Figures 3-7 and 3-8. The Radiation Cover was moved to the top of the instrument to provide easy access for reset as well as to provide room for a heater to keep the Frangibolt and Hinge warm during the cruise to the asteroid. Another layer of thermal isolation was added between the DAM and the DASS to increase the detector temperature margin as well as to provide some isolation from the now-warmer Tower. The new isolation layer consisted of four Torlon 5030 standoffs. The old Thermal Isolation Layer was changed to four titanium standoffs to increase the rigidity of that interface. The thermal strap material was changed to copper and the size of the radiator was increased. Shields were added over the both ends of the flexprints to protect them from handling damage and

excessive heat input. The Electronics Box design was changed to use wedgelocks to attach the electronics boards to the Electronics Box walls. The wiring harnesses that run from the Electronics Box to the Frangibolt, Radiation Cover Heater, and the PRTs on the DAM and Radiation Cover were added to the design. The SXM design changed slightly from PDR with the addition of a collimator to define the SDD field of view.

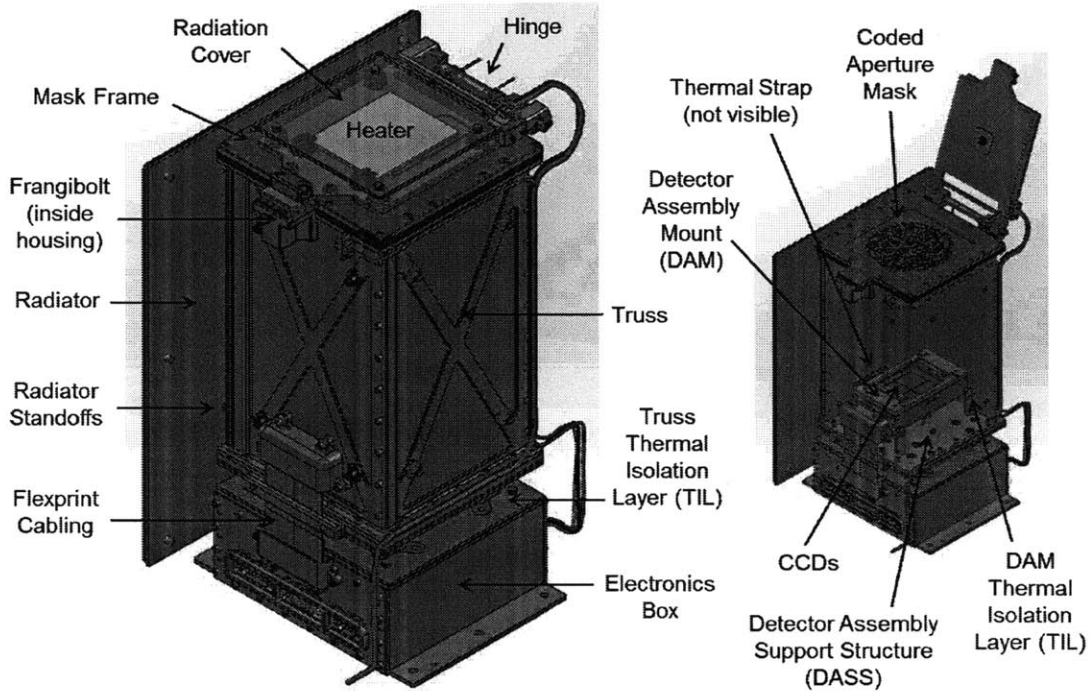


Figure 3-7: The REXIS Spectrometer design at CDR

### 3.4 SysML Models of REXIS at Milestone Reviews

This section describes at a high level the SysML models that were constructed at the SRR, SDR, and PDR milestones. The information described here was captured using only normal SysML syntax. The use of the custom SysML elements will be described in the following chapters.

#### 3.4.1 SysML Representation of SRR Design

The simplicity and uncertainty of the REXIS design at SRR is reflected in the SysML model. The internal block diagram contains only 28 lowest level parts and 109 ports. The structural

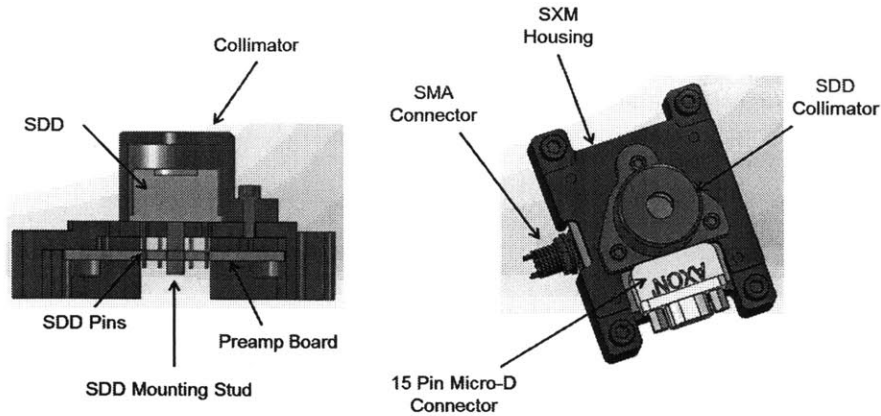


Figure 3-8: The REXIS SXM design at CDR. The bracket that attached the SXM to the spacecraft is not shown.

interfaces run from the spacecraft through the Electronics Box, Thermal Isolation Layer, Tower, to the Mask. The data interfaces run from the CCDs to the Detector Electronics, to the Main Electronics Board, and to the spacecraft. The CCDs receive photons from Bennu through the Mask, and from two  $^{55}\text{Fe}$  sources, one mounted in the Tower and one mounted on the Radiation Cover. The Hot and Cold Radiators dump heat to space from the Electronics Box and Optical Bench respectively. Many of the ports are not fully defined, with only 61 of the 109 being of the lowest abstraction level.

### 3.4.2 SysML Representation of SDR Design

The maturation of the design for SDR is reflected in the SysML model. The internal block diagram contains 57 lowest level parts and 210 ports, more than the SRR design. Detail has been added to the Radiation Cover, Detector Assembly, and Main Electronics Board. The design of the Electronics Box and SXM has changed little from SRR. The basic structural, thermal, and data interfaces remain the same as at SRR with the exceptions of the removal of the Cold Radiator, the addition of the Sun Shield, and the mounting of the SXM on the Sun Shield. The interfaces remain quite general with only 144 of the 210 being of the lowest abstraction level.

### 3.4.3 SysML Representation of PDR Design

The SysML model of the PDR design reflects the large increase in design detail. The internal block diagram contains 150 lowest level parts and 577 ports. The design of the Radiation Cover in particular has advanced substantially. The design fidelity of the Electronics Box and SXM have also greatly increased. The basic structural, thermal, and data architecture again still hold. The interface uncertainty has decreased substantially with 573 of the 577 interfaces being of the lowest abstraction level. The only ports that are not fully defined are those representing the interface between the electronics box and the spacecraft-controlled heaters and temperature sensors that control the temperature of the electronics box. That interface was not fully defined until CDR.

### 3.4.4 Design History Statistics

High level design statistics can be extracted from the SysML models of the SRR, SDR, and PDR designs. The statistics show the large increase in fidelity that occurred from SRR to PDR. Figure 3-9 shows the number of parts in each subassembly at each milestone review and Figure 3-10 shows the number of ports in each subassembly at each milestone review. Those statistics are repeated in Table 3.2. For all subassemblies, the number of parts and ports increased over the lifecycle. The average assembly started with 4.0 parts 15.6 ports and gained 17.4 parts and 66.9 ports between SRR and PDR. The subassembly with the biggest increase in both parts and ports was the DAM. The DAM contains the four CCDs that have many important optical, data, electronic, and radiation protection interfaces with the rest of the instrument. In the model, each CCD at PDR has 17 interfaces, the largest number of interfaces on a single part. Also, packaging and mounting each CCD requires many mechanical parts, driving the part count up. These two factors combine to drive both the part and port count for the DAM high above all other subassemblies.

In addition, the number of parts and ports was strongly correlated at each milestone review as shown in Figure 3-11. Assemblies with a higher number of parts tended to have more ports. The ratio of ports to parts at each milestone review was mostly between 3 and 5, as shown in Figure 3-12. This ratio of interfaces to parts is similar to previously published

interface to part ratios. Whitney found that on average, a part in a complex system has about 6 interfaces [34]. Some subassemblies have noticeable trends. The Detector Electronics dropped from 7 ports per part at SRR to less than 2 ports per part at PDR. This trend is due to the fact that the Detector Electronics was only one part at SRR while the interfaces with the CCDs and the MEB were known. As the Detector Electronics gained design fidelity, most of its interfaces stayed the same because they were already known. The fact that the board-mounted components were not modeled contributes to the low port-per-part number at PDR as well because all of the internal interfaces are missing. This fact also explains why the Main Electronics Board and Detector Electronics have the two lowest port-per-part numbers. The Electronics Box on the other hand experienced growth from under 3 ports per part at SRR to almost 5 ports per part at PDR. Like the Detector Electronics, it was modeled as only a few parts at SRR, lost a part for SDR, but then gained 15 parts at PDR. Unlike the Detector Electronics, the interfaces within the Electronics Box were modeled and the Electronics Box turns out to be a complicated mechanical assembly that features many internal interfaces as it features five standoffs and several MLI blankets.

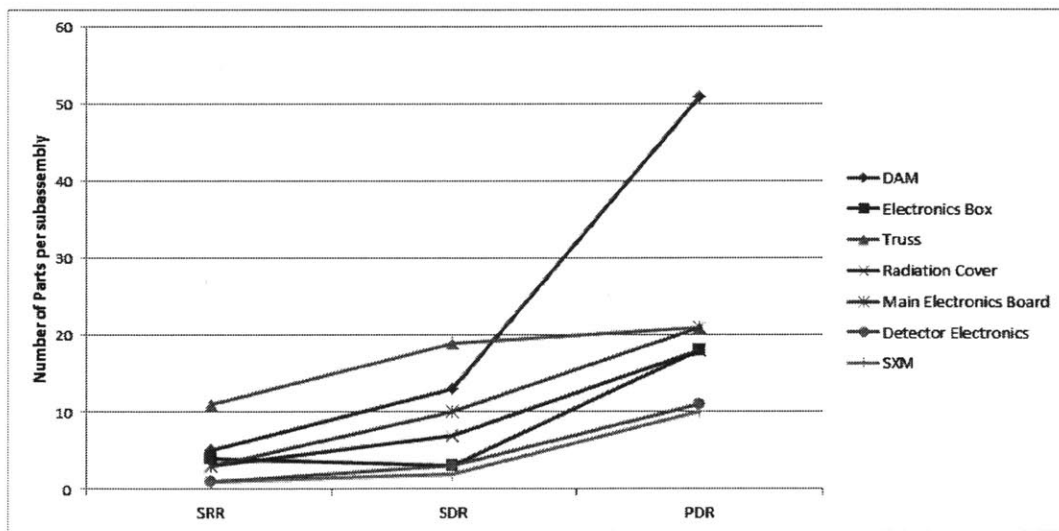


Figure 3-9: The number of parts per subassembly plotted at each milestone review. All subassemblies experienced growth in the number of parts over the lifecycle.

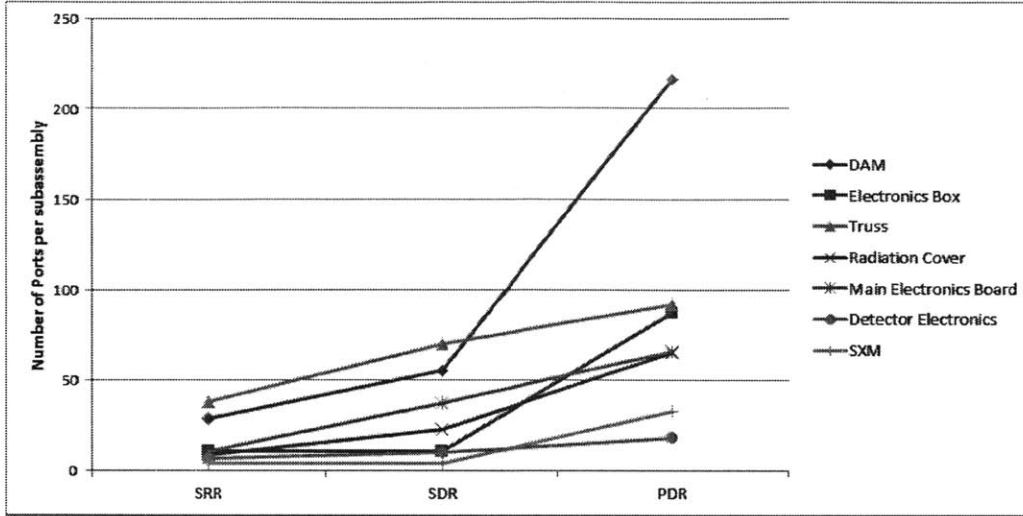


Figure 3-10: The number of ports per subassembly plotted at each milestone review. All subassemblies experienced growth in the number of ports over the lifecycle.

Table 3.2: Summary of the number of parts and ports at each milestone review.

Milestone Review	Number of Parts	Number of Ports
SRR	28	109
SDR	57	210
PDR	150	577

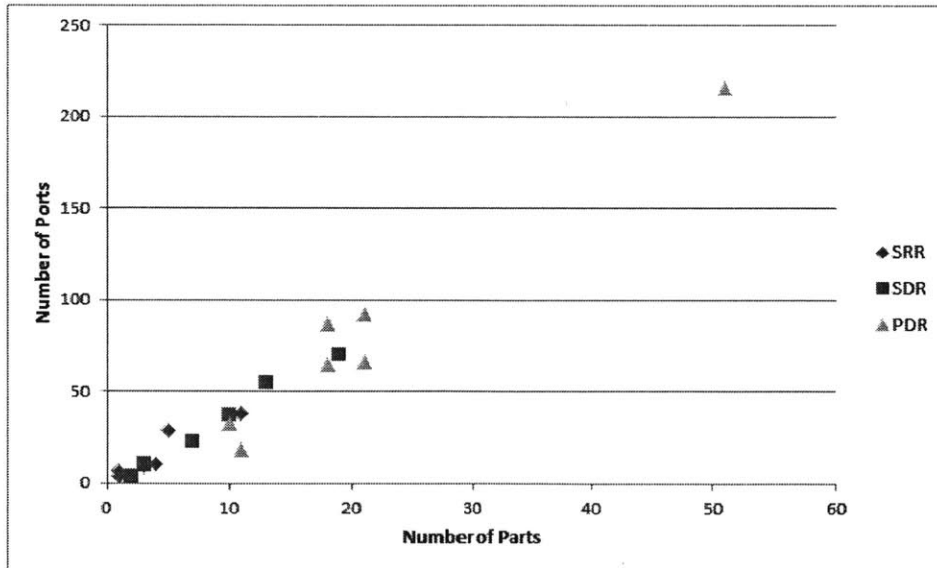


Figure 3-11: The number of ports per subassembly plotted against the number of parts per subassembly. Subassemblies with more parts tended to have more parts at all milestone reviews.

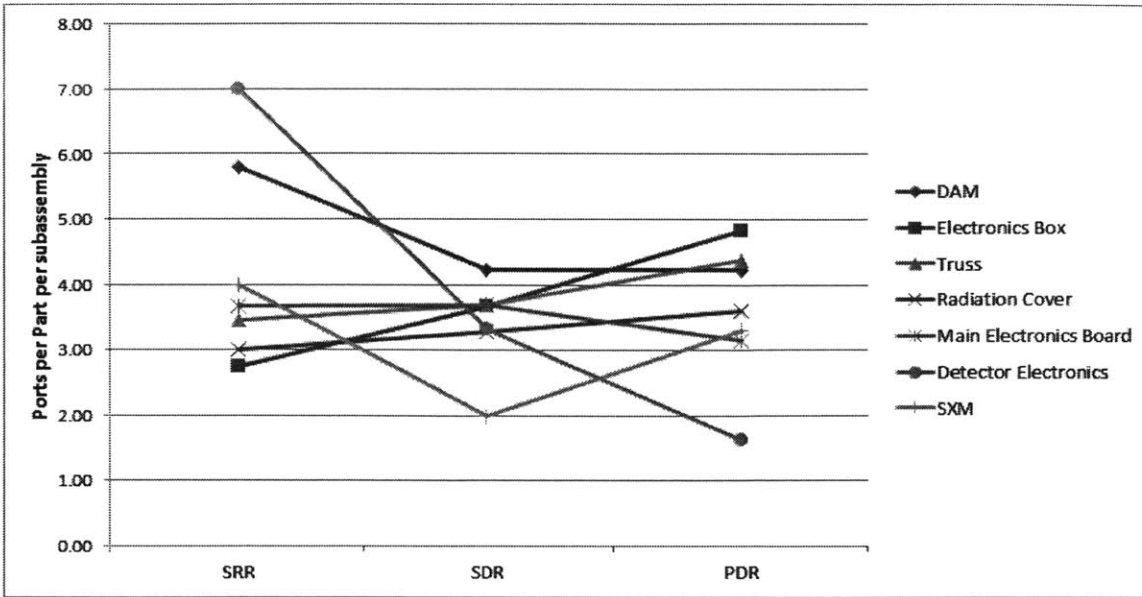


Figure 3-12: The number of ports per part for each subassembly plotted at each milestone review. The average ratio across all milestone reviews was about 4 but some subassemblies tended to gain more parts than ports over the design history whereas others tended to gain more parts than ports.

### 3.5 Limitations of using REXIS as a Case Study

The limitation of applying this methodology to REXIS is that REXIS was built by a student team whereas other systems are built by an experienced team. The REXIS team has, by design, no experience developing space systems. Therefore, certain design choices that were made may reflect this inexperience and would not have been made by a more experienced team. Therefore, the conclusions drawn in this thesis may be less pronounced on a more experienced team or a team developing a system with significant heritage. However, the results should hold for most teams who are developing a new system that shares little similarities with systems that the team has developed before.

Additionally, the system models do not capture all of the topological information that was known at each milestone. Parametric information about each interface, like thermal conductance or voltage, is not captured. Interfaces are not modeled at the lowest possible level. For example, individual bolts and pins within a connector are not modeled. Not all interfaces are modeled. For example, thermal radiative interfaces are not modeled and interfaces between parts that are sensitive to charged particles and the structural elements

that shield those parts are only modeled where the structural component was specifically sized to provide shielding. Not all system elements are modeled. The atomic part for the models is defined as a single mechanical piece, with some exceptions. For example, not all parts on each printed circuit board are modeled. While this missing information can be modeled in SysML, it is beyond the scope of this thesis to go into that level of detail.



# Chapter 4

## Improving the Design Process with Interface Uncertainty

This chapter tests the predictive ability of interface uncertainty statistics at a subassembly level using the REXIS instrument as a case study. Since interfaces with high uncertainty need to be fully defined as part of a complete design, they represent areas of the design that will change in the future. By focusing effort on reducing uncertainty early in the project, the rework associated with late changes can be avoided. Two hypotheses will be tested in this chapter: that subassembly interface uncertainty can predict future changes in that subassembly and that subassembly interface uncertainty can predict the eventual size of that subassembly. Both growth and size are measured in parts and ports within a subassembly. This chapter will show that the interface uncertainty statistics are not a good predictor of areas of future part and port growth or of the eventual size of a subassembly in terms of part or ports. Therefore, had interface uncertainty information been available during the REXIS design process, the design process would not have been improved.

### 4.1 Subassembly Interface Uncertainty Trends

Figure 4-1 shows the percentage of interfaces that are uncertain at each of the milestone reviews. An uncertain interface is one that is not at the lowest level of abstraction in the hierarchy defined in Figure 2-3. Because a complete design should have interfaces that are at

the lowest level of abstraction, any interfaces that are not at the lowest level of abstraction are places where change must take place. Therefore, subassemblies with many uncertain interfaces should expect to see more change than those with more developed interfaces. For example, at SRR, the data interface between the SXM and the C&DH components on the Main Electronics Board was of type Information because the exact nature of that interface was not known. The interface stayed type Information at SDR but at PDR, the interface was fully designed. It consisted of a connector mounted to the Preamp board that connected to a harness that ran from the SXM to the MEB. The harness carried information of type Analog Data from the Preamp to the MEB. In this example, one uncertain interface was fully defined as multiple parts and specific interfaces. To normalize against the differing number of interfaces among the subassemblies, the interface uncertainty for a given subassembly is defined as the percent of ports belonging to parts in that subassembly that are not at the lowest level of abstraction.

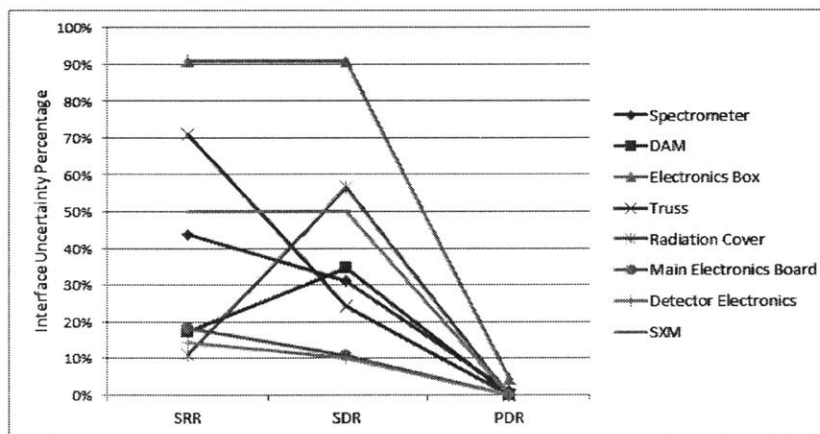


Figure 4-1: The percentage of uncertain interfaces in each subassembly at the key milestone reviews.

As the design is developed and frozen, the percentage of uncertain interface should decline. However, the data shows that is not always the case. Some subassemblies like the Truss display the expected pattern with initially high uncertainty steadily decreasing until the design is complete at PDR. Other subassemblies, like the Electronics Box and SXM show no change from SRR to SDR, indicating that little work was done on the interfaces of those subassemblies in that period. Finally, other interfaces show an increase in interface uncertainty from SRR to SDR before decreasing from SDR to PDR. Both the Radiation

Cover and DAM display this pattern. The reason for this difference is not clear in this figure but will be explored in the following sections.

## 4.2 Interface Uncertainty as a Predictor of Future Sub-assembly Change

The hypothesis that future subassembly growth can be predicted by interface uncertainty is tested by looking for correlation between interface uncertainty and the change in the number of parts or ports between milestone reviews. The intervals between SRR and SDR, SDR and PDR, and SRR and PDR were all analyzed. The conclusion for all three intervals is that interface uncertainty did not predict the amount of future growth correctly.

Figure 4-2 shows the increase in the number of parts and ports between SRR and SDR for all subassemblies plotted against the interface uncertainty of the subassemblies at SRR. Overall, there is no trend. Subassemblies at low and high interface uncertainty experience uniform growth in the number of parts and ports. The subassembly with the largest increase in parts and ports, the Truss, had a relatively high interface uncertainty, as the hypothesis suggested. However, the two subassemblies with the lowest increase in part and ports, the SXM and the Electronics Box, also had high interface uncertainty.

Figure 4-3 shows the increase in the number of parts and ports between SDR and PDR against the interface uncertainty of the subassemblies at SDR. It tells the same story as Figure 4-2. Contrary to the hypothesis, the subassembly with the largest increase in parts and ports was the DAM but it had a relatively low interface uncertainty. Yet, the opposite hypothesis is not supported either as the subassembly with the second largest increase in parts and ports, the Electronics Box, had the highest interface uncertainty.

Figure 4-4 shows the increase in the number of parts and ports between SRR and PDR for all subassemblies plotted against the interface uncertainty of the subassemblies at SRR. Again, the hypothesis is not supported. The subassembly with the highest increase in parts and ports, the DAM, has a low interface uncertainty. The subassembly with the lowest increase in parts, the SXM, has a middling interface uncertainty, while the subassembly

with the lowest increase in ports, the Detector Electronics, has a low interface uncertainty. The evidence in all three plots points towards the same conclusion: that interface uncertainty cannot predict the future growth of a subassembly.

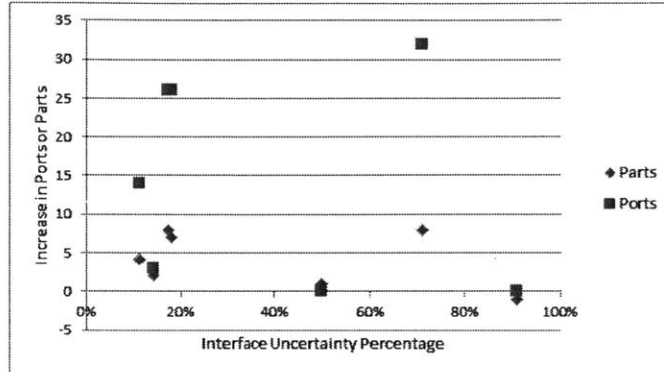


Figure 4-2: The increase in the number of parts and ports for each subassembly between SRR and SDR plotted against the interface uncertainty for each subassembly at SRR.

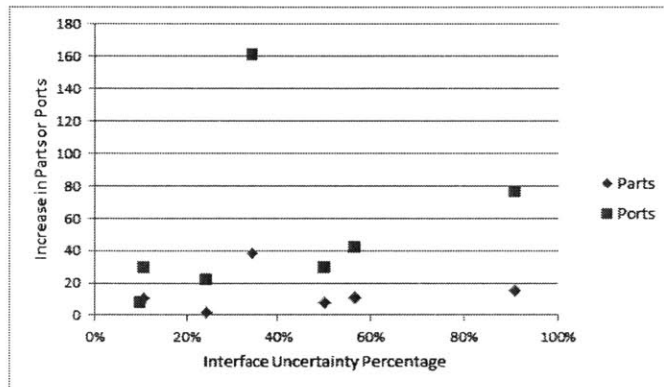


Figure 4-3: The increase in the number of parts and ports for each subassembly between SDR and PDR plotted against the interface uncertainty for each subassembly at SDR.

### 4.3 Interface Uncertainty as a Predictor of Final Subassembly Size

This section tests the hypothesis that interface uncertainty is a good predictor of final subassembly size. Again, the hypothesis is tested by looking for correlation between the interface uncertainty of a subassembly at a milestone review and the size of that subassembly as measured in parts and ports at a future milestone review. The intervals between SRR and SDR,

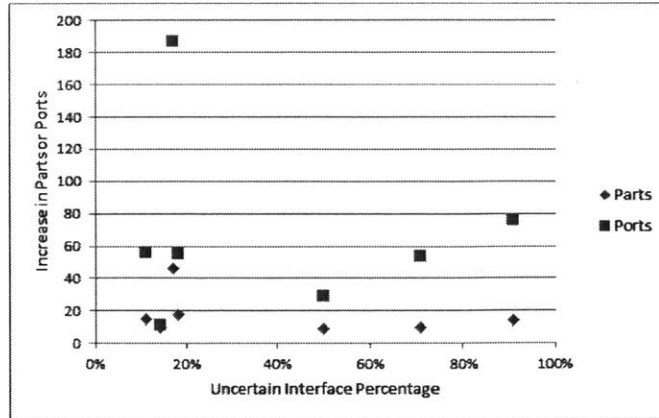


Figure 4-4: The increase in the number of parts and ports for each subassembly between SRR and PDR plotted against the interface uncertainty for each subassembly at SRR.

between SDR and PDR, and between SRR and PDR were analyzed. The conclusion for all three intervals is that interface uncertainty cannot predict the future size of a subassembly.

Figure 4-5 shows the number of parts and ports for each subassembly at SDR plotted against the interface uncertainty for each subassembly at SRR. There is no trend to the data. The subassembly with the highest number of parts and ports, the Truss, has a high interface uncertainty. However, the subassembly with the least number of parts and ports, the SXM, also has fairly high interface uncertainty.

Figure 4-6 shows the number of parts and ports for each subassembly at PDR plotted against the interface uncertainty for each subassembly at SDR. The size of a subassembly in terms of parts or ports is again independent of interface uncertainty. The biggest subassembly in terms of parts and ports, the DAM, has low interface uncertainty, while the subassembly with the lowest number of parts, the SXM has fairly high interface uncertainty.

Figure 4-7 shows the number of parts and ports for each subassembly at PDR plotted against the interface uncertainty for each subassembly at SRR. The relationship between the number of parts or ports in a subassembly and interface uncertainty is quite flat. The subassembly with the largest number of parts and ports, the DAM, has low interface uncertainty, while the subassembly with the second largest number of parts and ports has high interface uncertainty. Based on the evidence in the three ploats, the same conclusion must be drawn that was drawn in the last section: interface uncertainty cannot predict the future size of a subassembly.

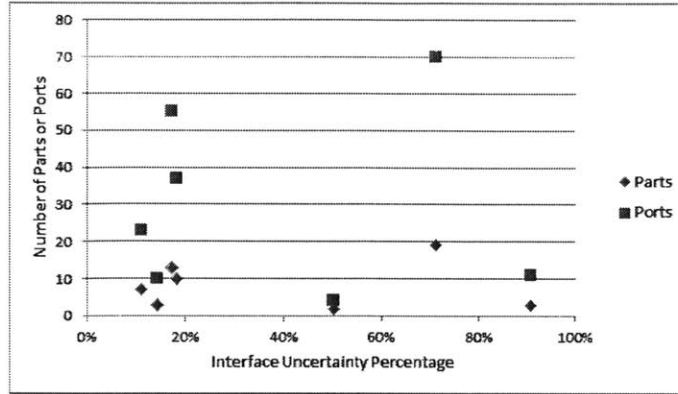


Figure 4-5: The number of parts and ports for each subassembly at SDR plotted against the interface uncertainty for each subassembly at SRR.

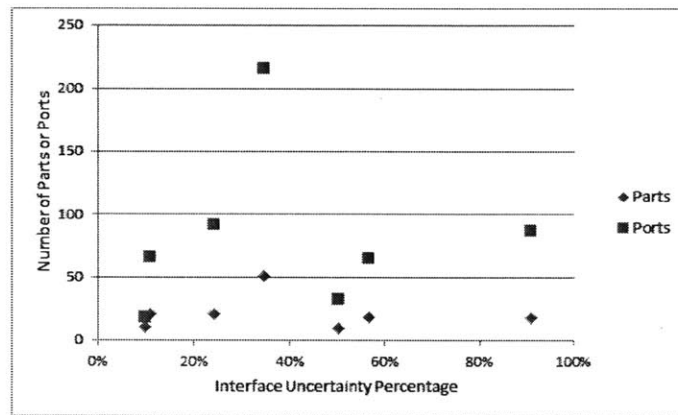


Figure 4-6: The number of parts and ports for each subassembly at PDR plotted against the interface uncertainty for each subassembly at SDR.

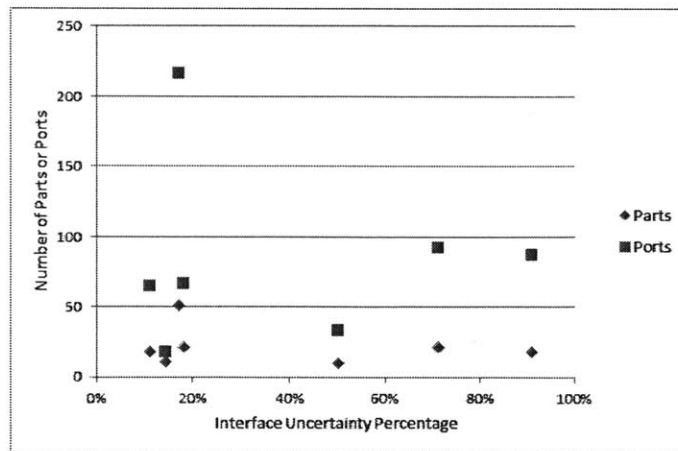


Figure 4-7: The number of parts and ports for each subassembly at PDR plotted against the interface uncertainty for each subassembly at SRR.

## 4.4 Summary

The hypotheses that interface uncertainty alone can predict future growth of a subassembly or the future size of a subassembly are both rejected. The evidence shows that both subassembly size and growth rate are independent of interface uncertainty. The reasons for this are that interface uncertainty does not capture all of the uncertainty that is present in the early stages of a design. The premise behind the hypotheses was that interface uncertainty captured the gaps in the design. An assumption behind this premise is that the parts of the system were already well defined and would not undergo change except for when needed to accommodate an interface change. In other words, systems whose parts as defined at SRR did not undergo much decomposition would fit the premise.

For some subassemblies, this premise appeared to hold. For example, the parts of the Truss were roughly correct at SRR. Most of the part and port addition to the Truss was done to get rid of the uncertainty that existed in the interfaces between the parts of the Truss. Accordingly, the data shows that the increase in parts and ports in the Truss were high when the interface uncertainty was high and the increase in parts and ports was low when the interface uncertainty was low.

Other subassemblies did not follow this pattern because much of the uncertainty in the subassembly was associated with the parts in the subassembly, not the interfaces. For example, the Radiation Cover grew from 3 parts at SRR to 18 parts at PDR and much of the increases were the decomposition of the block named Radiation Cover in the SRR system model. From an interface uncertainty point of view, those changes are unmeasurable because they take place within a part instead of between two parts.

Lastly, other subassemblies did not follow the assumed pattern because the method of quantifying interface uncertainty used in thesis weighted interfaces equally. For example, the massive growth of parts and ports within the DAM subassembly took the place of a single Spatial interface in the SRR model. Because many of the CCDs interfaces were known at SRR, the overall interface uncertainty of the DAM at SRR was low. A more accurate method would account for the fact that the structural connection from the CCDs to the Optical Bench in the SRR model really represented a very complicated structure.





## Chapter 5

# Improving the Design Process by Tracing Design Consequences

This chapter will test the design process improvements that can be made by analyzing design consequence paths using two examples from REXIS as case studies. The two case studies are the evolution of the REXIS thermal system and the evolution of the REXIS Radiation Cover. These elements of the design serve as good case studies because the design of both changed radically from SRR to CDR and because both elements are significant drivers of the overall REXIS design. Tracing design consequences can improve the design process by illuminating the full ramifications of design choices or requirements. The design team can then push back on those choices or make further changes to the system to fully accommodate all of the ramifications. The following sections will present the actual design history of the thermal system and Radiation Cover and then system models of both at important milestones in the design process, concluding with an alternative history that may have occurred had a system model been available. By comparing the two design histories, the advantages of having a model available are clear.

## 5.1 Design Evolution of the REXIS Thermal System

### 5.1.1 Historical Design Evolution

The thermal system has evolved from a two radiator, two thermal strap, one thermal isolation layer design at SRR to a one radiator, one thermal strap, two thermal isolation layer design at CDR. In addition, early designs had trim heaters located near the CCDs to stabilize the detector temperature, while later designs lost these heaters but gained a heater to warm the Radiation Cover mechanism. Because of the challenging requirement to keep the detectors below  $-60\text{ }^{\circ}\text{C}$  while operating, the thermal system design, to a large extent, drives the design of the entire instrument.

#### Thermal System at SRR

At SRR, the thermal system consisted of two radiators, each with a thermal strap, one to cool the detectors and one to reject heat from the Electronics Box. The cold radiator, the one that cooled the detectors, faced towards deep space while the hot radiator, the one that controlled the temperature of the Electronics Box, was placed opposite of the cold radiator on the truss, facing inwards towards the spacecraft. The thermal strap for the cold radiator ran from the optical bench to which the detectors were mounted through the truss to the cold radiator. The thermal strap for the hot radiator ran from the outside of the Electronics Box up to the hot radiator. Both radiators were stood off from the truss to provide some isolation from the warmer truss. A plate of G10 between the warm Electronics Box and cold Truss structure maintained the desired large temperature gradient. MLI wrapped around the truss insulated the instrument structure from the external radiative environment and the instrument was thermally isolated from the spacecraft deck. For a good schematic of the SRR thermal system design, refer to Figure 3-3. While the architecture of the thermal system at SRR was established, there was lots of uncertainty on the specifics of the design. Among the more important parameters, the radiator coating, thermal strap material and length, radiator standoff material and length, and the mechanism through which the thermal isolation was attached to the Electronics Box and Optical Bench were not defined.

## Thermal System at SDR

At SDR, the thermal system changed little from the SRR design, as shown in Figure 5-1. The basic architecture of using two radiators, one to cool the detectors and one to cool the Electronics Box was unchanged. The cold radiator from the SRR design was removed and instead the thermal straps, now two, that cooled the detectors ran from the DASS (called the old Optical Bench at SRR) to the truss. Part of the truss was left open to the environment to serve as the cold radiator. Instead of MLI wrapped around the truss, a G10 sunshade mounted on the sun-facing side of the truss protected the structure from the main external heat input. The final change was that the instrument was now thermally coupled to the spacecraft deck. A number of uncertain parameters from SRR were specified for the first time including the choice of stainless steel radiator standoffs and bolts that ran from the DASS to the Electronics Box to hold the thermal isolation in place. However, the radiator coating and thermal strap material and length remained undefined.

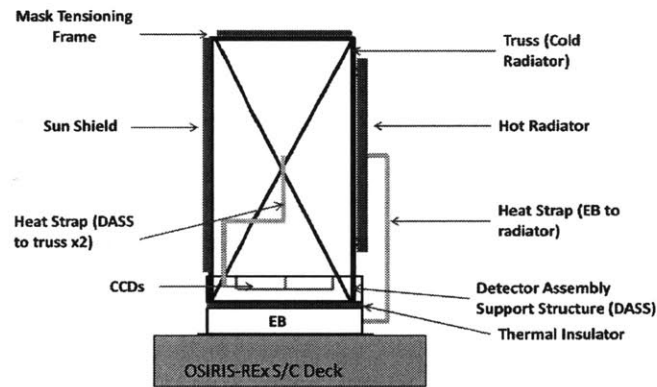


Figure 5-1: The design of the thermal system at SDR

## Thermal System at PDR

The thermal system underwent significant change between SDR and PDR, as shown in Figure 5-2. The hot radiator was removed and the heat from the electronics box was dumped to the spacecraft deck. The cold radiator was reintroduced as a distinct radiator stood off from the truss with a thermal strap running from the DASS through a hole in the truss to the cold radiator. The thermal isolation layer was changed from a plate of G10 to five Torlon 5030 standoffs to reduce the heat transfer from the Electronics Box to the DASS. All remaining

uncertain parameters were specified at PDR, including the choice of a graphite fiber thermal strap and NASA/GSFC NS74 white paint for the radiator.

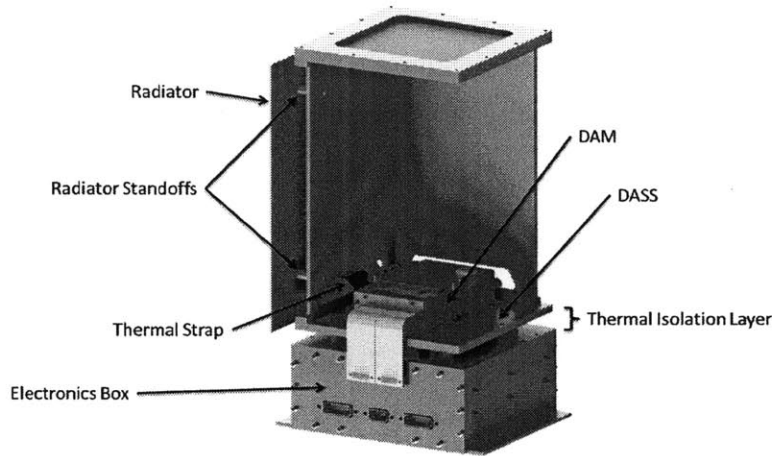


Figure 5-2: The design of the thermal system at PDR

### Thermal System at CDR

The thermal system changed only a little between PDR and CDR, as shown in Figure 5-3. The largest change was the introduction of another thermal isolation layer between the DASS and the DAM. The new isolation layer provided additional margin against the required detector temperature and also provided insulation between the new heater on the Radiation Cover, now at the top of the instrument, and the detectors. Another change was moving from five Torlon standoffs in the first isolation layer to four titanium standoffs to provide more structural margin at the cost of some thermal margin. Furthermore, the radiator was made slightly larger to provide more detector temperature margin, the radiator paint was changed to Z93C55 white paint, and the thermal strap was changed from graphite fiber to copper to reduce cost while maintaining performance.

### Summary

Historically, the design of the thermal system evolved between each milestone review. Figure 5-4 shows the thermal system architecture at each milestone review. Listed above the transition is the major driver of the design change between each review. If incorporating

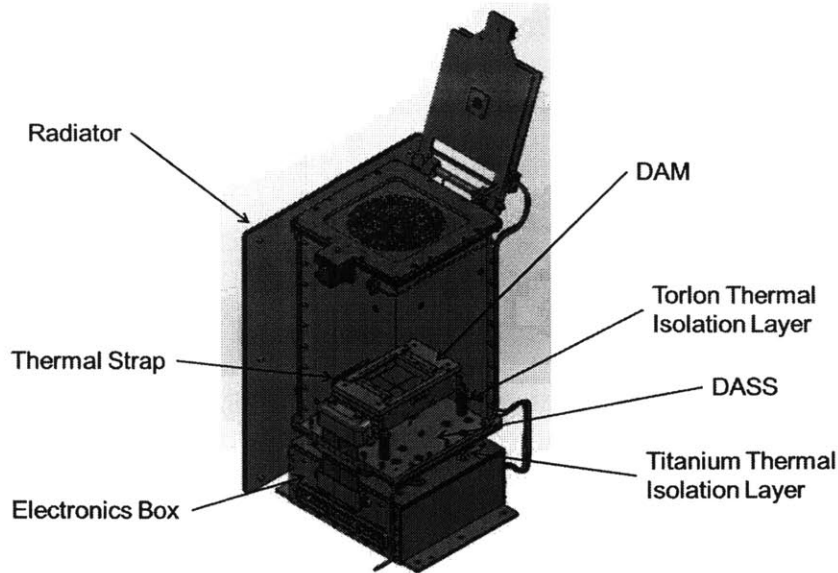


Figure 5-3: The design of the thermal system at CDR

information from design consequence tracing causes the design timeline to arrive at the final design timeline before CDR, then it can be concluded that access to system models would have made the REXIS design process more efficient.

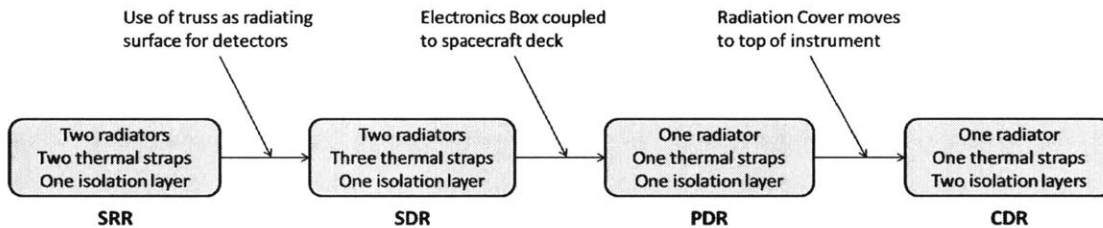


Figure 5-4: A flow chart of the historical design evolution of the REXIS thermal system. The design changed between each milestone review. The major driver of the design change is listed above each transition between reviews.

### 5.1.2 Model-Based Design Evolution of the Thermal System

This section shows how access to a system model improves decision-making ability in such a way that much of the evolution of the thermal system could have proceeded much more rapidly, saving time and effort.

## SysML Model of the Thermal System at SRR

Figure 5-5 shows the historical REXIS thermal system design at SRR in a SysML Internal Block Diagram (IBD). The dashed lines show how the consequences of the requirements propagate through the design. Starting at the operational temperature requirement on the temperature of the focal plane and following the blue path in Figure 5-5, the design must reject heat from the CCID-41s and prevent heat from making its way to the detectors. Therefore, a high conductance path exists from each detector to the Optical Bench, to the Cold Thermal Strap, to the Cold Radiator where the heat is radiated to deep space. To prevent heat from the Electronics Box from heating the detectors, following the red path in Figure 5-5, Thermal Isolation is put between the Electronics Box and the Optical Bench. To prevent heat from the Sun warming the Truss and therefore the Optical Bench, multi-layer insulation (MLI) is placed around the Truss. The requirements to keep the electronic components within their operational and survival ranges mean that some way is needed to regulate the temperature of the Main Electronics Board and the Detector Electronics. Because heat from the boards cannot be rejected through the Cold Radiator due to the Thermal Isolation between the Electronics Box and the Optical Bench, and because heat from the boards cannot be dumped to the spacecraft due to the spacecraft thermal isolation requirement (green path in Figure 5-5), heat from the boards must be rejected by the Hot Radiator via the Hot Thermal Strap (purple path in Figure 5-5). To isolate the radiators from the warmer Truss, the connection between the two, not yet fully defined at SRR, must be of low thermal conductivity.

From the trace of design consequences, the spacecraft thermal isolation requirement is revealed to be a big driver of the thermal system design. The conductive path to the spacecraft is required to be very low conductivity. Therefore, the heat from the Main Electronics Board and the Detector Electronics had to be rejected via a second radiator. The existence of the spacecraft thermal isolation requirement directly drove the addition of a second radiator and second thermal strap. Given the increase in complexity that the second thermal strap and radiator constitute, the REXIS team should have pushed back on the necessity of the spacecraft isolation requirement. Indeed, historically, when the spacecraft realized the

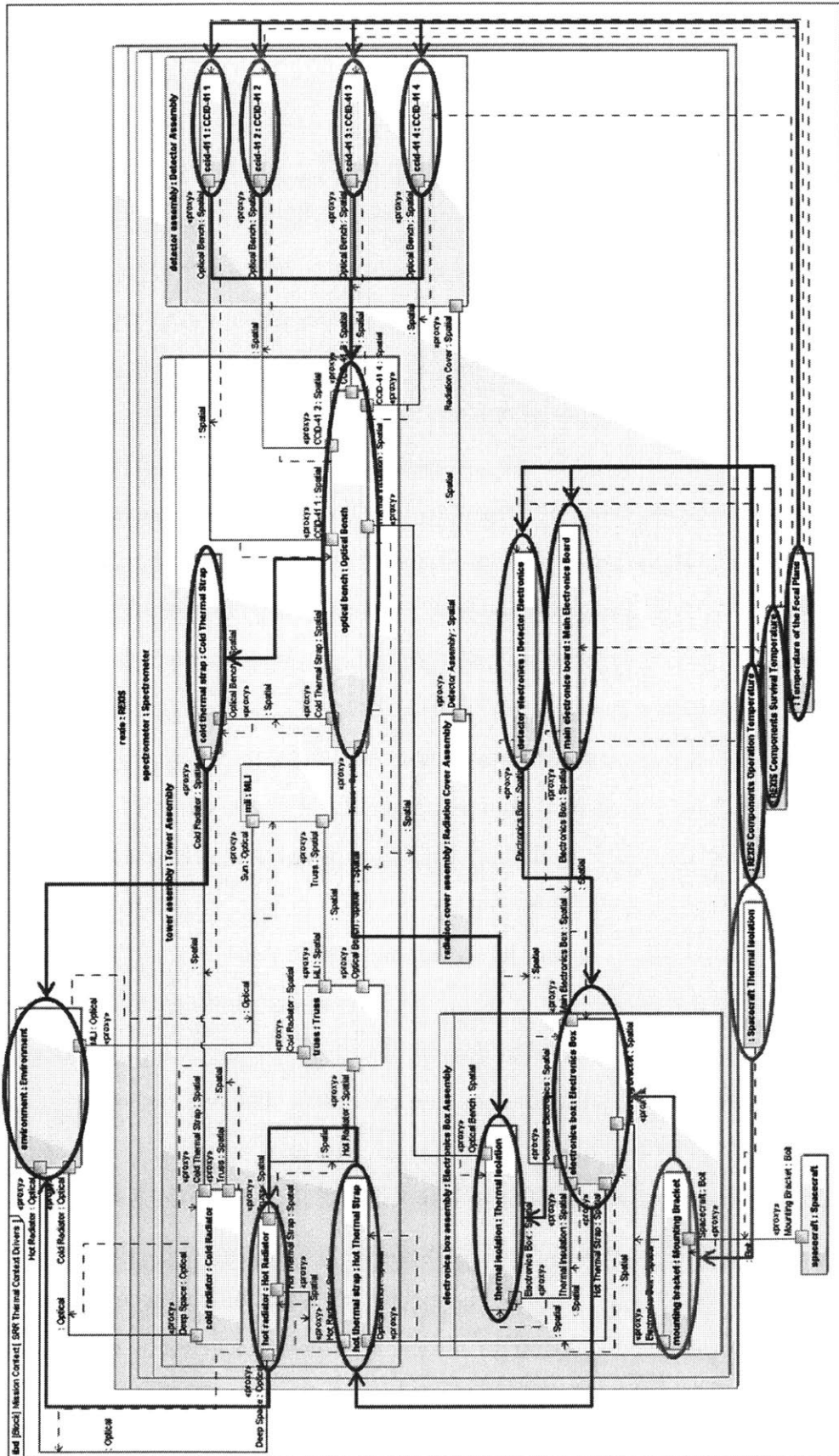


Figure 5-5: Design of the REXIS thermal system at SRR as represented in an Internal Block Diagram

impact of the requirement on the REXIS design, they agreed that the requirement was not necessary and removed it post SRR, but historically the REXIS design did not drop the Hot Radiator and Thermal Strap until PDR.

### **SysML Model of the Thermal System at SDR**

Figure 5-6 shows the historical REXIS thermal system at SDR in a SysML Internal Block Diagram (IBD). Again, the dashed lines show how the effects of the requirements propagate through the design. The model shows the increase in design detail since SRR but the thermal paths are largely the same. The requirement on detector array temperature necessitates a high conductivity path from each CCID-41 through the Package Base and Mount Plate to the DASS, through both Cold Thermal Straps to the Truss where the heat is radiated to deep space. That high conductivity path is protected from the heat of the Electronics Box by the Thermal Isolation layer and from the heat of the Sun by the Sun Shield. The Hot Radiator and Sun Shield are isolated from the colder Truss by low conductivity standoffs. A new requirement on detector temperature stability caused the introduction of two trim heaters placed on the Package Base that have a good conductive path to each CCID-41. The requirements on component operational and survival temperature are still present, again resulting in the heat rejection path from the Electronics Box to the Hot Thermal Strap to the Hot Radiator.

The SDR SysML model shows both the avoidable design oversight of the existence of the Hot Radiator despite the possibility of dumping Electronics Box heat to the spacecraft (green box in Figure 5-6) and a new design option of a second thermal isolation layer that historically was not noticed until CDR (blue box in Figure 5-6). The removal of the spacecraft isolation requirement leaves a clear path for Electronics Box heat rejection to the spacecraft. Yet historically, this disconnect was not noticed and the SDR design unnecessarily featured the Hot Thermal Strap and Hot Radiator. With a system model available, the design would have adapted immediately to the removal of the spacecraft isolation requirement post SRR. Also, the SDR model shows a new location for a thermal isolation layer that was not noticed at the time: between the Mounting Plate and the DASS. Because in SRR the detectors were mounted directly on the Optical Bench, this location did not exist and the Thermal Isolation



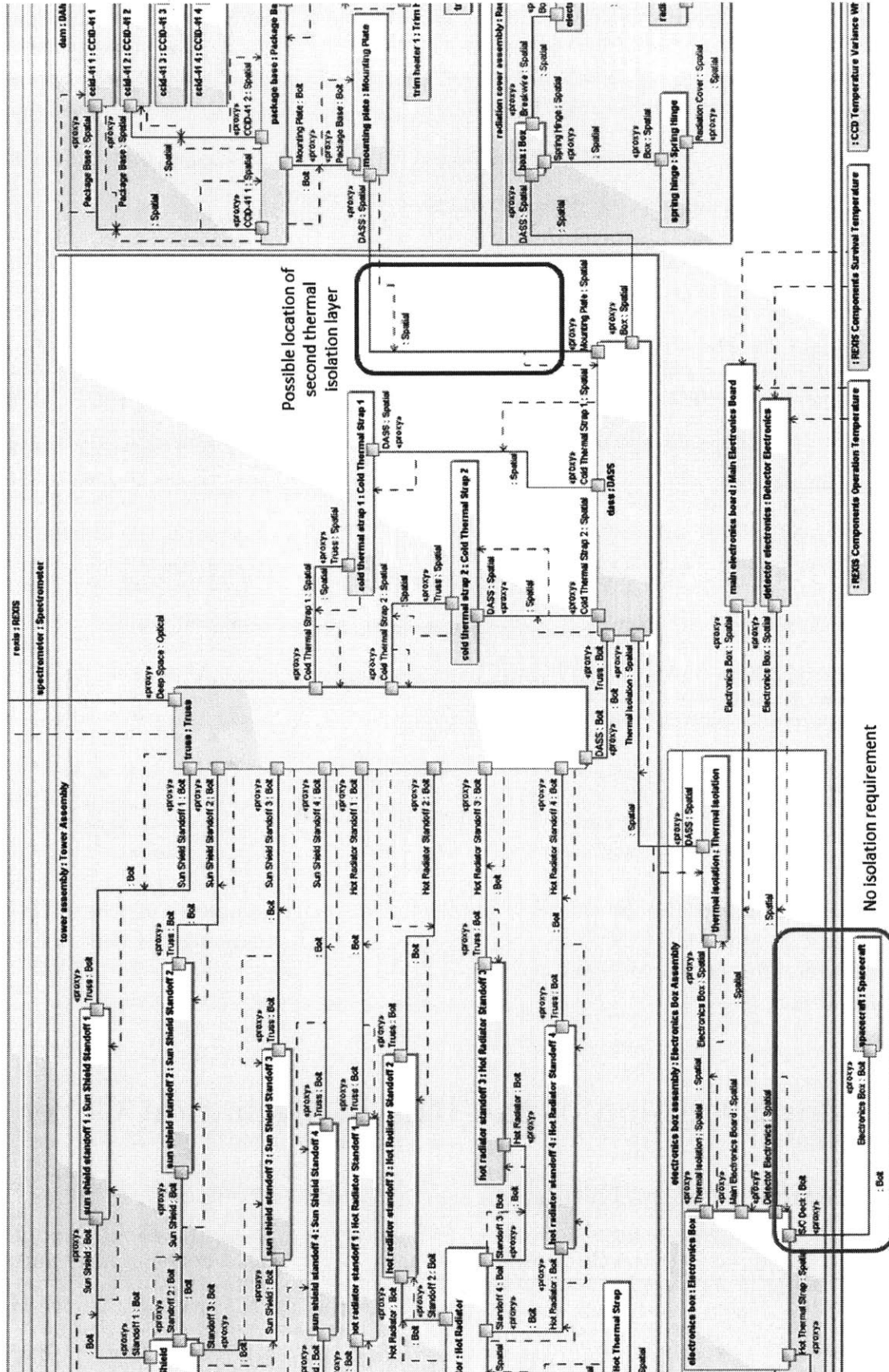


Figure 5-6: Design of the REXIS thermal system at SDR as represented in an Internal Block Diagram

was placed between the Electronics Box and the Optical Bench. Now, with the introduction of a separate plate to which the detector packages mount, the isolation layer could have been moved, or a second isolation layer added. Historically, this possibility was only noticed post PDR and a second thermal isolation layer was added. Again, access to the model presents the possibility of the speeding up the design evolution.

## Summary

The SysML model introduced two opportunities to make the design evolution of the thermal system more efficient: removing the Hot Radiator before SDR and introducing the second thermal isolation layer before PDR. Historically, those two design decisions were made post SDR and post PDR respectively. The hypothetical model-based thermal system design timeline incorporating those improvements is shown in Figure 5-7. Based on the alternative timeline, it can be concluded that incorporating information from design consequence tracing reduces rework and design iteration and makes the design process more efficient.

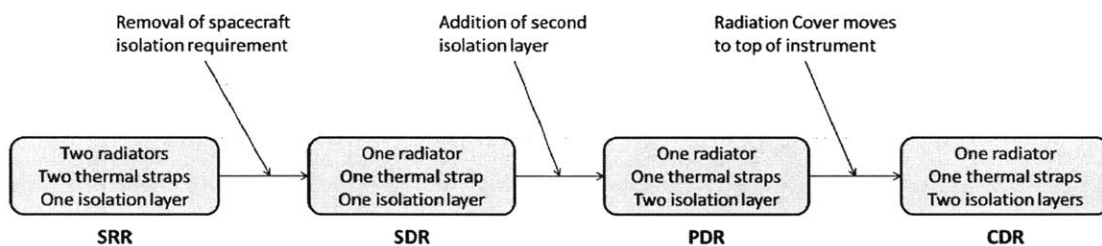


Figure 5-7: A flow chart of an alternative, model based design evolution of the REXIS thermal system. The historical CDR design is identified by PDR, reducing the time wasted analyzing and reviewing unused designs.

## 5.2 Design Evolution of the REXIS Radiation Cover

### 5.2.1 Historical Design Evolution

Over the REXIS design history, the Radiation Cover has evolved from a notional mechanism to a mature, tested design. The early designs placed the Radiation Cover just above the detectors but before CDR, the location was moved to atop the mask frame. Out of all the

REXIS subassemblies, the Radiation Cover has undergone the highest number of design changes as a reliable design was gradually developed.

### Radiation Cover at SRR

The Radiation Cover at SRR existed solely as a notional design. The hinge design, actuator, and size were all undetermined. The only design choices that had been made at this point were to mount the cover just above the detector assembly and to make the cover about 4mm thick.

### Radiation Cover at SDR

A preliminary Radiation Cover was designed for SDR. The Radiation Cover assembly consisted of an aluminum box mounted to the DASS surrounding the detectors, package base, and mounting plate with an aperture through which the detectors could view the asteroid. The Radiation Cover covered the aperture while closed. The cover was opened by a spring hinge and was held open by two spring latches mounted to a backstop. Because of the need to fit the cover inside the truss cavity, an electroresistive break wire mechanism was chosen to release the cover from its closed position. The thickness of the Radiation Cover was increased to around 6mm. Figure 5-8 shows the SDR Radiation Cover design.

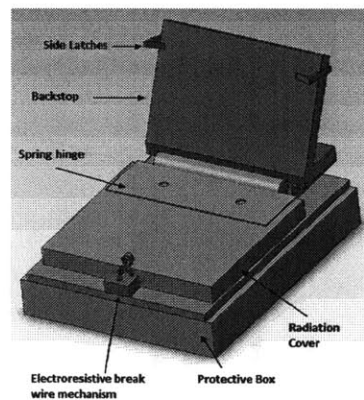


Figure 5-8: The design of the Radiation Cover at SDR

## Radiation Cover at PDR

The design of the Radiation Cover matured significantly between SDR and PDR, as shown in Figure 5-9. The box from the SDR design became part of the DAM. The hinge design changed from a COTS spring hinge to a custom design featuring a stainless steel shaft with Vespel bushings to provide a low friction surface between the Radiation Cover and top shield of the DAM and the shaft. Two torsion springs provided the torque to open the cover. The actuation mechanism was changed from an electroresistive break wire to a TiNi FD04 Frangibolt. Two Belleville washers provided some kick off force to counter and stiction or cold welding that may have occurred during cruise. The backstop and latches from the SDR design were removed. Instead, the cover was held open against the inside of the truss by the torsion springs. A bumper made of casted epoxy provided reduced the shock of the cover hitting the inside of the truss. An  $^{55}\text{Fe}$  calibration source was mounted on the underside of the cover to illuminate the detectors when the cover was closed. An oversight at SDR was a design feature that allowed reset of the cover without disassembly of the entire truss. Therefore, at PDR, as shown in Figure 3-6, a hole was put in the truss opposite of the Radiation Cover hinge to allow the Radiation Cover subassembly to be removed and reset without disturbing the rest of the instrument.

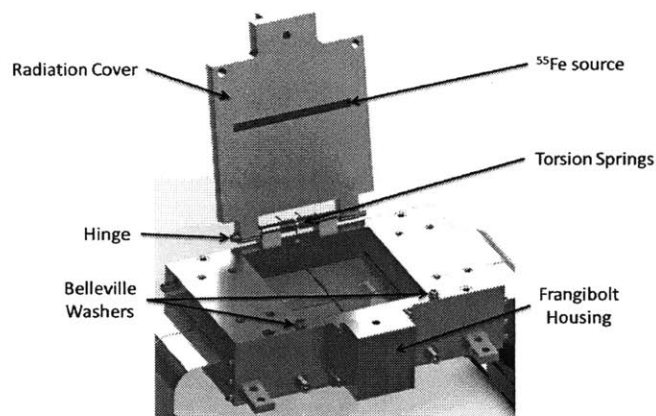


Figure 5-9: The design of the Radiation Cover at PDR

## Radiation Cover at CDR

The design of the Radiation Cover at CDR is shown in Figure 5-10. At CDR, the cover was moved from the DAM to the Mask Frame. This change was driven by the difficulty of accessing the cover to reset it once it had actuated when it was inside the truss, the desire to keep the Frangibolt and hinge warm during the cruise phase while still keeping the detectors cool to allow collection of calibration data, and the new truss panel design that incorporated the old tungsten side shield into each truss panel. The new truss design now blocked radiation whereas previously only the DAM and Radiation Cover blocked high energy particles. The design of the cover was kept generally similar to the PDR design except for those changes that allowed it to be mounted on the mask frame and the addition of a heater and mechanical thermostats to maintain the cover system at relatively warm temperatures during cruise. One small change was the introduction of more Vespel bushings to keep the torsion springs from contacting the shaft. Another was the addition of MLI buttons as now that the cover was mounted on the outside of the instrument, it would be exposed to the external thermal environment.

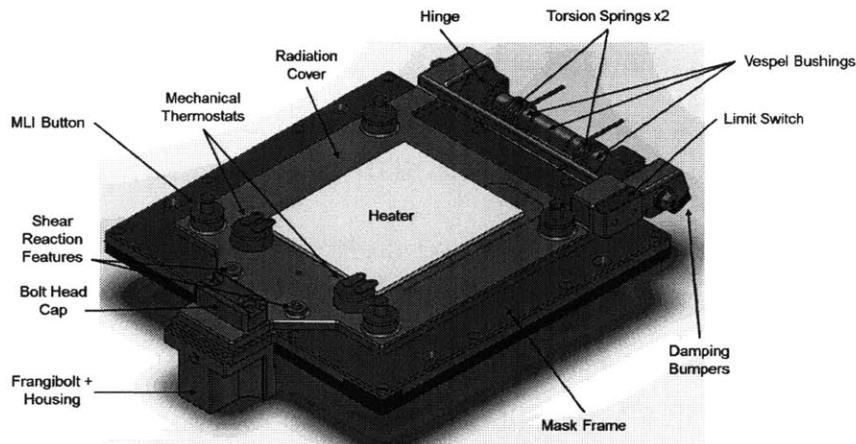


Figure 5-10: The design of the Radiation Cover at CDR

## Summary

Like the thermal system, historically, the design of the radiation changed between each milestone review. Figure 5-11 shows the design evolution of the Radiation Cover with the major design changes between the milestone review listed above each transition between

reviews. If incorporating information from design consequence tracing causes the design timeline to arrive at the final design timeline before CDR, then it can be concluded that access to system models would have made the REXIS design process more efficient.

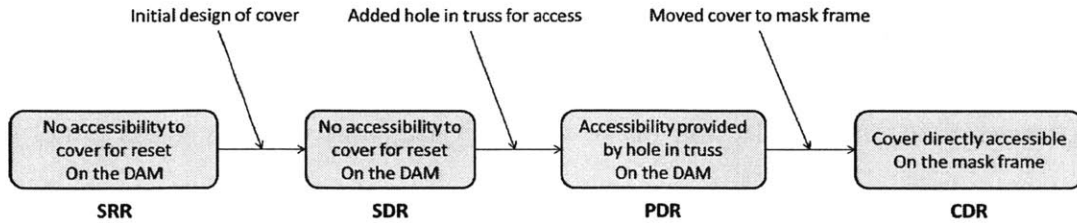


Figure 5-11: A flow chart of the historical design evolution of the REXIS Radiation Cover. The design changed between each milestone review. The major driver of the design change is listed above each transition between reviews.

## 5.2.2 Model-Based Design Evolution of the Radiator Cover

This section shows how having a system model available would have sped up the development of the Radiation Cover. In particular, using the information about how design decisions ripple through the design, the cover could have been moved to the mask frame months ahead of when that change was made historically.

### SysML Model of the Radiation Cover at SDR

Figure 5-12 shows the historical Radiation Cover design at SDR. The dashed lines show how the effects of design decisions propagate through the rest of the design. The SDR represents the first complete Radiation Cover design. Although each physical piece of the REXIS structure protects the detectors from radiation in a small way, the radiation protection for each CCD was designed to be provided by the combination of the DASS, Box, and Radiation Cover. The Radiation Cover is mounted to the Box via a Spring Hinge and is released by an Electroresistive Break Wire. The choice of an Electroresistive Break Wire is driven by the size of space between the Truss and the Box (blue box in Figure 5-12).

The oversight in the SDR design of no access to the Radiation Cover without disassembly of the truss is notable by its absence in the model. There should be an «Accessibility» relationship from the Radiation Cover Assembly to the Environment. This oversight was

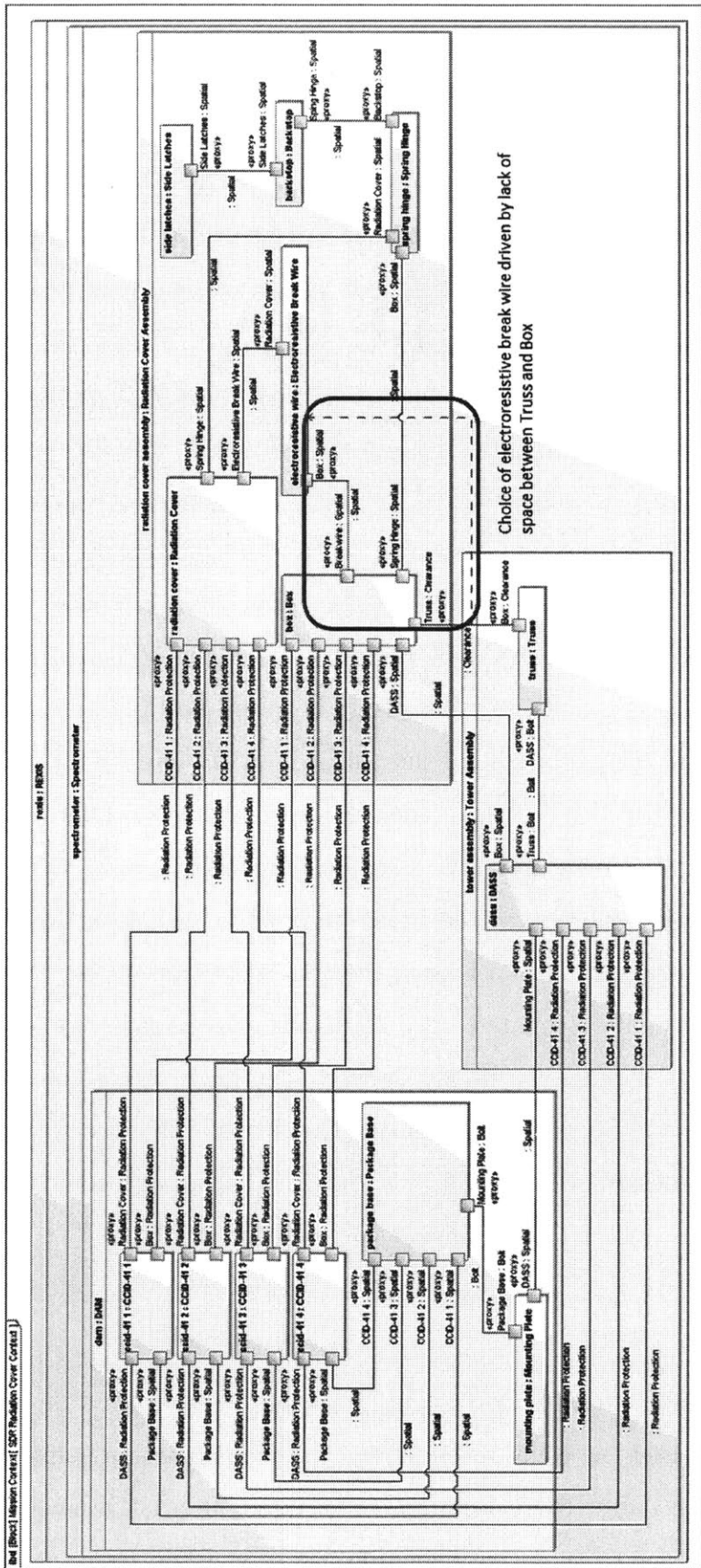


Figure 5-12: Design of the REXIS Radiation Cover at SDR as represented in an Internal Block Diagram

corrected in the PDR design.

### **SysML Model of the Radiation Cover at PDR**

Figure 5-13 shows the historical Radiation Cover design at PDR. The dashed lines show how the effects of design decisions propagate through the rest of the design. The PDR design was of much higher fidelity than the SDR design, as evidenced by the increase in the number of parts. The radiation protection for each CCD is provided by the combination of the Radiation Cover, DASS, Top Shield, Side Shield 1, Side Shield 2, Top End Shield 1, Top End Shield 2, Bottom End Shield 1, and Bottom End Shield 2. The Electroresistive Break Wire has been replaced by the Frangibolt and the hinge design now includes Vespel bushings between the Shaft and the Radiation Cover and the Top Shield. Although the actuator has changed, the choice of actuator is still driven by the size of the space between Side Shield 1 where the actuator is mounted and the -Y Truss Panel (blue box in Figure 5-13). In contrast to the SDR design, the «Accessibility» relationship between the Radiation Cover Assembly and the Environment now exists. Following the purple path in Figure 5-13, the consequences of that relationship flow to the -Y Truss Panel, in which there is a hole to permit access, and then to the Truss Hole Cover that covers the hole for flight.

The PDR SysML model shows that the choice to put the Radiation Cover on the DAM caused the awkward design feature of a hole in the truss to permit access to the cover. The whole reset process was underestimated until PDR where risks like contamination of the CCDs and foreign object damage to the CCDs were identified. Access to a system model showing the ramifications of placing the Radiation Cover inside the truss could have alerted the REXIS team to the design accommodations and risks earlier and could have caused the move of the cover to the mask frame to have occurred before PDR.

### **Summary**

The SysML model of the Radiation Cover could have helped to avoid a design oversight at SDR and could have caused the cover to be moved to the mask frame before PDR. Historically, the cover was moved after PDR. Figure 5-14 shows an alternative, model based design evolution with the historical CDR design being discovered at PDR. Based on the alternative





timeline, it can be concluded that incorporating information from design consequence tracing reduces rework and design iteration and makes the design process more efficient.

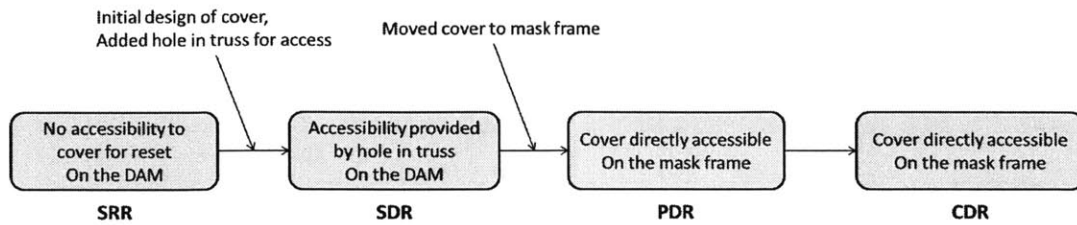


Figure 5-14: A flow chart of an alternative, model based design evolution of the REXIS Radiation Cover. The historical CDR design is identified by PDR, reducing the time wasted analyzing and reviewing unused designs.

# Chapter 6

## Conclusion

### 6.1 Thesis Summary

This thesis examined how access to topological information from a SysML model can improve the design process. The topological information captured in the model included the interfaces between all parts of the system, the uncertainty associated with each interface, and the path along which the consequences of selected design choices or requirements flow. Using statistical data from SysML models of the REXIS instrument constructed at each milestone review, the hypothesis that interface uncertainty can improve the design process was rejected. However, case studies of the REXIS thermal system and Radiation Cover showed that design consequence tracing can provide information that makes the design process more efficient. Therefore, the research hypothesis that applying model-based systems in the design process will result in design process improvements is confirmed.

### 6.2 Generalization of Results

The methodology used in this thesis is not exclusive only to REXIS. Capturing topological information in a system model, including interface uncertainty and design rationale, will help the design process for any space system. For example, the MSL Descent Stage propulsion system experienced development troubles driven partly by the much larger propellant tube diameters. The larger diameters had stiffnesses on the order of the structure, meaning that

heritage propellant tube stress analysis practices could not be used [35]. By tracing the design rationale from the flow rate requirements through the propellant tube diameter size to the necessary test and qualification program, the increased complexity of the system over past designs would have been illuminated. Based on that information, the MSL team could have either pursued alternative designs or emphasized the qualification program early in the schedule, avoiding the late surprises that actually occurred.

### **6.3 Contributions and Future Work**

The contributions of this thesis are the introduction of two new ways to capture topological information in SysML and evidence that design tracing can improve the design process. The new methods of capturing topological information are capturing interface uncertainty through the fraction of generic ports in a subassembly and design consequence tracing. In the REXIS case study, design consequence tracing uncovered design issues before they were discovered in the historical timeline. This result shows that applying model-based systems to the design process reduces rework and shortens design iteration, thereby improving the design process. The work done in this thesis can be extended further by:

- Modeling other space systems that are in development in SysML and quantifying the improvement to the design process by using information from those models
- Capturing the interface uncertainty and design rationale flows of other systems and quantifying the usefulness of that information
- Comparing the REXIS part and interface statistics to that of other systems to develop rules for the proper amount of design detail at each point in the design process
- Use the models in this thesis to predict future REXIS design issues that may arise
- Extend the capture of design rationale flow into an automated design system making sure that all consequences of design choices or requirements are satisfied
- Model system behavior in SysML behavior diagrams to search for other ways to improve the design process by applying MBSE

# Bibliography

- [1] S.J. Kapurch, editor. *NASA Systems Engineering Handbook*. DIANE Publishing, 2007.
- [2] C. Haskins, editor. *INCOSE Systems Engineering Handbook*. INCOSE, 2006.
- [3] Deloitte Consulting. *Can we afford our own future? Why A&D programs are late and over-budget - and what can be done to fix the problem*. Deloitte LLP, 2008.
- [4] S. Friedenthal, R. Griego, and M. Sampson. INCOSE Model Based Systems Engineering (MBSE) Initiative. In *INCOSE 2007 Symposium*, 2007.
- [5] D.L. Emmons, M. Lobbia, T. Radcliffe, and R.E. Bitten. Affordability Assessments to Support Strategic Planning and Decisions at NASA. In *Aerospace Conference, 2010 IEEE*, 2010.
- [6] D.A. Bearden. A complexity-based risk assessment of low-cost planetary missions: when is a mission too fast and too cheap? *Acta Astronautica*, 52(26):371 – 379, 2003.
- [7] E.C. Honour. Understanding the Value of Systems Engineering. In *Proceedings of the INCOSE International Symposium*, pages 1–16, 2004.
- [8] Joseph P Elm and Dennis Goldenson. The Business Case for Systems Engineering Study: Results of the Systems Engineering Effectiveness Survey. Technical report, Carnegie Mellon University, 2012.
- [9] R.J. Hamann, B.T.C. Zandbergen, and P.J. Zijdemans. Does Systems Engineering in Space Projects Pay? In *Proceedings of the 19th Annual International Council on Systems Engineering (INCOSE) International Symposium*, 2009.
- [10] S. Saunders. Reducing Project Re-Work By The Use Of Model Based Systems Engineering Processes and Tools. In *Proceedings of SETE*, 2003.
- [11] G.A. Hazelrigg. A Framework for Decision-Based Engineering Design. *Journal of Mechanical Design*, 120(4):653–658, 1998.
- [12] C. Paredis. Model-Based Systems Engineering: A Roadmap for Academic Research. In *Frontiers in Model-Based Systems Engineering*, 2011.
- [13] Y. Bernard. Requirements Management within a Full Model-Based Engineering Approach. *Systems Engineering*, 15(2):119–139, 2012.

- [14] M.S. Soares and J. Vrancken. Model-driven User Requirements Specification using SysML. *Journal of Software*, 3(6):57–68, 2008.
- [15] M. Russell. Using MBSE to Enhance System Design Decision Making. *Procedia Computer Science*, 8:188–193, 2012.
- [16] K. Bhasin, P. Barnes, J. Reinert, and B. Golden. Applying Model-Based Systems Engineering to NASA’s Space Communications Networks. In *Systems Conference (SysCon), 2013 IEEE International*, 2013.
- [17] T.J. Bayer, S. Chung, B. Cole, B. Cooke, F. Dekens, C. Delp, I Gontijo, K. Lewis, M. Moshir, R. Rasmussen, and D. Wagner. Model Based Systems Engineering on the Europa Mission Concept Study. In *Aerospace Conference, 2012 IEEE*, pages 1–18, 2012.
- [18] T. Bayer, Seung Chung, B. Cole, B. Cooke, F. Dekens, C. Delp, I Gontijo, and D. Wagner. Update on the Model Based Systems Engineering on the Europa Mission Concept Study. In *Aerospace Conference, 2013 IEEE*, 2013.
- [19] S.C. Spangelo, D. Kaslow, C. Delp, B. Cole, L. Anderson, E. Fosse, B.S. Gilbert, L. Hartman, T. Kahn, and J. Cutler. Applying Model Based Systems Engineering (MBSE) to a Standard CubeSat. In *Aerospace Conference, 2012 IEEE*, pages 1–20, 2012.
- [20] Y. Cao, Y. Liu, and C. Paredis. System-Level Model Integration of Design and Simulation for Mechatronic Systems Based on SysML. *Mechatronics*, 21(6):1063–1075, 2011.
- [21] H. Kim, D. Fried, P. Menegay, G. Soremekun, and C. Oster. Application of Integrated Modeling and Analysis to Development of Complex Systems. *Procedia Computer Science*, 16(0):98 – 107, 2013. 2013 Conference on Systems Engineering Research.
- [22] D. Kaslow, G. Soremekun, H. Kim, and S. Spangelo. Integrated Model-Based Systems Engineering (MBSE) Applied to the Simulation of a CubeSat Mission. In *Aerospace Conference, 2014 IEEE*, pages 1–14, 2014.
- [23] M. Giffin, O. de Weck, G. Bounova, R. Keller, C. Eckert, and P.J. Clarkson. Change Propagation Analysis in Complex Technical Systems. *Journal of Mechanical Design*, 131(8), 2009.
- [24] T.A.W. Jarratt, C.M. Eckert, N.H.M. Caldwell, and P.J. Clarkson. Engineering change: an overview and perspective on the literature. *Research in Engineering Design*, 22(2):103–124, 2011.
- [25] E. Fricke, B. Gebhard, H. Negele, and E. Igenbergs. Coping with Changes: Causes, Findings, and Strategies. *Systems Engineering*, 3(4):169–179, 2000.
- [26] P John Clarkson, Caroline Simons, and Claudia Eckert. Predicting change propagation in complex design. *Journal of Mechanical Design*, 126(5):788–797, 2004.

- [27] B. Morkos and J.D. Summers. Requirement Change Propagation Prediction Approach: Results From An Industry Case Study. In *ASME 2010 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages 111–121, 2010.
- [28] OMG. OMG System Modeling Language. <http://www.omgsysml.com>, 2012. Accessed 08-07-2014.
- [29] S. Friedenthal, A. Moore, and R. Steiner. *A Practical Guide to SysML: The Systems Modeling Language*. Elsevier, 2011.
- [30] T.U. Pimmler and S.D. Eppinger. *Integration Analysis of Product Decompositions*. Alfred P. Sloan School of Management, Massachusetts Institute of Technology Cambridge, MA, 1994.
- [31] D. Lauretta and OSIRIS-REx Team. An Overview of the OSIRIS-REx Asteroid Sample Return Mission. *Lunar and Planetary Institute Science Conference Abstracts*, 43:2491, 2012.
- [32] J. Goldsten, R. McNutt, R. Gold, S. Gary, E. Fiore, S. Schneider, J. Hayes, J. Trombka, S. Floyd, W. Boynton, S. Bailey, J. Brückner, S. Squyres, L. Evans, P. Clark, and R. Starr. The X-ray/Gamma-ray Spectrometer on the Near Earth Asteroid Rendezvous Mission. *Space Science Reviews*, 82:169–216, 1997.
- [33] S. Barthelmy, L. Barbier, J. Cummings, E. Fenimore, N. Gehrels, D. Hullinger, H. Krimm, C. Markwardt, D. Palmer, A. Parsons, G. Soto, M. Suzuki, T. Takahashi, M. Tashiro, and J. Tueller. The Burst Alert Telescope (BAT) on the SWIFT Midex Mission. *Space Science Reviews*, 120:143–164, 2005.
- [34] D.E. Whitney, Q. Dong, J. Judson, and G. Mascoli. Introducing knowledge-based engineering into an interconnected product development process. In *Proceedings of the 1999 ASME Design Engineering Technical Conferences*, 1999.
- [35] A. Eppers, M. Rober, D. Lee, C. Guernsey, M. Long, and M. Knopp. Mars Science Laboratory Descent Stage Propulsion Tubing Configuration and Design. In *Aerospace conference, 2009 IEEE*, pages 1–10, 2009.





# Appendix A

## REXIS Requirements

This appendix presents the current version of the REXIS Measurement and Level 4 requirements.

### A.1 REXIS Measurement Requirements

This section presents the Level 2 and Level 3 Measurement requirements.

#### A.1.1 Level 2 Measurement Requirements

**ID: REX-1 Meteorite Group Classification**

REXIS shall measure global elemental abundances of asteroid Bennu with sufficient sensitivity to distinguish the mean global composition of the asteroid among major meteorite groups, including CI, CM, CV carbonaceous chondrites for a minimum A3.3 solar state and 4 MK single temperature incident X-ray spectrum.

**ID: REX-2 Iron Abundance Map Sensitivity**

REXIS shall spatially resolve, on a scale of  $< 50$  m and to at least 6 sigma,  $> 4x$  enhancements in the Fe concentration of Bennu relative to the baseline composition of a CI chondrite assuming a 5.5 MK solar coronal temperature (A8.0 state).

**ID: REX-226 Iron/Magnesium Ratio Map Sensitivity**

REXIS shall spatially resolve, on a scale  $< 100$  m, the Fe/Mg ratio of Bennu to within 50%

accuracy, assuming the baseline composition of a CI chondrite and a 5.5 MK solar coronal temperature (A8.0 state).

**ID: REX-3 Global Element Ratio Measurement Accuracy**

REXIS shall be able to measure the global ratios of Mg/Si, Fe/Si, and S/Si of Bennu within 25% for that of a CI chondrite illuminated by a single temperature Sun at 4 MK (A3.3 state).

**ID: REX-6 Observation Time**

REXIS shall meet performance requirements given no less than 420 hours of observation time of Bennu.

**ID: REX-7 Coronal Temperature Measurement Accuracy**

REXIS shall measure solar coronal temperature to within 0.1 MK at least every 60 sec while observing Bennu assuming a single temperature model of the solar corona.

### **A.1.2 Level 3 Measurement Requirements**

**ID: REX-8 Spectrometer Spectral Resolution**

REXIS shall have a spectral resolution (FWHM) as follows:

< 198 eV @ 0.71 keV (Fe-L)

< 204 eV @ 1.25 keV (Mg-K)

< 210 eV @ 1.74 keV (Si-K)

< 227 eV @ 2.31 keV (S-K)

< 260 eV @ 5.9 keV (Mn-K)

**ID: REX-9 Spectrometer Field of View**

REXIS shall have a field of view (FOV) of 0.25 steradian.

**ID: REX-11 Detector Array Effective Area**

REXIS shall have an effective area as follows:

0.6-0.8 keV: > 6.75 cm<sup>2</sup>

1.15-1.35 keV: > 14.85 cm<sup>2</sup>

1.65-1.85 keV: > 13.27 cm<sup>2</sup>

2.2-2.4 keV: > 15.30 cm<sup>2</sup>

5.6-6.2 keV: > 11.25 cm<sup>2</sup>

**ID: REX-12 Detection Efficiency**

REXIS shall detect X-ray events at 0.5 to 7.5 keV with on-axis quantum efficiencies as follows:

0.6-0.8 keV: > 0.30

1.15-1.35 keV: > 0.66

1.65-1.85 keV: > 0.59

2.2-2.4 keV: > 0.68

5.6-6.2 keV: > 0.5

**ID: REX-13 Spatial Monitoring of Stable Gain During Science Observations**

REXIS shall monitor the spatial variation of the gain drift of the detector array with 15 eV accuracy every 10 hrs while the radiation cover is open.

**ID: REX-14 Global Monitoring of Stable Gain During Science Observations**

REXIS shall monitor the global gain drift of the detector array with 15 eV accuracy every hour while the radiation cover is open.

**ID: REX-15 Spatial Monitoring of Stable Gain During Internal Calibration**

REXIS shall measure the gain for every 200x200 pixel section of the detector array with 15 eV accuracy before the start of science observations of Bennu.

**ID: REX-20 Event Digitization**

REXIS shall convert X-ray events to a digital signal.

**ID: REX-21 Frame Readout Cadence**

REXIS shall read out a frame every 4 seconds.

**ID: REX-23 Internal Clock Accuracy**

REXIS shall maintain an internal clock that is accurate to within 0.01 sec of the space-craft clock 98.5% of the time.

**ID: REX-24 Maximum Continuous Data Acquisition Period**

REXIS shall be able to take data continuously for 24 hours/day.

**ID: REX-25 Temperature of Detector Array**

The REXIS detector array shall be passively cooled to < -60 °C during operation.

**ID: REX-26 Component Operational Temperature Range**

While operating, the temperature of all REXIS components shall be maintained within operability limits.

**ID REX-27 Component Survival Temperature Range**

All REXIS components shall always be maintained at a temperature within their survivability range.

**ID: REX-33 Detector Array Total Non-Ionizing Dose (TNID)**

The REXIS detector array shall tolerate a total non-ionizing dose of  $1.9 \times 10^8$  MeV/g before the opening of the radiation cover during Detailed Survey.

**ID: REX-41 Mission Life**

REXIS shall meet full performance requirements through the end of the nominal Sample Collection phase.

**ID: REX-42 Internal Background**

Internal background flux shall be less than  $< 20$  counts/second in the energy range of interest.

**ID: REX-43 Mask-Detector X&Y Translational Deformation**

The mask shall not shift in relation to any CCD in the x or y directions by more than 0.467 mm.

**ID: REX-44 Mask-Detector Z Translational Deformation**

The mask shall not shift in relation to any CCD in the z direction by more than 1.047 mm.

**ID: REX-45 Mask-Detector X&Y Rotational Deformation**

The mask shall remain parallel to any CCD within a  $0.372^\circ$  rotation about the x and y axes.

**ID: REX-46 Mask-Detector Z Rotational Deformation**

The mask shall not rotate about the z-axis in relation to any CCD by more than  $0.497^\circ$ .

**ID: REX-73 SXM Spectral Resolution**

The SXM shall have a spectral resolution (FWHM) that is less than 200 eV from 1 to 4 keV.

**ID: REX-74 SXM Quantum Efficiency (QE)**

The SXM shall detect X-ray events from 1 to 4 keV with on-axis quantum efficiencies as follows:

1.0 - 1.2 keV:  $> 0.09$

1.6 - 2.8 keV:  $> 0.65$

2.8 - 4 keV:  $> 0.85$

**ID REX-75 SXM Field of View (FOV)**

The SXM shall have a full width zero intensity (FWZI) FOV of no greater than 60° full cone and a full width full intensity (FWFI) FOV of no less than 10° full cone.

**ID: REX-225 SXM Integration Time**

The SXM shall integrate counts for 32 seconds to form a spectrum.

**ID: REX-236 SXM Gain Stability**

The SXM gain shall drift by no more than 5% over an integration period.

**ID: REX-237 SXM Calibration Requirement**

From the SXM data, the gain corresponding to the 1.9 keV solar spectral lines shall be determined to 5% accuracy over each consecutive 5 integration time period. The gain corresponding to the 5.9 keV solar spectrum line will be determined to 5% accuracy over each consecutive 5 integration time period.

**ID: REX-238 Range of Solar States**

The SXM shall be able to measure a solar spectrum from an A1.0 to an A8.0 solar state. The SXM will be able to provide useful solar spectra for flaring up to a B7.0 state.

## **A.2 REXIS Level 4 Requirements**

This section presents the Level 4 requirements on the Detectors, Electronics, Structure, and Flight Software.

### **A.2.1 Level 4 Detector Requirements**

**ID: REX-47 Detector Noise**

The total noise of the detector shall be less than:

15.7 e-/pixel/readout RMS @ 0.71 keV (Fe-L)

16.1 e-/pixel/readout RMS @ 1.25 keV (Mg-K)

16.5 e-/pixel/readout RMS @ 1.74 keV (Si-K)

18.2 e-/pixel/readout RMS @ 2.31 keV (S-K)

20.5 e-/pixel/readout RMS @ 5.9 keV (Mn-K)

**ID: REX-48 Detector Integration Time Reconfigurability**

Detector frame integration time shall be reconfigurable by command.

**ID: REX-51 Detector Spectral Resolution**

The detectors shall have a spectral resolution (FWHM) as follows:

< 139 eV @ 0.71 keV (Fe-L)

< 146 eV @ 1.25 keV (Mg-K)

< 153 eV @ 1.74 keV (Si-K)

< 170 eV @ 2.31 keV (S-K)

< 206 eV @ 5.9 keV (Mn-K)

**ID: REX-52 Detector Quantum Efficiency**

The detectors shall detect X-ray events at 0.5 to 7.5 keV with quantum efficiencies as follows:

0.6-0.8 keV: > 0.70

1.15-1.35 keV: > 0.80

1.65-1.85 keV: > 0.78

2.2-2.4 keV: > 0.77

5.6-6.2 keV: > 0.51

**ID: REX-52 Optical Light Reduction**

The OBF shall reduce the flux of photons between 100 and 10000 nm that encounter the CCD top surface such that less than 1e-7% of photons in that band are detected by the CCD.

**ID: REX-55 Detector Array Stable Gain**

The rate of change of the global gain of the REXIS detector array shall be less than 15 eV/hr.

**ID: REX-56 Detector Effective Area**

Each REXIS detector shall meet the quantum efficiency requirement over at least 90% of its area.

**ID: REX-57 CCD Survival Temperature Range**

The temperature of the CCDs shall always be greater than -120 °C and less than 85 °C.

**ID: REX-58 CCD Lower Operation Temperature**

The temperature of the CCDs shall be greater than -120 °C while operating.

**ID: REX-59 CCD Temperature Variance for Calibration**

The temperature of the CCDs shall not vary by more than  $\pm 2.5$  °C per hour while operating.

**ID: REX-62 Pre Phase 5A Boresight Calibration**

REXIS shall conduct one boresight calibration before Phase 5B and after Phase 2 by localizing a celestial point source within 2.7 arcminutes.

**ID: REX-63 Pre Phase 5A Flux Calibration**

REXIS shall conduct one flux calibration before Phase 5B but after Phase 2 by measuring the flux better than 5% accuracy in 30 eV bins on average over the 0.5-7 keV range.

**ID: REX-64 Post Phase 5A Boresight Calibration**

REXIS shall conduct one boresight calibration after Phase 5B by localizing a celestial point source within 2.7 arcminutes.

**ID: REX-65 Post Phase 5A Boresight Calibration**

REXIS shall conduct one flux calibration after Phase 5B by measuring the flux better than 5% accuracy in 30 eV bins on average over the 0.5-7 keV range.

**ID: REX-232 Internal Background Calibration**

REXIS shall conduct an internal background calibration by collecting data before opening the radiation cover for 5% accuracy in 30 eV bins on average over the 0.5-7 keV range.

**ID: REX-233 External Background Calibration**

REXIS shall conduct an external background calibration before the boresight calibration by collecting data with radiation cover open, staring at no bright X-ray source for flux measurement better than 5% accuracy in 30 eV bins on average over the 0.5-7 keV range.

**ID: REX-66 Monitoring CCD noise dependent on gain variation during Phase 5B**

During Phase 5B, the Fe-55 source shall monitor all four output nodes of each CCD.

**ID: REX-67 Monitoring CTI dependent gain variation during Phase 5B**

During Phase 5B, the Fe-55 source shall monitor CTI variation across the CCD readout direction.

**ID: REX-68 Fe-55 Source Counts During Phase 5B**

The count rate of all the Fe-55 sources as measured by the CCD shall be less than 2 counts/sec.

**ID: REX-69 Detector Health**

The detectors shall survive and remain operational through Phase 8.

**ID: REX-70 Detector Operation Time**

The detectors shall be able to operate continuously for 24 hours/day.

**ID: REX-71 Molecular Contamination**

The maximum mass density of molecular contamination on the detector at the end of the mission shall be less than  $0.00054 \text{ g/cm}^2$  assuming a C6O composition.

**ID: REX-72 Particulate Contamination**

The maximum percentage of detector area obscured by particulate contamination at the end of mission shall be less than 0.25%.

**ID: REX-227 Pin Holes in OBF**

The percentage of detector area made unavailable for data collection due to pinholes in the OBF shall be less than 0.75%.

**ID: REX-76 SXM Preamp Survival Temperature**

The temperature of the SXM preamp shall always be greater than  $-55 \text{ }^\circ\text{C}$  and less than  $+85 \text{ }^\circ\text{C}$ .

**ID: REX-77 SXM Preamp Operating Temperature**

The temperature of the SXM preamp shall be greater than  $-40 \text{ }^\circ\text{C}$  and less than  $+85 \text{ }^\circ\text{C}$  while operating.

**ID: REX-223 SDD Survival Temperature Range**

The temperature of the SDD shall always be greater than  $-65 \text{ }^\circ\text{C}$  and less than  $150 \text{ }^\circ\text{C}$ .

**ID: REX-229 SDD Operation Temperature Range**

The temperature of the SDD shall be less than  $-30 \text{ }^\circ\text{C}$  and greater than  $-70 \text{ }^\circ\text{C}$ .

**ID: REX-78 SXM Health**

The SXM shall survive and remain operational through the end of Phase 8.

**ID: REX-79 SXM Operation Time**

The SXM shall be capable of operating 24 hours/day.

## **A.2.2 Level 4 Electronics Requirements**

**ID: REX-98 S/C Power Control**



The MEB shall be powered on/off on by the spacecraft.

**ID: REX-99 MEB Power Distribution**

The MEB shall distribute power for REXIS during all phases of the mission.

**ID: REX-100 MEB Operation Time Limit**

The MEB shall be able to operate continuously for 24 hours per day.

**ID: REX-101 Detector Power**

The main electronics board (MEB) shall provide +24VDC, +12VDC, -12VDC, +5VDC, and +3.3VDC to the detector electronics (DE).

**ID: REX-102 SXM Preamp Power**

The MEB shall provide +5VDC, and -5VDC and -110VDC to the Solar X-ray Monitor Preamp.

**ID: REX-103 FPGA Power**

The MEB shall provide +3.3VDC, +2.5VDC, and +1.0VDC to the radiation-hardened Xilinx Virtex-5 FPGA.

**ID: REX-104 Radiation Cover Release Mechanism Power**

The MEB shall provide a minimum of +9VDC to the radiation cover release mechanism.

**ID: REX-127 MEB-Supplied SXM TEC Voltage**

The MEB shall provide a variable voltage between 0 and 5VDC to the SXM preamp to power the TEC.

**ID: REX-105 C&DH Interface with OSIRIS-REx**

The MEB shall provide the REXIS Command and Data Handling interface to the OSIRIS-REx spacecraft in accordance with the OSIRIS-REx to REXIS IRCD.

**ID: REX-106 Processing Unit**

The MEB shall provide a central processing unit and supporting environment for Flight Software.

**ID: REX-107 Non-volatile Memory**

The MEB shall provide non-volatile, writeable memory.

**ID: REX-108 Memory Size**

The MEB shall provide volatile memory sufficient to store at least two full frames from the CCDs .

**ID: REX-109 Interface with Detector Electronics**

The MEB shall provide the control and data handling interface to the Detector Electronics.

**ID: REX-110 Camera Link Interface**

The MEB shall implement a Camera Link interface for frame data transfer from the Detector Electronics.

**ID: REX-111 Readout Format**

The MEB shall store the CCD frame data in volatile memory.

**ID: REX-112 Control Interface with Detector Electronics**

The MEB shall implement a UART interface for control of the Detector Electronics.

**ID: REX-113 Control of Radiation Cover Release Mechanism**

The MEB shall provide control capability for the Radiation Cover Release Mechanism.

**ID: REX-114 Redundancy in Radiation Cover Release Mechanism**

The MEB shall include redundant Radiation Cover Release Mechanism control hardware to prevent inadvertent activation of the Radiation Cover Release Mechanism.

**ID: REX-116 Housekeeping Data Collection**

The MEB shall provide capability to acquire Housekeeping data.

**ID: REX-117 Interface with SXM**

The MEB shall provide the control and data interface for the Solar X-ray Monitor.

**ID: REX-118 Image Processing Requirement**

REXIS shall process the x-ray image and generate x-ray event results using the onboard radiation-hardened Xilinx Virtex-5 FPGA.

**ID: REX-120 Read Noise**

The Detector Electronics shall produce less than 7 electrons/pixel/readout RMS.

**ID: REX-121 Detector Electronics Health**

The Detector Electronics shall survive and remain operational through the end of Sample Collection Phase.

**ID: REX-122 Detector Electronics Operation Time**

The Detector Electronics shall be capable of operating 24 hours/day.

**ID: REX-123 MEB Survival Temperature**

The temperature of the MEB shall always be greater than  $-40\text{ }^{\circ}\text{C}$  and less than  $+80\text{ }^{\circ}\text{C}$  (TBR).

**ID: REX-124 MEB Operating Temperature**

The temperature of the MEB shall be greater than  $-25\text{ }^{\circ}\text{C}$  and less than  $+50\text{ }^{\circ}\text{C}$  while operating (TBR).

**ID: REX-125 Detector Electronics Survival Temperature**

The temperature of the Detector Electronics (DE) shall always be greater than  $-40\text{ }^{\circ}\text{C}$  and less than  $+80\text{ }^{\circ}\text{C}$  (TBR).

**ID: REX-126 Detector Electronics Operating Temperature**

The temperature of the Detector Electronics shall be greater than  $-25\text{ }^{\circ}\text{C}$  and less than  $+50\text{ }^{\circ}\text{C}$  while operating (TBR).

### **A.2.3 Level 4 Structures Requirements**

**ID: REX-80 Mask Transmission**

The mask shall have a transmission in the fully coded FOV of no less than 40%.

**ID: REX-81 Mask Pixel Size**

The mask shall have a pixel size of 1.536 mm.

**ID: REX-82 Mask Sampling Requirement**

The ratio of mask pixel size to detector pixel size shall be greater than 2.

**ID: REX-83 Mask Opacity**

Opaque parts of the mask shall block at least 99.9% of the X-ray flux across the energy range of interest.

**ID: REX-84 Coding Noise**

The maximum attainable SNR in using the mask shall be no less than 40.

**ID: REX-234 Focal Length**

The distance between the mask and any CCD shall be 20 cm.

**ID: REX-235 Mask Diameter**

The diameter of the coded aperture mask shall be 9.84 cm.

**ID: REX-85 Radiation Cover Deployment Reliability**

The radiation cover shall open at least 5 consecutive times in flight-like environmental conditions before launch.

**ID: REX-86 Radiation Cover Thickness**

The radiation cover shall be thicker than 5.416 mm Al equivalent.

**ID: REX-87 Radiation Cover Opening**

The radiation cover shall open prior to the observation of an external X-ray source.

**ID: REX-226 Actuation Time**

The radiation cover shall open within 2 minutes of application of power to the Frangibolt.

**ID: REX-88 Frangibolt Survival Temperatures**

The temperature of the Frangibolt shall always be greater than  $-40\text{ }^{\circ}\text{C}$  and less than  $85\text{ }^{\circ}\text{C}$ .

**ID: REX-89 Frangibolt Operating Temperature**

The temperature of the Frangibolt shall be greater than  $-40\text{ }^{\circ}\text{C}$  and less than  $105\text{ }^{\circ}\text{C}$  while operating.

**ID: REX-90 Hinge Survival Temperature**

The temperature of the hinge shall always be greater than  $-40\text{ }^{\circ}\text{C}$  and less than  $100\text{ }^{\circ}\text{C}$ .

**ID: REX-91 Hinge Operating Temperature**

The temperature of the hinge shall be greater than  $-40\text{ }^{\circ}\text{C}$  and less than  $100\text{ }^{\circ}\text{C}$  while operating.

**ID: REX-92 Electronics Box Survival Temperature Range**

The temperature of the Electronics Box shall always be greater than  $-40\text{ }^{\circ}\text{C}$  at the center of the baseplate on the inside and less than  $80\text{ }^{\circ}\text{C}$  on the center of the baseplate on the inside.

**ID: REX-93 Electronics Box Operating Temperature Range**

The temperature of the Electronics Box shall be greater than  $-25\text{ }^{\circ}\text{C}$  at the center of the baseplate on the inside and less than  $50\text{ }^{\circ}\text{C}$  at the center of the baseplate on the inside while any component within is operating.

**ID: REX-230 Locking Fasteners**

REXIS shall use a locking feature for all threaded fasteners.

**ID: REX-231 Friction Fit**

REXIS shall use a stable, positive location system where precision component location is needed.

**ID: REX-90 Structural Fluorescence**

The structural fluorescence coating shall attenuate at least 99.99% of the x-ray photons generated by structural fluorescence.

**A.2.4 Level 4 Flight Software Requirements****ID: REX-127 Safe Mode**

The FSW shall provide a safe mode in which the instrument is in a known powered state with the detectors and optics protected from damage or degradation, as specified in Section 4 of the REXIS Science Modes Document (REX-CD-021).

**ID: REX-128 System Initialization**

The FSW shall enter a safe configuration (safe mode) upon system initialization.

**ID: REX-129 Safe Mode Power**

When in safe mode, only the following REXIS components shall be powered on, by default: +5V regulator, +3.3V, +2.5V, +1.0V, regulators.

**ID: REX-130 Safe Mode CCD Power**

When in safe mode, the following REXIS CCD voltages shall be powered off by default: +/- 12V supply, +24V supply.

**ID: REX-131 Safe Mode SXM Power**

When in safe mode, the following REXIS SXM voltages shall be powered off: - 5V regulator, - 110V reference, and the voltage across the SDD TEC.

**ID: REX-132 Frangibolt Actuation**

The Frangibolt power shall only be activated in Safe Mode.

**ID: REX-133 Safe Mode Commands**

The list of commands acceptable in Safe Mode shall be specified in the Command and Telemetries Dictionary (REX-CD-032). No other commands shall be accepted.

**ID: REX-134 Science Mode**

The FSW shall provide a Science Mode in which the instrument is in a state that allows collection of images...etc, as specified in the REXIS Science Modes Document (REX-CD-021).

**ID: REX-135 Science Mode Tasks**

Internal Calibration, External Calibration, and Science observations shall occur during science mode.

**ID: REX-136 Science Mode Power**

When in science mode, the following REXIS components shall be powered on, by default: +5V regulator, +3.3V, +2.5V, +1.0V, regulators.

**ID: REX-137 Science Mode CCD power**

When in science mode, the following REXIS CCD voltages shall be powered on: +/- 12V supply, +24V supply.

**ID: REX-138 Science Mode SXM power**

When in safe mode, the following REXIS SXM voltages shall be powered on: - 5V regulator, - 110V reference, and the voltage across the SDD TEC.

**ID: REX-139 Science Mode TEC power**

Upon entering science mode, the default duty cycle for the SXM TEC PWM signal shall be zero.

**ID: REX-140 Science Mode TEC control**

The duty cycle for the SXM TEC shall be adjustable by FSW to control SDD temperature.

**ID: REX-141 Science Mode Commands**

The list of commands acceptable in Science Mode shall be specified in the Command and Telemetries Dictionary (REX-CD-032). No other commands shall be accepted.

**ID: REX-142 Mode Transition-Science**

The FSW shall transition from safe mode to science mode by command.

**ID: REX-143 Mode Transition-Safe**

The FSW shall transition from science mode to safe mode by command.

**ID: REX-144 Safe Mode Transition-Fault**

The FSW shall transition from science mode to safe mode after certain fault detections, IAW the REXIS Fault Handling Document (REX-CD-033).

**ID: REX-145 Mode Specific Commands**

If REXIS receives a command not applicable to its current operational mode, the FSW shall reject the command IAW the REXIS Fault Handling Document (REX-CD-033).

**ID: REX-146 Mode Transition during data collect**

If the REXIS FSW should receive a “change to safe mode” command while the instrument is processing a frame, then it shall change mode after the current frame has been processed.

**ID: REX-147 FSW IRCD Compliance**

The FSW shall receive and transmit packetized data to the spacecraft bus in accordance with the OSIRIS-REx - REXIS IRCD (GSFC Doc# PLA-OSIRIS-REx-ICD-0005).

**ID: REX-148 Command structure**

The FSW shall receive and transmit commands and tlm data with the spacecraft bus using the header structure as specified in Section 3.2.8.4.3 of the OSIRIS-REx - REXIS IRCD (GSFC Doc# PLA-OSIRIS-REx-ICD-0005).

**ID: REX-151 Command Definitions**

The REXIS FSW shall process commands IAW the Command and Telemetries Dictionary (REX-CD-032).

**ID: REX-154 Command precedence**

REXIS FSW shall process commands in the order in which they are received.

**ID: REX-155 Command Definitions**

The FSW shall be capable of receiving and executing all commands defined in the Command and Telemetry Dictionary.

**ID: REX-156 Invalid Opcode**

The FSW shall reject commands with an invalid opcode.

**ID: REX-157 Out of Range**

The FSW shall reject commands that contain parameters with out of range values, the list parameter ranges are defined in the command dictionary.

**ID: REX-158 Incorrect Length**

The FSW shall reject commands that have mismatched packet length and IDP length field.

**ID: REX-159 Invalid Checksum**

The FSW shall reject commands that have an invalid or incorrect IDP checksum value (CRC).

**ID: REX-160 Illegal in Current State**

The FSW shall reject commands that are illegal in the current state of the instrument, as defined in the Command and Telemetries Dictionary.

**ID: REX-161 Telemetry Definitions**

The REXIS FSW shall generate telemetry messages IAW the Command and Telemetry Dictionary (REX-CD-032).

**ID: REX-164 Current Mode Indication**

The FSW shall indicate in housekeeping telemetry the current instrument mode.

**ID: REX-165 Science Telemetry**

When in Science mode, transmission of science telemetry shall be enabled and disabled by command.

**ID: REX-166 Science Telemetry**

Upon entering Science Mode science telemetry transmission is disabled by default.

**ID: REX-167 Fault Telemetry**

The REXIS FSW shall generate fault telemetry messages IAW the REXIS Fault Handling Document (REX-CD-033).

**ID: REX-168 Aliveness Message**

When in safe mode, the FSW shall transmit aliveness messages to the spacecraft in the form of a NOOP message. The Structure of NOOP message is defined in the REXIS Command and Telemetry Dictionary (REX-CD-032).

**ID: REX-169 Science Mode Aliveness Message**

When in Science Mode and science telemetry transmission is enabled, the FSW shall transmit aliveness messages to the s/c in the form of a science data packet.

**ID: REX-170 Science Mode Aliveness Message**

When in Science Mode and science telemetry transmission is disabled, the FSW shall transmit an aliveness message in the form of a NOOP message.

**ID: REX-171 Aliveness Message Frequency**

The default frequency for the transmission of aliveness messages to the s/c is once every 4 seconds.

**ID: REX-172 Aliveness Message Frequency**

The frequency for the aliveness messages shall be adjustable via command.

**ID: REX-173 Time Tagged Events**

The FSW shall implement time tagged event logging.



**ID: REX-174 Time Tagged Events**

The time tagged event logging shall be downlinked via command from SC.

**ID: REX-175 Time Tagged Events**

The event log shall be stored in a default location in SDRAM.

**ID: REX-176 Time Tagged Events**

The location in SDRAM where the logs are kept shall be configurable via command.

**ID: REX-177 Time Tagged Events**

FSW shall delete log entries in event when log is filled, oldest first.

**ID: REX-178 Time Tag Accuracy**

The FSW shall time tag science and telemetry packets accurate to 20.0 ms relative to the spacecraft clock.

**ID: REX-179 Internal Clock Format**

The FSW shall maintain an internal clock consisting of a 32 bit "seconds" field, with resolution of 1 second per bit, and a 16 bit "subsecond" field with resolution of  $1/(2^{16})$  sec per bit.

**ID: REX-180 Internal Clock Syncing**

The FSW shall sync its internal clock to the spacecraft time based on the time update, which will occur at a frequency specified in the OSIRIS-REx - REXIS IRCD (GSFC Doc# PLA-OSIRIS-REx-ICD-0005).

**ID: REX-181 Missed Time Update**

The FSW shall increment its internal clock in the event of a missing time update.

**ID: REX-182 Time Update Processing**

The FSW shall process the time update before the time tick for that update is received.

**ID: REX-183 Clock Updating**

The FSW shall update its internal clock at the time the time tick is received, with the time received in the last time update.

**ID: REX-184 Indication of Missed Time Update**

The FSW shall include in telemetry an indication of a missed time update.

**ID: REX-185 Indication of Missed Time Tick**

The FSW shall include in telemetry an indication of a missed time tick.

**ID: REX-186 Internal Operations**

The FSW shall continue operations based on an internal timer if the spacecraft time sync pulse is not received.

**ID: REX-187 Time Tag Accuracy**

The FSW shall affix time tags accurate to +/- 0.02 seconds of OSIRIS-REx spacecraft time, 98.5% of the time, to each packet it sends to the OSIRIS-REx spacecraft.

**ID: REX-192 Frangibolt Firing Procedure**

The FSW shall provide ARM and FIRE commands to actuate the Frangibolt radiation cover release mechanism.

**ID: REX-193 Frangibolt Firing Procedure**

The FSW shall verify that the Frangibolt ARM command has been sent before the FIRE command is processed.

**ID: REX-195 CCD Integration time**

The CCDs in the REXIS detector plane shall operate with an integration time of 4 seconds.

**ID: REX-196 CCD-SXM sync**

The start of a CCD integration period shall be synchronized with the start of an SXM histogram integration period to within 200ms.

**ID: REX-197 Bias map**

Upon entering science mode, the FSW shall generate a bias map and write it to the SDRAM.

**ID: REX-198 Bias Map command**

The FSW shall generate a bias map via command.

**ID: REX-199 Bias map deletion**

When a new bias map is generated the old bias map shall be written over.

**ID: REX-200 Bias map and image processing**

The FSW shall not process individual frames until bias map generation is complete.

**ID: REX-201 Bias map aliveness messages**

The FSW shall send aliveness messages on schedule during bias map generation.

**ID: REX-202 Event finding**

The FSW shall generate an X-ray event list IAW the REXIS Science Modes Document (REX-CD-021).

**ID: REX-203 Event finding**

Upon the assertion of a hardware interrupt, the FSW shall read the volatile memory to access the candidate x-ray events list (CXEL).

**ID: REX-204 Event finding**

The FSW shall implement the event finding algorithm as specified in 5.2.2 of the Science Modes Document (REX-CD-021).

**ID: REX-205 Parameter table**

The FSW shall create a Parameter Table as specified in Table 4 of the REXIS Science Modes Document (REX-CD-021) and store it in SDRAM.

**ID: REX-206 Event Threshold**

The event threshold (ET) value for each node on each CCD shall be stored as an entry in the parameters table.

**ID: REX-207 Event Threshold**

The event threshold (ET) value for each node on each CCD shall be configurable by command.

**ID: REX-208 Event Threshold**

The default ET values in FSW shall be TBR ADU.

**ID: REX-209 CXEL generation**

The FSW shall evaluate each entry in the CXEL to identify the pixel containing the maximum energy within a 3x3 pixel region.

**ID: REX-210 Event grading**

The FSW shall implement the event grading algorithm as specified in 5.2.2.6 of the Science Modes Document (REX-CD-021).

**ID: REX-211 Split Threshold**

The split threshold (ST) value for each node on each CCD shall be stored as an entry in the parameters table.

**ID: REX-212 Split Threshold**

The split threshold (ST) value for each node on each CCD shall be configurable by command.

**ID: REX-213 Split Threshold**

The ST value shall be restricted to values less than the ET value for that particular node.

**ID: REX-214 Split Threshold**

The default ST shall be TBR.

**ID: REX-215 Event Grading**

The event grade shall consist of a single 3-bit value.

**ID: REX-216 Event Grading Flag**

The FSW shall maintain an 8-bit event grade filtering flag.

**ID: REX-217 Event Grading Flag**

The event grade filtering flag shall be adjustable via command.

**ID: REX-218 High Energy Event limit**

The FSW shall maintain a high-energy limit.

**ID: REX-219 High Energy Event limit**

The high-energy limit shall be adjustable via command.

**ID: REX-240 High Energy Event limit**

The default value for the high-energy limit shall be TBR.

**ID: REX-241 Event Grading Flag**

8 bits of the event grade filtering flag shall correspond to the 8 possible event grades.

**ID: REX-242 Graded Events List**

The graded events shall be included in the science data if its corresponding bit in the event grade filtering flag is set to 1.

**ID: REX-243 Graded Events List**

The graded events shall be excluded in the science data if its corresponding bit in the event grade filtering flag is set to 0.

**ID: REX-244 Graded Events Histogram**

The FSW shall maintain a 8x24-bit histogram to keep track of the number of events in each grade.

**ID: REX-245 Event Grading Histogram**

This histogram shall be included in the housekeeping packet.

**ID: REX-246 Event Grading Histogram**

The histogram shall be reset to zero after creation of every housekeeping packet.

**ID: REX-247 Graded Events List**

The graded events shall be packaged for transmission in the format specified in 6.2 of the Science Modes Document (REX-CD-021).

**ID: REX-248 Graded Events List**

The energy value of each event shall be truncated to include the most significant 9 bits.

**ID: REX-249 Event Filtering**

The FSW shall implement the event filtering scheme as specified in Table 1 of the Science Modes Document (REX-CD-021).

**ID: REX-252 Diagnostics**

The subsection of the bias map desired in the diagnostics packet shall be specified via command.

**ID: REX-253 Diagnostics**

The subsection of the raw frame desired in the diagnostics packet shall be specified via command.

**ID: REX-254 Diagnostics**

A diagnostics packet shall be generated via command.

**ID: REX-255 Detector Electronics FPGA Register Read and Write**

The FSW shall provide read and write access to all available Detector Electronics FPGA registers through commands.

**ID: REX-256 CCID-41 Integration Time Configurability**

The FSW shall provide a command to configure the CCID-41 integration times controlled by the Detector Electronics.

**ID: REX-257 CCID-41 Integration Time Default**

The default detector electronics integration time shall be 4 seconds.

**ID: REX-258 CCID-41 Parameter Configurability**

The FSW shall configure the Detector Electronics CCID-41 control parameters stored in Detector Electronics memory.

**ID: REX-259 Detector Electronics Configurations**

The FSW shall command the Detector Electronics for all science configurations described in the REXIS Science Modes Document.

**ID: REX-260 Detector Electronics Status**

The FSW shall provide a command to read status information from the Detector Electronics.

**ID: REX-261 Frame Readout Command**

The FSW shall send a command to the Detector Electronics to read image out from the CCID-41s.

**ID: REX-262 Starting Memory Address**

FSW shall store the image and the bias map separately in the external volatile memory.

**ID: REX-263 Frame Grabber Reset**

FSW shall provide a command to reset the state of Frame Grabber hardware module to its default state.

**ID: REX-264 SXM Data Collection**

Collection of SXM histogram data shall be disabled by default and toggled on/off via command.

**ID: REX-265 SXM Histogram access**

The FSW shall read the specified volatile memory location to access the SXM energy histogram.

**ID: REX-266 Integration Time**

The integration time of the SXM energy histogram shall be adjustable via command.

**ID: REX-267 Histogram Formation**

The default integration time of the SXM energy histogram shall be 32 seconds.

**ID: REX-268 Histogram Packaging**

At the end of each integration interval the histogram shall be package into a SXM histogram data packet and transmitted to the spacecraft.

**ID: REX-269 Histogram binning**

The SXM histogram shall consists of 512 bins (default width of 15.6eV).

**ID: REX-270 Histogram Packaging**

The number of events in each bin shall be represented by a 16-bit word.

**ID: REX-271 Histogram high-energy limit**

The upper energy limit of the histogram shall be adjustable via command.

**ID: REX-272 Histogram high-energy limit**

The default upper energy limit of the histogram shall be 8 keV.

**ID: REX-273 Histogram low-energy limit**

The lower energy limit of the histogram shall be adjustable via command.

**ID: REX-274 Histogram low-energy limit**

The default lower energy limit of the histogram shall be 0.3 keV.

**ID: REX-275 Out of bounds events**

Events falling below the SXM histogram lower energy limit shall be counted in a 2-byte counter.

**ID: REX-276 Out of bounds events**

Events above the upper energy limit shall be counted in a 2-byte counter.

**ID: REX-277 Total events counter**

FSW shall maintain a 4-byte counter to count the total number events counted within an integration period.

**ID: REX-278 Total events counter**

The total events counter shall be reset after every integration interval.

**ID: REX-279 Event overflow**

FSW shall halt the collection of data for the histogram if any event bins in the SXM histogram overflows. An event bin overflows with  $2^{16}-1$  events.

**ID: REX-280 Event overflow**

In case of interruption of the integration due to overflow, the FSW shall record the time since the beginning of current integration.

**ID: REX-281 Total events counter**

In case of interruption, the total events counter shall continue to increment until the end of the integration interval.

**ID: REX-282 Histogram Binning**

The FSW shall be able to include or ignore any bin in the SXM histogram data collection via command.

**ID: REX-283 Histogram Binning**

By default, all bins are included in the SXM histogram data collection.

**ID: REX-284 Histogram reset**

The FSW shall reset the histogram to zero once the SXM data packet has been assembled (and transmitted).

**ID: REX-285 Fault Protection**

REXIS shall implement fault protection IAW the Fault Handling Document (REX-CD-033).

**ID: REX-286 Fault Handling**

REXIS shall detect faults by checking conditions in Table 2 of the Fault Handling Document (REX-CD-033).

**ID: REX-287 Reset Record**

The FSW shall transmit the reset record to the spacecraft upon command.

**ID: REX-288 Reset Command**

The FSW shall provide a reset command that reinitializes all hardware modules and transitions the FSW to Safe Mode.

**ID: REX-289 Safe Me Request**

The FSW shall provide a 'safe me' request IAW with the IRCD (GSFC Doc# PLA-OSIRIS-Rex-ICD-0005) .

**ID: REX-290 Ground Override of Fault Protection**

The FSW shall provide the capability for the ground to override autonomous fault protection functions.

**ID: REX-291 User-Configurability**

The FSW shall implement user-configurable parameters for fault protection functions.

**ID: REX-292 Fault Protection Status**

The FSW shall include in telemetry all fault protection status.

**ID: REX-293 Fault Reporting**

The FSW shall report all faults IAW the Fault Protection Document (REX-CD-033).

**ID: REX-294 Fault Response**

The FSW shall response to all fault conditions IAW Table 2 of the Fault Handling Document (REX-CD-033).

**ID: REX-295 Enable Fault Detection**

The FSW shall provide a command to ENABLE and DISABLE autonomous FSW fault handling.



**ID: REX-296 Default enable fault protection**

REXIS FSW fault detection shall default to an Enabled state after start-up or warm reset.

**ID: REX-297 Failed Temp sensor**

The FSW shall provide a method for marking a temperature sensor as failed.

**ID: REX-298 Failed Temp sensor**

The FSW shall eliminate any failed temperature sensors from use in any fault response or calculations.

**ID: REX-299 Watchdog Reset**

The FSW shall periodically reset the hardware watchdog counter.

**ID: REX-300 Software Version ID**

The FSW shall include in telemetry a unique identifier of the software being executed that is changed with each software update.

**ID: REX-301 Software Uploads**

The FSW shall have the capability for additional auxiliary function uploads.

**ID: REX-303 Integrity of Software Upload**

The FSW shall provide a method of verifying the integrity of the uploaded functions prior to use.

**ID: REX-304 Software at Launch**

The entire FSW image shall be contained in non-volatile memory at launch.

**ID: REX-305 Parameter Table Storage**

The FSW shall store the science parameter table in non-volatile memory.

**ID: REX-306 Read Access to Memory / Memory Dump**

The FSW shall provide read access to all read-accessible memory via command.

**ID: REX-307 Memory Dump**

The data from memory dump shall be downlinked through telemetry (See telemetry dictionary).

**ID: REX-308 Large Data Packets**

If the data to be downlinked is bigger than a telemetry packet, the data shall be split between several telemetry packets.

**ID: REX-309 Write Access to Memory**

The FSW shall provide write access to all write-accessible memory via command.

**ID: REX-310 Memory Write command structure**

The data to be written by the memory write command shall be specified as part of the command (See command dictionary).

**ID: REX-311 CPU Throughput Margin at CDR**

The FSW will maintain a minimum margin for CPU throughput at CDR of 40%.

**ID: REX-312 Memory Margin at CDR**

The FSW will maintain a minimum margin for storage of software instruction and data memory in non-volatile memory at CDR of 40%.

**ID: REX-313 CPU Throughput Margin at Launch**

The FSW will maintain a minimum margin for CPU throughput at launch of 33%.

**ID: REX-314 Memory Margin at Launch**

The FSW will maintain a minimum margin for storage of software instruction and data memory in non-volatile memory at launch of 30%.