



## **MBSE PARA ENGENHARIA DE SISTEMAS DE CUBESATS**

### **RELATÓRIO FINAL DE PROJETO DE INICIAÇÃO CIENTÍFICA (PIBIC/INPE/CNPq)**

Érika Rizzo Aquino (ITA, Bolsista PIBIC/CNPq)  
E-mail: erikarizzoaquino@gmail.com

Geilson Loureiro (LIT/INPE, Orientador)  
E-mail: geilson@lit.inpe.br

#### **COLABORADORES**

Eduardo Escobar Bürger (LIT/INPE)

Julho de 2018

## RESUMO

O presente projeto de Iniciação Científica tem como objetivo a implementação da Engenharia de Sistemas baseada em modelos (MBSE – *Model Based Systems Engineering*) para o nanossatélite AESP-14, primeira plataforma CubeSat brasileira, desenvolvido pelo ITA (Instituto Tecnológico de Aeronáutica) com apoio do INPE (Instituto Nacional de Pesquisas Espaciais). A metodologia MBSE tem o potencial de reduzir ambiguidades na comunicação entre desenvolvedores e stakeholders ao propor uma representação visual, estruturada e facilmente inter-relacionável da engenharia de sistemas de um produto, e por isso tem ganhado popularidade nos últimos anos. Em particular, a aplicação da metodologia em um CubeSat deve-se às grandes potencialidades tecnológicas deste, em uso científico ou comercial. O trabalho pretende estruturar a documentação já existente da engenharia de sistemas do nanossatélite AESP-14 em modelos. Para isso, definiu-se o uso da linguagem SysML (*Systems Modeling Language*), específica para engenharia de sistemas, cujos diagramas permitem o modelamento de duas facetas do sistema: estrutural e comportamental. A execução da modelagem foi realizada no *software Rational Rhapsody*, desenvolvido pela empresa IBM e disponível nos servidores do INPE. A citar, foram feitos diagramas MBSE englobando *stakeholders* e seus objetivos, requisitos, cenário operacional, arquitetura, subsistemas e interfaces, com as correlações adequadas entre eles. Esse conjunto de documentos modelou o CubeSat AESP-14 e pode servir de referência para trabalhos similares.

Palavras-chave: Engenharia de Sistemas; MBSE; SysML; CubeSat.

## LISTA DE ILUSTRAÇÕES

	<b><u>Pág.</u></b>
Figura 5.1 – Requisitos de <i>stakeholders</i> do AESP14 modelado por <i>use case diagram</i>	10
Figura 5.2 – Rastreabilidade e ampliação dos requisitos por <i>requirements diagram</i> ....	11
Figura 5.3 – Modelo físico do AESP14 por <i>block definition diagram</i> .....	13
Figura 5.4 – Linha do tempo após lançamento modelado por <i>sequence diagram</i> .....	15
Figura 5.5 – Mensagens entre elementos do sistema durante operação nominal modelado por <i>sequence diagram</i> .....	16
Figura 5.6 – Transições entre estados e modos modelado por <i>activity diagram</i> .....	17

## LISTA DE SÍMBOLOS E ABREVIATURAS

AEB	Agência Espacial Brasileira
ACT	<i>Activity Diagram</i>
AESP	Aeroespacial
BBD	<i>Block Definition Diagram</i>
CNPq	Conselho Nacional de Desenvolvimento Científico e Tecnológico
COTS	<i>Commercial off-the-shelf</i>
ECSS	<i>European Cooperation for Space Standardization</i>
ENV	<i>Environment</i>
GSE	<i>Ground Support Equipment</i>
IARU	<i>International Amateur Radio Union</i>
IBM	<i>International Business Machines</i>
INCOSE	<i>International Council of Systems Engineering</i>
INPE	Instituto Nacional de Pesquisas Espaciais
ITA	Instituto Tecnológico de Aeronáutica
ITU	<i>International Telecommunication Union</i>
LIT	Laboratório de Integração e Testes
MBSE	<i>Model Based Systems Engineering</i>
PLSV	<i>Polar Satellite Launch Vehicle</i>
POD	<i>Picosatellite Orbital Deployer</i>
RF	Radiofrequência
SD	<i>Sequence Diagram</i>
SysML	<i>Systems Modeling Language</i>
TT	Telemetria e Telecomando
UC	<i>Use Case Diagram</i>
UML	<i>Unified Modeling Language</i>
VEGA	<i>Vettore Europeo di Generazione Avanzata</i>
VLM	Veículo Lançador de Microsatélite
VLS	Veículo Lançador de Satélite

## SUMÁRIO

	<u>Pág.</u>
1 INTRODUÇÃO .....	1
2 OBJETIVOS DO TRABALHO .....	3
3 FUNDAMENTAÇÃO TEÓRICA.....	4
3.1 Diagramas estruturais .....	4
3.2 Diagramas comportamentais.....	6
4 MATERIAIS E MÉTODOS UTILIZADOS .....	7
4.1 Modelamento de Requisito .....	7
4.2 Modelamento de Sistema Físico.....	8
4.3 Modelamento de Processo .....	8
5 ANÁLISES E RESULTADOS .....	10
5.1 Diagramas para modelamento de requisito .....	10
5.2 Diagrama para modelamento de sistema físico .....	11
5.3 Diagramas para modelamento de processo .....	14
5.4 Discussão sobre utilização de MBSE para engenharia de sistema de CubeSat .....	17
6 CONCLUSÃO.....	19
REFERÊNCIAS BIBLIOGRÁFICAS.....	20

## 1 INTRODUÇÃO

O CubeSat é um tipo de nanossatélite, composto por unidades cúbicas de 10 cm de lado, e com massa de até 1,33 kg. É normalmente desenvolvido com componentes de prateleira (COTS – *Commercial off-the-shelf*), e lançado como carga secundária, isto é, afixado a carga primária de um foguete. A integração entre a plataforma CubeSat e o *payload* primário é feito por meio de um dispositivo mecânico denominado POD (*Picosatellite Orbital Deployer*).

A fim de não danificar o veículo lançador e carga primária, é importante que o projeto da plataforma CubeSat obedeça a requisitos de interface e de segurança. Nesse sentido, é essencial seguir boas práticas de engenharia ao longo do desenvolvimento do nanossatélite, o que pode ser facilitado com a aplicação da engenharia de sistemas.

De acordo com definição do INCOSE (*International Council of Systems Engineering*), a engenharia de sistemas é uma abordagem interdisciplinar que viabiliza a realização de sistemas bem-sucedidos. Por meio de um processo sistemático e iterativo, consegue-se partir dos objetivos dos *stakeholders*, entes interessados no produto, e atingir uma estrutura final do sistema capaz de satisfazer os interesses dos envolvidos.

Por muito tempo, a engenharia de sistemas foi baseada em extensa documentação. De um tempo para cá, uma nova abordagem vem ganhando popularidade – a engenharia de sistemas baseada em modelos (MBSE – *Model Based System Engineering*).

A metodologia de engenharia de sistemas MBSE utiliza modelos, isto é, representações gráficas, para promover o intercâmbio de informações entre os participantes do projeto. O benefício da utilização de representações visuais consiste no potencial de prover uma comunicação sem ambiguidades.

Além da aplicação em novos projetos, a implementação dessa metodologia em sistemas já concluídos permite estruturar a documentação existente em um novo formato, baseado em modelos. Depois de ser feita uma primeira vez, a estruturação pode ser replicada em projetos similares.

Um projeto para o qual essa estruturação possa ser proveitosa trata-se justamente dos nanossatélites do tipo CubeSat, que já possuem especificações técnicas e conceitos de *design* normatizados. Particularmente, pode-se utilizar a documentação do nanossatélite

AESP14, primeiro CubeSat brasileiro, desenvolvido em parceria entre o ITA e o INPE, para a confecção dessa estrutura.

Para a aplicação da metodologia MBSE, deve-se adotar uma linguagem para confecção dos modelos. Uma possibilidade é a linguagem SysML (*System Modelling Language*), que se destaca pela sua especificidade para engenharia de sistemas.

O presente trabalho tem como objetivo a implementação da engenharia de sistema baseada em modelos para satélites do tipo CubeSats, particularmente para o nanosatélite AESP14, com o intuito de servir de referência para projetos similares. A modelagem foi feita utilizando-se os diagramas modelos da linguagem SysML, devido à especificidade já citada. Para estudo da linguagem, utilizou-se o livro-texto “*SysML for Systems Engineering*”, dos autores Jon Holt e Simon Perry.

## 2 OBJETIVOS DO TRABALHO

O objetivo geral do trabalho é implementar a engenharia de sistemas baseadas em modelos (MBSE) para nanossatélites do tipo Cubesat. Em particular, a implementação foi concretizada para o CubeSat AESP14, utilizando a linguagem SysML.

Para a progressão do trabalho, levanta-se os seguintes objetivos específicos, que ditam as etapas do projeto de iniciação científica:

- Estudar a linguagem SysML e seus diagramas modelo;
- Aprender a utilizar um *software* de modelagem para engenharia de sistemas, exercitando com modelamentos-base;
- Compreender a documentação do CubeSat AESP14;
- Implementar a engenharia de sistemas baseada em modelos para o AESP14;
- Analisar criticamente sobre a utilização da metodologia MBSE.



### 3 FUNDAMENTAÇÃO TEÓRICA

A linguagem SysML é uma linguagem de modelagem específica para engenharia de sistemas. Ela surgiu como extensão da linguagem UML (*Unified Modeling Language*), cuja aplicação inicial abrange projetos de *software*, especialmente. O estudo teórico acerca da linguagem SysML foi baseado na referência bibliográfica “*SysML for Systems Engineering*”, dos autores Jon Holt e Simon Perry.

Para compreender um sistema, é necessário modelá-lo em dois aspectos: estrutural, que compreende o que o sistema faz; e comportamental, que representa como o sistema faz, isto é, como desempenha as funções esperadas. Na linguagem SysML, o aspecto estrutural é modelado por *structural diagrams*, e o aspecto comportamental é modelado por *behavioural diagrams*.

A referência bibliográfica apresentou cinco diagramas estruturais, a citar *block definition diagram*, *internal block diagram*, *package diagram*, *parametric diagram* e *requirement diagram*, e quatro diagramas de comportamento, os quais foram *state machine diagram*, *sequence diagram*, *activity diagram* e *use case diagram*. A seguir, é feito um breve comentário sobre cada diagrama, contemplando principais elementos e finalidade.

#### 3.1 Diagramas estruturais

O *Block definition diagram* é considerado o esqueleto da linguagem SysML e pode ser utilizado para modelar praticamente qualquer sistema e interesse. Seus principais elementos são os blocos e as relações entre eles. Adicionalmente, é possível adicionar aos blocos: propriedades, que representam características; operações, que representam ações ou atividades desempenhadas; e restrições.

Além disso, esse diagrama também é capaz de representar interações entre os blocos, por meios de *flow ports* para matéria, informação e energia, ou *standard ports*, para serviços providos e requeridos. Já as relações também apresentam os mais variados tipos, sendo algumas das possibilidades: agregação e composição para representação um elemento que é feito de outros; dependência para elementos que, quando alterados,

afetam outros; e especialização para definição de novos elementos de um determinado tipo.

O *internal block diagram* possui aparência semelhante ao diagrama anterior, entretanto, por dar um enfoque maior à relação lógica entre os elementos do que a uma simples composição, é utilizado para modelar estruturas internas de um bloco e a implementação de elementos do sistema. O principal elemento desse diagrama é uma parte, que descreve um conjunto de blocos.

O *package diagram* é usado para identificar e relacionar pacotes, os quais são, por sua vez, um conjunto de diagramas. Em especial, a definição de acesso aos parâmetros de um outro pacote é feita por meio da relação de dependência, e pode ser de dois tipos: pública, quando pode ser compartilhada para outros pacotes de maneira indireta, ou privada, quando o compartilhamento é feito apenas de maneira direta, isto é, diretamente ao pacote e não a outros conectados a ele. Suas aplicações são destinadas a um modelamento em alto nível do sistema, ou para mostrar uma biblioteca de partes que é usada na produção de um produto.

O *parametric diagram* é essencial para a definição de restrições e regras. Pode ser considerado um tipo especial de *internal block diagram*, no qual os blocos são estereotipados do tipo restrição. Fazendo-se uma rede de restrições, é possível calcular *outputs* interessantes para o sistema por meio da definição/obtenção de alguns *inputs*.

O *requirement diagram* é utilizado para modelar requisitos do sistema e suas relações. Algumas relações utilizadas para relacionar requisitos entre si são: *nesting*, quando há uma decomposição do requisito em outros, *derive*, quando o requisito deriva outros, e *copy*, quando o requisito é copiado e utilizado em um outro contexto. Uma outra relação estabelece-se especificamente entre um caso de teste e um requisito, e é denominada *verify*. Relações genéricas entre requisitos e outros elementos são *satisfy*, para indicar um elemento do modelo que satisfaz o requisito, *refine*, para indicar um elemento que refina o requisito, e *trace*, para indicar uma relação entre requisito e elementos.

### 3.2 Diagramas comportamentais

O *state machine diagram* modela o comportamento de um bloco durante o seu tempo de vida. É normalmente denominado por *timing model*, uma vez que modela o tempo lógico: ordem de execução das atividades e ações, e condições de realização. Os elementos principais são estados e transições, sendo que esses últimos representam mudança entre estados. Se nessa transição há o envio de mensagem de um estado para o outro, este envio é modelado como evento. Cabe ressaltar a diferença entre atividades e ações: atividades tem tempo lógico não nulo, de forma que podem ser interrompidas, ao passo que ações tem tempo lógico nulo, isto é, são atômicas, não podendo ser interrompidas.

O *sequence diagram* relaciona-se diretamente a uma operação particular de um sistema. Escolhe-se um cenário, no qual os participantes envolvidos são representados por *Lifelines*, e as mensagens entre indivíduos é idealizada por meio de setas no diagrama. Essas mensagens são um tipo de ocorrência de evento.

O *activity diagram* pode ser usado para modelamento do sistema em baixo nível, retratando o comportamento em uma operação, ou em estado maior, modelando *workflows* e processos. Os fluxos de controle ou informação são iniciados, podem ser separados e reunidos, assim como encerrados. Há também como instituir pontos de decisão e partições para indicar o indivíduo/instituição destacado para cada atividade.

O *use case diagram* evidencia o aspecto funcional do sistema, ao invés de controle e tempos lógicos. Seus principais elementos são os *use cases*, atores, associações e fronteira do sistema. Podem ser usados para modelar contextos, em especiais físico e de negócio, e para modelar requisitos.

## 4 MATERIAIS E MÉTODOS UTILIZADOS

Para a modelagem do nanossatélite AESP14, foi utilizado o *software IBM Rational Rhapsody*, desenvolvido pela empresa IBM e disponível nos servidores do INPE. O modelamento foi baseado nas três estruturas-base enfatizadas no livro-texto, a citar requisito, sistema físico e processo.

A seguir, explica-se separadamente as estratégias de cada tipo de modelamento, e diagramas aplicados para o AESP14.

### 4.1 Modelamento de Requisito

A modelagem de requisitos parte, primeiramente, da identificação dos entes com interesse no projeto, denominados *stakeholders*, e de seus respectivos objetivos em relação ao produto. Esses objetivos são denominados requisitos, e ditam o restante do projeto.

De acordo com o livro-texto de referência, há três tipos de requisitos: relativos a negócios, funcionais e não-funcionais. Os requisitos de negócios são relacionados aos objetivos mercadológicos da organização, como “ganhar dinheiro” ou “fazer os clientes felizes”. Os requisitos funcionais referem-se ao que o sistema deve fazer, e deriva as funcionalidades do sistema. Já os requisitos não-funcionais são restrições de implementação, como normas a serem seguidas ou infraestruturas existentes a serem utilizadas.

Os requisitos podem ser organizados de acordo com o tipo, produzindo contextos do sistema, que são traçados em linguagem SysML com *use case diagrams*. Nesse diagrama, os requisitos são modelados por *use cases*, e são associados aos *stakeholders* de origem, que motivaram aquele objetivo, por meio de associações. Os *stakeholders* são modelados por elementos denominados *actors*, que representam, acima de tudo, um papel, e não um indivíduo.

Os contextos representam os requisitos apenas em um alto nível. Em seguida, é necessário fazer um processo de refinamento. Para isso, utilizam-se as relações disponíveis na linguagem: a associação do tipo “*include*” permite quebrar uma funcionalidade em outras de mais baixo nível; a associação do tipo “*extend*” amplia uma

funcionalidade, adicionando um caso particular dependendo do que acontece com o sistema quando ele está em operação. Adicionalmente, podem ser utilizados estereótipos do tipo “*constraint*” para derivar requisitos não-funcionais que delimitam funcionalidades. O uso dessas relações permite manter a rastreabilidade entre requisitos, e avançar do mais alto nível para o mais baixo nível.

A documentação do nanossatélite AESP14 engloba numerosos requisitos, em diversos níveis. Assim, optou-se por fazer o modelamento apenas dos requisitos dos *stakeholders*, com o uso do *use case diagram*, e a primeira extensão para requisitos de projeto e missão, com o uso do *requirement diagram*. A extensão para o restante do sistema segue método análogo, baseado no levantamento de requisitos e *stakeholders*, associação entre eles por meio das estruturas de *use case* e *actors*, e refinamentos dos requisitos utilizando as relações apropriadas.

#### **4.2 Modelamento de Sistema Físico**

A modelagem de um sistema físico parte da identificação dos componentes do sistema e da interface entre eles. Para o nanossatélite AESP14, essas informações já estão disponíveis na documentação disponibilizada, e, na época, foram levantadas por meio da observação de outras plataformas CubeSat.

Em relação às interfaces, é importante ressaltar a diferença entre a estrutura *flow port*, que é usada para passar ou receber material, energia ou dados, e a *standard port*, que é usada quando uma parte do sistema invoca operações em outra parte.

O modelamento físico corresponde ao aspecto estrutural do sistema, de forma que são usados diagramas estruturais. Optou-se por utilizar o modelo *block definition diagram* ao invés do *internal block diagram*, uma vez que se deu mais ênfase aos componentes do sistema e suas interfaces, do que às relações lógicas entre eles.

#### **4.3 Modelamento de Processo**

O modelamento de processos tem vasta aplicação, e objetiva confeccionar orientações sobre as metodologias a serem seguidas operacionalmente. A vantagem da utilização da engenharia de sistema baseada em modelos consiste na visualização rápida da ordem

das atividades, condições entre elas, e dos seus responsáveis, ao longo de todo o ciclo de vida.

Na referência utilizada, um exemplo de modelamento de processo é feito com a utilização de sete diagramas diferentes da linguagem SysML, para mostrar o procedimento por inteiro, desde o levantamento das atividades, até a organização da ordem e dos responsáveis.

A modelagem foi aplicada para o processo de operação do AESP14. Como todos os processos já estavam documentados por escrito, optou-se por traçar apenas dois tipos de diagramas: utilizou-se dois diagramas do tipo *sequence diagram*, um para modelar o ciclo de vida do produto, e outro para modelar as mensagens entre os elementos do sistema durante uma operação normal; ademais, utilizou-se um *activity diagram* para modelar os estados e modos do satélite, e como são feitas as transições entre eles. Nota-se que o modelamento de processo se ocupou da parte comportamental do nanossatélite, completando a outra faceta do sistema.

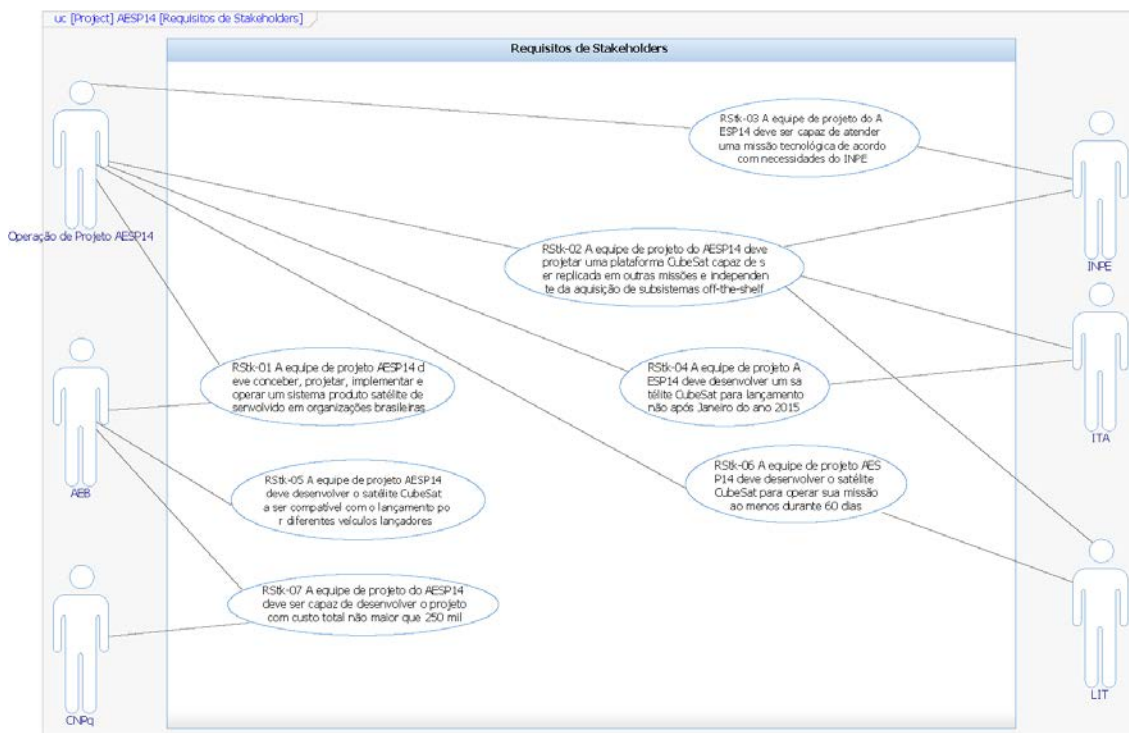
## 5 ANÁLISES E RESULTADOS

Nesta seção, são apresentados os diagramas desenvolvidos para o CubeSat AESP14, conforme metodologia apresentada na seção anterior. Em sequência, apresenta-se o modelamento de requisitos, de sistema físico e de processo.

### 5.1 Diagramas para modelamento de requisito

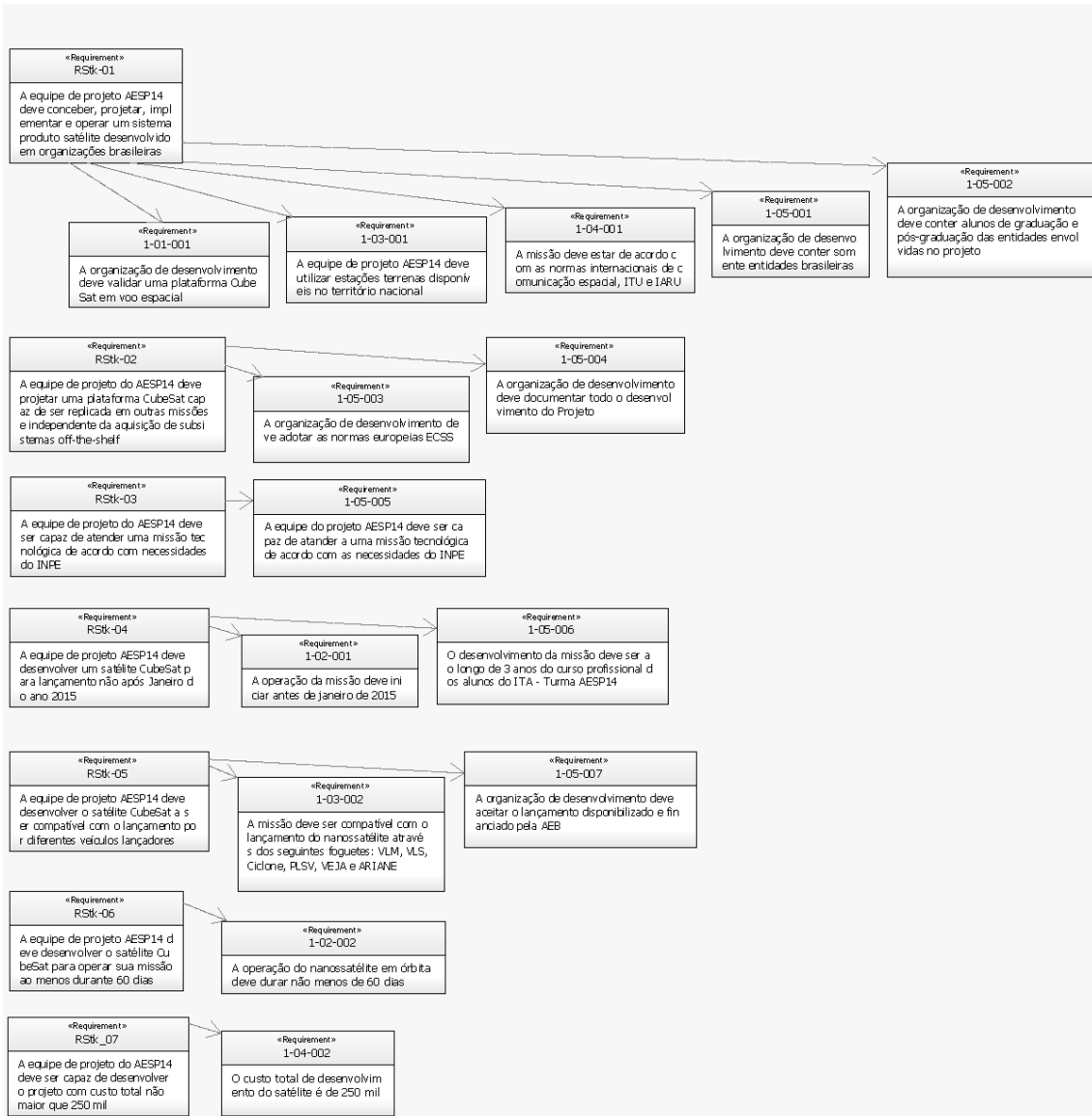
Para o modelamento de requisitos do AESP14, utilizou-se a estrutura *use case diagram*. Devido a vastidão de requisitos, o modelamento foi aplicado apenas para os de mais alto nível, denominados requisitos de *stakeholders*.

Figura 5.1 – Requisitos de *stakeholders* do AESP14 modelado por *use case diagram*



Adicionalmente, utilizou-se a estrutura *requirements diagram* para mostrar como pode ser feito o rastreamento entre requisitos, e como é feita a ampliação a partir dos requisitos de mais alto nível.

Figura 5.2 – Rastreabilidade e ampliação dos requisitos por *requirements diagram*



## 5.2 Diagrama para modelamento de sistema físico

Para o modelamento do sistema físico do AESP14, utilizou-se a estrutura *block definition diagram*. Conforme documentação, o sistema AESP14 é dividido em quatro segmentos: lançador, que contempla veículo lançador, base de lançamento e adaptador com veículo lançador; espacial, que contempla o CubeSat em si e GSEs (*Ground Support Equipment*) para o satélite; solo, que engloba centro de controle e rastreamento e centro de missão; e usuário, que recebe os dados finais compilados.

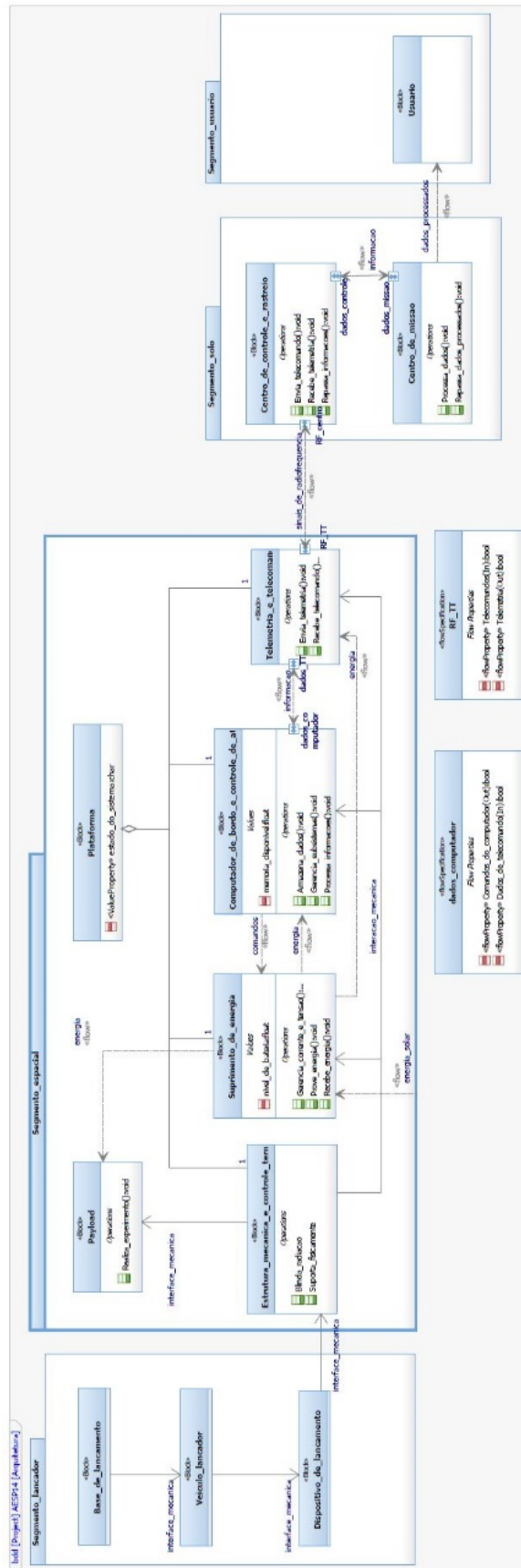


Particularmente, o CubeSat é composto pelo *payload*, a carga útil onde vão os experimentos, e a plataforma, principal objeto de desenvolvimento do grupo. Esta por sua vez é dividida nos subsistemas: estrutura mecânica e controle térmico; suprimento de energia; computador de bordo e controle de atitude; telemetrias e telecomandos.

A construção do diagrama estrutural permite visualizar com mais clareza as interações entre sistemas e subsistemas. Destaca-se a interface mecânica entre o segmento lançador e o espacial, feita pelo dispositivo POD, as interfaces de dados entre segmentos espacial, solo e usuário, e as interfaces mecânicas, de dados e de energia internamente à plataforma.

Uma forma de incrementar o modelamento de um sistema físico consistem em utilizar os conceitos de *constraint* e *parametric diagram*. Implementando expressões matemáticas e/ou lógicas, é possível, a partir da definição de alguns valores, computar as variáveis dependentes e se estas satisfarão os requisitos. Esse diagrama pode ser aplicado no procedimento de determinação da órbita do sistema, para verificar se a altura é compatível com o tempo mínimo de operação da missão, com o tempo máximo de descarte e com os veículos lançadores disponíveis no mercado.

Figura 5.3 – Modelo físico do AESP14 por *block definition diagram*



### 5.3 Diagramas para modelamento de processo

Para o modelamento de processo aplicado ao AESP14, foram utilizados os diagramas *sequence diagram* e *activity diagram*. O processo modelado foi a própria operação do satélite AESP14.

A operação esperada do AESP14 após lançamento consiste na ejeção da primária do foguete e inicialização do sistema; funcionamento preliminar com envio de telemetrias de teste; funcionamento nominal; validação da plataforma e desligamento do satélite. Cada estágio define um estado do sistema. Outros estados são definidos quando o funcionamento não segue conforme o esperado.

Particularmente, para o AESP14, são definidos os seguintes estados: repouso, quando desligado esperando ejeção; em condicionamento, quando está ligando os subsistemas, desdobrando antena e enviando uma mensagem identificadora para a central de terra; operacional, quando desempenha funções, envia telemetrias e recebe telecomandos; perdido, quando em órbita mas sem comunicação com estações terrenas; e desligado, quando a plataforma é validade e o satélite é desligado, aguardando reentrada.

Adicionalmente, o CubeSat pode apresentar três modos quando operacional: nominal, quando completamente ligado; seguro, quando a bateria está com pouca energia ou problema de recarga e apenas funcionalidades críticas são mantidas; ou degradado, quando algum sistema não está funcionando corretamente, porém a telemetria está.

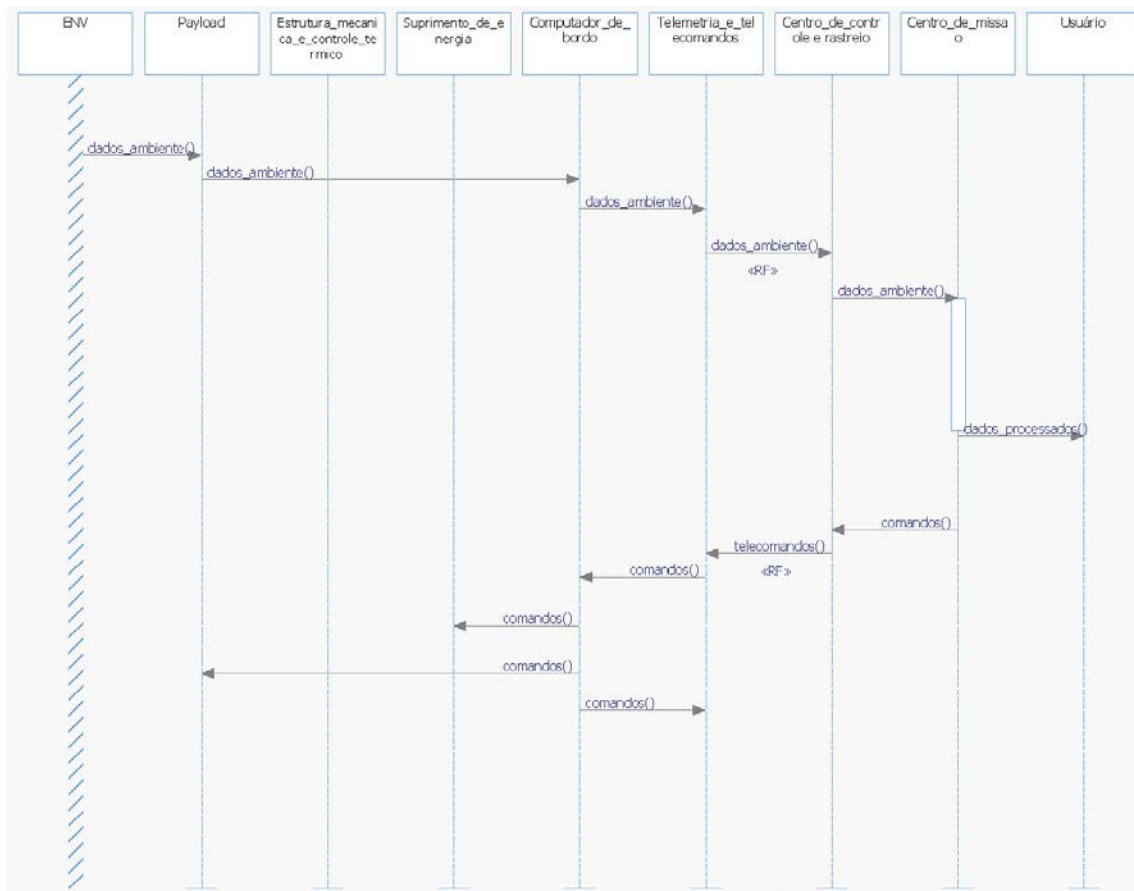
Primeiramente, o *sequence diagram* foi utilizado para traçar a linha do tempo básica de operação do produto, quando este funciona corretamente, da ejeção pelo veículo lançador ao desligamento por telecomando da estação de terra.

Figura 5.4 – Linha do tempo após lançamento modelado por *sequence diagram*



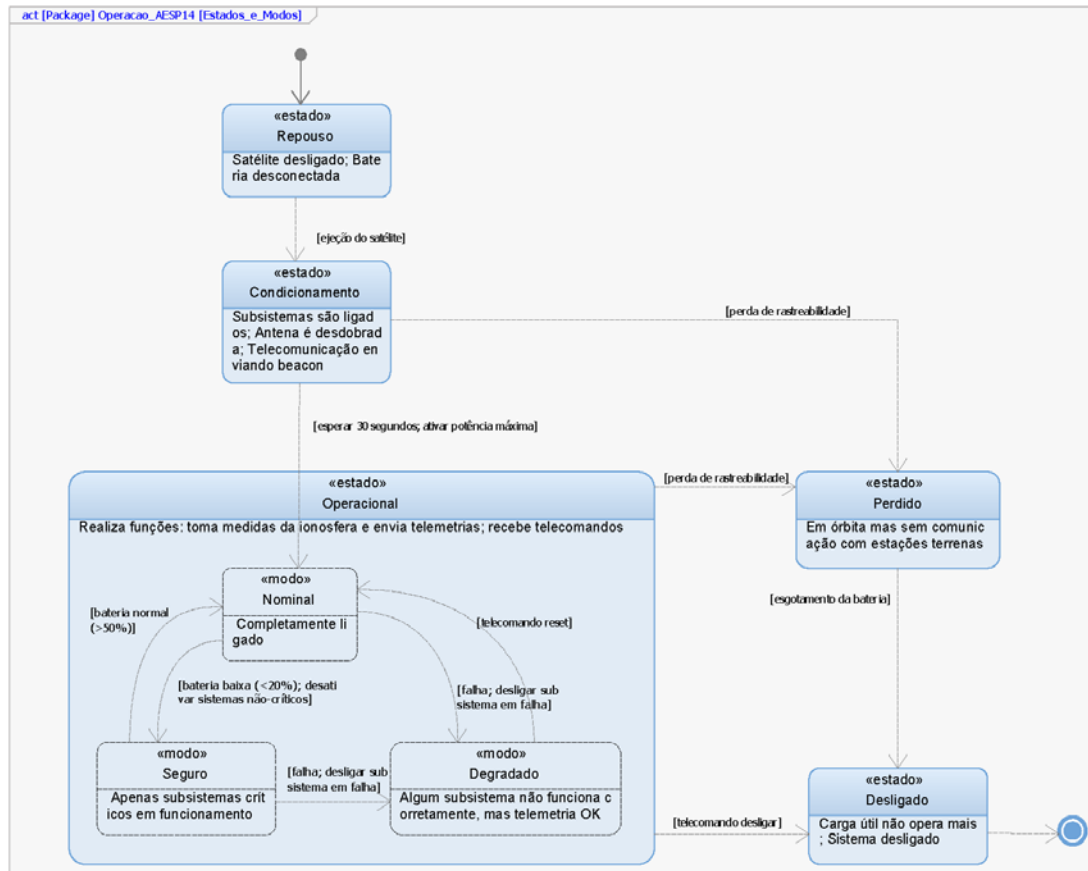
Em seguida, o *sequence diagram* é utilizado de outra forma, para evidenciar mensagens durante o cenário operacional entre os entes do sistema. Cada elemento do sistema é representado por uma *lifeline*, e as setas representam mensagens entre eles.

Figura 5.5 – Mensagens entre elementos do sistema durante operação nominal modelado por *sequence diagram*



Para completar o modelo comportamental do AESP14, o *activity diagram* foi utilizado para modelar como o sistema reage a acontecimentos internos, as condições de mudança entre estados e modos e as funcionalidades desempenhadas em cada estado ou modo. Cada retângulo representa um estado, e as setas entre eles representam transições.

Figura 5.6 – Transições entre estados e modos modelado por *activity diagram*



Ressalta-se que esse modelamento pode ser aplicado também para todo o processo de engenharia de sistemas desempenhado para a elaboração da plataforma CubeSat AESP14, incorporando como entrada e saída das atividades os documentos respectivamente necessários e obtidos em cada fase da engenharia de sistemas.

#### 5.4 Discussão sobre utilização de MBSE para engenharia de sistema de CubeSat

Analisando-se criticamente, a utilização da engenharia de sistemas baseada em modelos é realmente benéfica para o entendimento do sistema. As interações e os processos ficam visualmente mais intuitivos do que descritos textualmente. Percebe-se que a documentação já continha alguns fluxogramas, porém a utilização de uma linguagem permite a padronização desses desenhos entre desenvolvedores, como já acontece com desenhos técnicos, por exemplo.

A dificuldade de utilização da ferramenta consiste principalmente na inércia para adoção de metodologia: é necessário aprender uma nova linguagem, um novo programa e efetuar alguns exemplos para possuir segurança e facilidade com a implementação da engenharia de sistema baseada em modelos.

O modelamento executado para o nanossatélite AESP14 pode ser aprofundado em relação ao que foi desenvolvido. Os diagramas referentes a requisitos e estrutura física podem ser ampliados para níveis mais baixos. Outros diagramas podem ser desenvolvidos, como um *parametric diagram* para determinação da órbita, e modelamento de processo para o processo de engenharia de sistemas, conforme mencionado. A fim de criar uma estrutura de fato facilmente replicável para outros projetos CubeSat, uma documentação mais extensa deve ser gerada.

## 6 CONCLUSÃO

O presente projeto de iniciação científica foi satisfatório em cumprir os seus objetivos iniciais. Especificamente, os estudos acerca da linguagem e operação do *software* foram desempenhados, assim como compressão do funcionamento do CubeSat AESP14. Os diagramas construídos permitiram atingir o objetivo primordial de implementação da engenharia de sistemas baseada em modelos para CubeSats.

Os resultados obtidos, porém, não representam um modelamento em sua completude do sistema AESP14. Por conseguinte, a rastreabilidade entre os diagramas não foi desenvolvida em sua potencialidade, a qual é uma das grandes vantagens da linguagem SysML.

O trabalho constatou os benefícios na utilização de MBSE para engenharia de sistemas, e, em particular, suas potencialidades para plataformas CubeSats, que já possuem engenharia bem estruturada. Futuros desenvolvimentos podem ampliar os modelamentos e criar, de fato, uma metodologia detalhada para aplicação de MBSE para CubeSats e, até mesmo, outras tecnologias, facilitando sua utilização e aumentando ainda mais a popularidade.



## REFERÊNCIAS BIBLIOGRÁFICAS

CALPOLY. **CubeSat Design Specification: (CDS)**. Rev 13. ed. San Luis Obispo - CA: [s.n.], 2014. Disponível em: <[www.cubesat.org/resources](http://www.cubesat.org/resources)>. Acesso em: 20 jul. 2018.

HOLT, J.; PERRY, S. **SysML for Systems Engineering**. 1.ed. London, UK: The Institution of Engineering and Technology, 2008. 352p. ISBN 978-0-86341-825-9.

IBM. **System Engineering Tutorial for Rational Rhapsody 7.4**. Disponível em: [ftp://ftp.software.ibm.com/software/rationalsdp/documentation/product\\_doc/Rhapsody/version\\_7-5/tutorial\\_Systems\\_Eng.pdf](ftp://ftp.software.ibm.com/software/rationalsdp/documentation/product_doc/Rhapsody/version_7-5/tutorial_Systems_Eng.pdf). Acesso em: 10 out. 2017.

IBM. **IBM Rational Rhapsody 7.6.1 releases documentation**. Disponível em: [https://www.ibm.com/support/knowledgecenter/SSB2MU\\_7.6.1/com.ibm.rhp.homepage.doc/helpindex\\_rhapsody.html](https://www.ibm.com/support/knowledgecenter/SSB2MU_7.6.1/com.ibm.rhp.homepage.doc/helpindex_rhapsody.html). Acesso em: 10 out. 2017.

IBM. **Introduction: Basic systems engineering design in Rational Rhapsody**. Disponível em: [https://www.ibm.com/support/knowledgecenter/SSB2MU\\_8.1.2/com.ibm.rhp.systemseng.tutorial.doc/topics/rhp\\_tu\\_intro\\_systems\\_eng1.html](https://www.ibm.com/support/knowledgecenter/SSB2MU_8.1.2/com.ibm.rhp.systemseng.tutorial.doc/topics/rhp_tu_intro_systems_eng1.html)>. Acesso em: 10 out. 2017.

INCOSE. **Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities**. Ver.3. San Diego: International Council on Systems Engineering (INCOSE), 2006. 185 p. INCOSE-TP-2003-002-03.

MENDES, W. S. **Um método de modelagem descritiva de sistemas de engenharia para possibilitar a geração automática de requisitos textuais aplicado a um satélite**. 2018. 299 p. IBI: <8JMKD3MGP3W34P/3QBJFHL>. (sid.inpe.br/mtc-m21b/2018/01.08.19.24-TDI). Tese (Doutorado em Engenharia e Gerenciamento de Sistemas Espaciais) - Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos, 2018. Disponível em: <http://urlib.net/8JMKD3MGP3W34P/3QBJFHL>. Acesso em: 20 jul. 2018.

