



MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÃO
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

sid.inpe.br/mtc m21d/2023/12.12.02.02-TDI

MÉTODO PARA DESENVOLVIMENTO DE SISTEMAS DE INFORMAÇÃO PARA APOIAR PROCESSOS DO CICLO DE VIDA DE SATÉLITES

Ana Claudia de Paula Silva

Tese de Doutorado do Curso de Pós-Graduação em Engenharia e Tecnologia Espaciais/Engenharia e Gerenciamento de Sistemas Espaciais, orientada pelo Dr. Geilson Loureiro, aprovada em 07 de dezembro de 2023.

URL do documento original:

<<http://urlib.net/8JMKD3MGP3W34T/4ACABS5>>

INPE
São José dos Campos
2023

PUBLICADO POR:

Instituto Nacional de Pesquisas Espaciais - INPE
Coordenação de Ensino, Pesquisa e Extensão (COEPE)
Divisão de Biblioteca (DIBIB)
CEP 12.227-010
São José dos Campos - SP - Brasil
Tel.:(012) 3208-6923/7348
E-mail: pubtc@inpe.br

CONSELHO DE EDITORAÇÃO E PRESERVAÇÃO DA PRODUÇÃO INTELLECTUAL DO INPE - CEPPII (PORTARIA Nº 176/2018/SEI-INPE):

Presidente:

Dra. Marley Cavalcante de Lima Moscati - Coordenação-Geral de Ciências da Terra (CGCT)

Membros:

Dra. Ieda Del Arco Sanches - Conselho de Pós-Graduação (CPG)
Dr. Evandro Marconi Rocco - Coordenação-Geral de Engenharia, Tecnologia e Ciência Espaciais (CGCE)
Dr. Rafael Duarte Coelho dos Santos - Coordenação-Geral de Infraestrutura e Pesquisas Aplicadas (CGIP)
Simone Angélica Del Ducca Barbedo - Divisão de Biblioteca (DIBIB)

BIBLIOTECA DIGITAL:

Dr. Gerald Jean Francis Banon
Clayton Martins Pereira - Divisão de Biblioteca (DIBIB)

REVISÃO E NORMALIZAÇÃO DOCUMENTÁRIA:

Simone Angélica Del Ducca Barbedo - Divisão de Biblioteca (DIBIB)
André Luis Dias Fernandes - Divisão de Biblioteca (DIBIB)

EDITORAÇÃO ELETRÔNICA:

Ivone Martins - Divisão de Biblioteca (DIBIB)
André Luis Dias Fernandes - Divisão de Biblioteca (DIBIB)



MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÃO
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

sid.inpe.br/mtc m21d/2023/12.12.02.02-TDI

MÉTODO PARA DESENVOLVIMENTO DE SISTEMAS DE INFORMAÇÃO PARA APOIAR PROCESSOS DO CICLO DE VIDA DE SATÉLITES

Ana Claudia de Paula Silva

Tese de Doutorado do Curso de Pós-Graduação em Engenharia e Tecnologia Espaciais/Engenharia e Gerenciamento de Sistemas Espaciais, orientada pelo Dr. Geilson Loureiro, aprovada em 07 de dezembro de 2023.

URL do documento original:

<<http://urlib.net/8JMKD3MGP3W34T/4ACABS5>>

INPE
São José dos Campos
2023

Dados Internacionais de Catalogação na Publicação (CIP)

Silva, Ana Claudia de Paula.

Si38M Método para desenvolvimento de sistemas de informação para apoiar processos do ciclo de vida de satélites / Ana Claudia de Paula Silva. – São José dos Campos : INPE, 2023.

xxii + 195 p. ; (sid.inpe.br/mtc m21d/2023/12.12.02.02-TDI)

Tese (Doutorado em Engenharia e Tecnologia Espaciais/Engenharia e Gerenciamento de Sistemas Espaciais) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2023.

Orientador : Dr. Geilson Loureiro.

1. Engenharia de sistemas espaciais. 2. Sistemas de informação. 3. Desenvolvimento de satélites. 4. MBSE. I.Título.

CDU 629.78:004.775



Esta obra foi licenciada sob uma Licença [Creative Commons Atribuição-NãoComercial 3.0 Não Adaptada](https://creativecommons.org/licenses/by-nc/3.0/).

This work is licensed under a [Creative Commons Attribution-NonCommercial 3.0 Unported License](https://creativecommons.org/licenses/by-nc/3.0/).



MINISTÉRIO DA
CIÊNCIA, TECNOLOGIA
E INOVAÇÃO



INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

DEFESA FINAL DE TESE ANA CLAUDIA DE PAULA SILVA BANCA Nº 290/2023, REGISTRO 85286/2018

No dia 07 de dezembro de 2023, às 09h00, o(a) aluno(a) mencionado(a) acima defendeu seu trabalho final (apresentação oral seguida de arguição) perante uma Banca Examinadora, cujos membros estão listados abaixo. O(A) aluno(a) foi APROVADO(A) pela Banca Examinadora, por unanimidade, em cumprimento ao requisito exigido para obtenção do Título de Doutora em Engenharia e Tecnologias Espaciais / Engenharia e Gerenciamento de Sistemas Espaciais. O trabalho não precisa de correções.

Título: “Método para desenvolvimento de sistemas de informação para apoiar processos do ciclo de vida de satélites”.

Membros da Banca:

Maurício Gonçalves Vieira Ferreira – Presidente – INPE
Geilson Loureiro – Orientador – INPE
Walter Abrahao dos Santos – Membro Interno – INPE
Eduardo Escobar Bürger – Membro Externo – UFSM
Lisandra Manzoni Fontoura – Membro Externo – UFSM



Documento assinado eletronicamente por **Geilson Loureiro, Coordenador-Geral de Infraestrutura e Pesquisas Aplicadas**, em 11/12/2023, às 14:00 (horário oficial de Brasília), com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Mauricio Goncalves Vieira Ferreira, Coordenador de Rastreo, Controle e Recepção de Satélites**, em 11/12/2023, às 15:37 (horário oficial de Brasília), com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **LISANDRA MANZONI FONTOURA (E), Usuário Externo**, em 12/12/2023, às 09:36 (horário oficial de Brasília), com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Walter Abrahão dos Santos, Tecnologista**, em 12/12/2023, às 09:50 (horário oficial de Brasília), com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Eduardo Escobar Burger (E), Usuário Externo**, em 12/12/2023, às 10:06 (horário oficial de Brasília), com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



A autenticidade deste documento pode ser conferida no site <https://sei.mcti.gov.br/verifica.html>, informando o código verificador **11561159** e o código CRC **941902C1**.

Referência: Processo nº 01340.010057/2023-23

SEI nº 11561159

“Estudante: aplica-te aos teus livros com espírito de apóstolo, intimamente convencido de que essas horas e horas são já - agora! - um sacrifício espiritual oferecido a Deus, proveitoso para a humanidade, para o teu país, para a tua alma”.

SÃO JOSEMARÍA ESCRIVÁ
“Sulco”

em

*A meus amados pais, **Vanderlei José da Silva** e **Maria Elisa de Paula Silva**, que com admirável sabedoria escolheram o estudo como a herança que deixariam aos filhos.*

*A meus amados filhos, **Rafael de Paula Silva** e **Marcela de Paula Silva**, que ressignificaram minhas conquistas e são a razão das minhas escolhas.*

AGRADECIMENTOS

Aos professores do INPE, com os quais tive a honra de cursar as disciplinas, por dividir comigo sua vasta experiência na área de engenharia de sistemas espaciais. Especialmente ao meu orientador, Dr. Geilson Loureiro, pela confiança no meu trabalho.

Aos senhores membros da banca pela disposição em avaliar e contribuir com meu trabalho.

Aos colegas de disciplinas, pelo companheirismo e amizade que tornaram o caminho mais leve.

Ao meu esposo Luiz Alexandre, pelo apoio pessoal necessário para que eu pudesse dedicar as muitas horas de estudo para a realização deste trabalho.

Aos meus filhos, Rafael e Marcela, pela, talvez futura, compreensão das minhas horas de ausência.

Aos meus pais que instigaram em mim e em meus irmãos, desde muito cedo, o interesse pelos estudos.

E ao meu bom e amado Deus, que a todos esses colocou no meu caminho, interviu nos meus impossíveis, me capacitou e fortaleceu.

RESUMO

Muitas informações levantadas pelo engenheiro de sistemas, no domínio da Engenharia de Sistemas, para desenvolver um produto espacial também são necessárias e levantadas pelo engenheiro de software, no domínio da Engenharia de Software, para desenvolver os sistemas de informação que darão suporte aos processos do ciclo de vida desse produto. Este trabalho propõe um método para o desenvolvimento simultâneo e colaborativo de satélites e dos sistemas de informação unindo esses dois domínios, de forma que o esforço realizado no domínio da Engenharia de Sistemas seja aproveitado também no domínio da Engenharia de Software. Assumindo o uso da *Model-Based Systems Engineering* (MBSE) e da *Systems Concurrent Engineering* (SCE) como abordagens para a engenharia de um satélite, foram analisadas as atividades realizadas pelo engenheiro de sistemas para identificar como o trabalho do engenheiro de software pode ser inserido no processo de engenharia de sistemas. Um método foi proposto e exemplificado. O satélite utilizado como exemplo foi o AMAZONIA 1 e o processo do ciclo de vida a ser apoiado pelo sistema de informação foi o processo de montagem, integração e testes desse satélite. Foi concluído que o trabalho simultâneo e colaborativo entre os domínios da Engenharia de Sistemas e da Engenharia de Software é viável e vantajoso para ambas as partes.

Palavras-chave: Engenharia de sistemas espaciais. Sistemas de informação. Desenvolvimento de satélites. MBSE.

METHOD FOR DEVELOPING OF INFORMATION SYSTEMS TO SUPPORT SATELLITE LIFE CYCLE PROCESSES

ABSTRACT

A lot of information gathered by the systems engineer, in the Systems Engineering domain, to develop a space product is also necessary and gathered by the software engineer, in the Software Engineering domain, to develop the information systems that will support the life cycle processes of that product. This work proposes a method for the simultaneous and collaborative development of satellites and information systems, linking these two domains, so that the effort made in the Systems Engineering domain can also be used in the Software Engineering domain. Assuming the use of *Model-Based Systems Engineering* (MBSE) and *Systems Concurrent Engineering* (SCE) as approaches to engineering a satellite, the activities carried out by the systems engineer were analyzed to identify how the software engineer's work can be inserted into the systems engineering process. A method was proposed and exemplified. The satellite used as an example was AMAZONIA 1 and the life cycle process to be supported by the information system was the process of assembly, integration and testing of this satellite. It was concluded that simultaneous and collaborative work between the Systems Engineering and Software Engineering domains is feasible and can benefit both sides.

Keywords: Space systems engineering. Information systems. Satellite development. MBSE.

LISTA DE FIGURAS

	<u>Pág.</u>
1.1 Processo típico de AIT.	2
1.2 Papel de um sistema de informação durante testes de integração elétrica em um satélite.	5
2.1 <i>Total View Framework</i>	18
2.2 Método SCE.	19
3.1 Quantidade de trabalhos sobre o tema “MBSE na área espacial” - bases Scopus e Web of Science em 17/09/2023.	24
3.2 Estado da arte do tema “MBSE na área espacial”.	25
3.3 Iniciativas de adesão à MBSE na NASA - linha do tempo.	26
3.4 Resumo do planejamento de cinco anos (2016 a 2020) da NASA para implantação da MBSE.	28
3.5 Quantidade de trabalhos sobre o tema “sistemas de informação no contexto do ciclo de vida de produtos complexos” - bases Scopus e Web of Science em 17/09/2023.	35
4.1 Processos do ciclo de vida do satélite AMAZONIA 1.	39
4.2 <i>Stakeholders</i> do produto e seus <i>concerns</i>	41
4.3 <i>Stakeholders</i> da organização e seus <i>concerns</i>	42
4.4 Desdobramento de <i>concerns</i> em requisitos de <i>stakeholder</i> (<i>stakeholders</i> do produto).	43
4.5 Desdobramento de <i>concerns</i> em requisitos de <i>stakeholder</i> (<i>stakeholders</i> da organização).	43
4.6 Estrutura funcional da organização que implementa o processo de AIT do AMAZONIA 1.	47
4.7 Parte do modelo de comportamento funcional da organização que implementa o processo de AIT do AMAZONIA 1.	48
4.8 Diagrama de Blocos SysML simplificado de parte do modelo de voo do satélite AMAZONIA 1.	50
4.9 Diagrama de Blocos Internos SysML simplificado representando o subsistema AOCS do modelo de voo do satélite AMAZONIA 1.	51
4.10 Diagrama de Blocos SysML simplificado representando parte da arquitetura de implementação do LIT (elementos físicos).	52
4.11 Diagrama de Blocos SysML simplificado representando parte da arquitetura de implementação do LIT (elementos organizacionais - equipes).	53

5.1	Proposta de especialização da abordagem SCE.	58
5.2	Método proposto.	61
6.1	Diagrama de Casos de Uso emergente da aplicação do método proposto.	75
6.2	Diagrama de Classes emergente da aplicação do método proposto - 1.	76
6.3	Diagrama de Classes emergente da aplicação do método proposto - 2.	76
6.4	Diagrama de Classes emergente da aplicação do método proposto - 3.	77
6.5	IBD com Partes classificadas com estereótipos.	79
6.6	Código groovy para ler os modelos do Rhapsody.	81
6.7	Código groovy para criar objetos para o SI a partir dos elementos dos modelos do Rhapsody.	82
6.8	Compilação das repostas da questão 1.	86
6.9	Compilação das repostas da questão 2.	86
6.10	Compilação das repostas da questão 3.	87
6.11	Compilação das repostas da questão 4.	87

LISTA DE TABELAS

	<u>Pág.</u>
1.1 Principais objetivos dos principais documentos de AIT.	3
1.2 Classificação da pesquisa.	8
3.1 Benefícios da MBSE demonstrados pelas equipes do Pathfinder 1.	27
3.2 Tabela comparativa de trabalhos correlatos.	36
5.1 MBSCE X desenvolvimento de sistemas de informação.	59
5.2 Exemplo de mapeamento entre o domínio da Engenharia de Sistemas e o domínio da Engenharia de Software.	60
6.1 Influências dos requisitos dos <i>stakeholders</i> no SI - 1.	70
6.2 Influências dos requisitos dos <i>stakeholders</i> no SI - 2.	70
6.3 Influências dos requisitos dos <i>stakeholders</i> no SI - 3.	71
6.4 Influências dos requisitos dos <i>stakeholders</i> no SI - 4.	71
6.5 Influências dos modelos de estrutura funcional da organização no SI - 1.	72
6.6 Influências dos modelos de estrutura funcional da organização no SI - 2.	72
6.7 Influências dos modelos de estrutura funcional da organização no SI - 3.	72
6.8 Influências dos modelos de comportamento funcional da organização no SI - 1.	73
6.9 Influências dos modelos de comportamento funcional da organização no SI - 2.	73
6.10 Influências dos modelos de comportamento funcional da organização no SI - 3.	74
6.11 Influências dos modelos de comportamento funcional da organização no SI - 4.	74
6.12 Mapeamento entre os dois domínios.	80

LISTA DE ABREVIATURAS E SIGLAS

AIT	–	<i>Assembly, Integration and Testing</i> (Montagem, Integração e Testes)
BDD	–	<i>Block Definition Diagram</i> (Diagrama de Definição de Blocos)
ESA	–	<i>European Space Agency</i> (Agência Espacial Européia)
GSE	–	<i>Ground support equipment</i> (Equipamento de apoio em solo)
IBD	–	<i>Internal Block Diagram</i> (Diagrama de Blocos Internos)
INPE	–	Instituto Nacional de Pesquisas Espaciais
LIT	–	Laboratório de Integração e Testes
MBSE	–	Model-Based Systems Engineering (Engenharia de Sistemas baseada em modelos)
NASA	–	<i>National Aeronautics and Space Administration</i> (Administração Nacional da Aeronáutica e Espaço)
NCR	–	<i>Non-Conformance Report</i> (Relatório de Não Conformidade)
SCE	–	Systems Concurrent Engineering (Engenharia Simultânea de Sistemas)
SI	–	Sistema de informação
SysML	–	Systems Modeling Language (Linguagem de Modelagem de Sistemas)
UML	–	Unified Modelling Language (Linguagem de modelagem unificada)

SUMÁRIO

	<u>Pág.</u>
1 INTRODUÇÃO	1
1.1 Relevância de sistemas de informação para apoiar o processo de AIT de satélites	2
1.2 Motivação	6
1.3 Objetivo	7
1.3.1 Objetivo geral	7
1.3.2 Objetivos específicos	7
1.4 Metodologia e ferramentas	8
1.4.1 Classificação da pesquisa	8
1.4.2 Condução da pesquisa e ferramentas utilizadas	9
1.5 Organização da tese	11
2 FUNDAMENTAÇÃO TEÓRICA	13
2.1 Engenharia de software	13
2.2 UML	14
2.3 Processos de negócio e sistemas de informação	15
2.4 Ciclo de vida satélites	16
2.5 SCE	17
2.5.1 Atividades propostas pela SCE	17
2.6 MBSE	19
2.7 SysML	21
3 REVISÃO DE LITERATURA	23
3.1 MBSE na área espacial	23
3.1.1 Palavras-chave e critérios de exclusão	23
3.1.2 Estado da arte do tema “MBSE na área espacial”	24
3.1.2.1 Iniciativas de adesão à MBSE na NASA	25
3.1.2.2 Iniciativas de adesão à MBSE na ESA	28
3.1.2.3 Iniciativas de adesão à MBSE na Thales Alenia Space	29
3.1.2.4 Iniciativas de adesão à MBSE na Airbus Defence and Space	30
3.1.2.5 Iniciativas de adesão à MBSE no INPE	31
3.2 Sistemas de informação no contexto do ciclo de vida de produtos complexos	32
3.2.1 Palavras-chave e critérios de exclusão	32

3.2.2	Estado da arte do tema “sistemas de informação no contexto do ciclo de vida de produtos complexos”	34
3.3	Trabalhos correlatos	35
4	DEMONSTRAÇÃO DA VIABILIDADE DE INTEGRAR O DESENVOLVIMENTO DE SISTEMAS DE INFORMAÇÃO NO PROCESSO DA SCE	37
4.1	Definição da missão e processos do ciclo de vida do satélite	38
4.2	Análises de <i>stakeholders</i> e de requisitos e suas influências nos sistemas de informação	40
4.3	Análise funcional e sua influência nos sistemas de informação	45
4.3.1	Análise funcional da organização	46
4.3.1.1	Estrutura funcional da organização que implementa o processo de AIT	46
4.3.1.2	Comportamento funcional da organização que implementa o processo de AIT	47
4.3.2	Análise funcional do produto	49
4.4	Arquitetura de implementação e seu relacionamento com os sistemas de informação	49
4.4.1	Arquitetura de implementação do produto	50
4.4.2	Arquitetura de implementação da organização	52
5	PROPOSTA DE UM MÉTODO PARA INTEGRAR O DESENVOLVIMENTO DE SISTEMAS DE INFORMAÇÃO COM O DESENVOLVIMENTO DE SISTEMAS ESPACIAIS	55
5.1	Uso da MBSE na prática da SCE - MBSCE	55
5.2	Especialização da SCE para contemplar o desenvolvimento dos sistemas de informação	56
5.3	Convenção de modelagem e integração dos modelos com o sistema de informação	60
5.4	Método proposto	61
5.4.1	Descrição do método	61
5.4.2	Requisitos para utilização do método	63
5.4.3	Aplicabilidade do método	65
6	EXEMPLO DE APLICAÇÃO E AVALIAÇÃO DO MÉTODO	69
6.1	Exemplo de aplicação do método	69
6.1.1	Definições	69
6.1.2	Análises MBSCE	69

6.1.3	Especificação do sistema de informação a partir dos artefatos da MBSCE	69
6.1.4	Mapeamento e integração dos modelos dos dois domínios	77
6.2	Avaliação do método	82
6.2.1	Elaboração do questionário	83
6.2.2	Resultados	85
7	DISCUSSÃO	91
7.1	Comparação do método proposto com as abordagens da fundamentação teórica	91
7.2	Contribuições frente ao estado da arte dos temas pesquisados	92
7.3	Análise crítica do método proposto	93
7.4	Limitações desta pesquisa	93
7.5	Considerações finais	94
8	CONCLUSÃO	97
8.1	Atendimento aos objetivos	97
8.1.1	Atendimento aos objetivos específicos	97
8.2	Contribuições	98
8.3	Trabalhos futuros	99
8.3.1	Desenvolvimento de uma ferramenta direcionada para a aplicação do método proposto.	99
8.3.2	Desenvolvimento de um <i>framework</i> para direcionar a integração automatizada de modelos de engenharia de sistemas com modelos de engenharia de software.	100
8.3.3	Estudar a possibilidade de gerar modelos para o SI automaticamente a partir dos modelos do sistema espacial.	100
8.3.4	Estudar a possibilidade de evoluir o método proposto para a utilização em conjunto com PBSE (<i>Pattern-Based Systems Engineering</i> - Engenharia de Sistemas baseada em padrões).	100
	REFERÊNCIAS BIBLIOGRÁFICAS	101
	APÊNDICE A - QUESTIONÁRIO PARA AVALIAÇÃO DO MÉTODO PROPOSTO	109
	ANEXO A - ARTIGOS PUBLICADOS	117
A.1	Influências da Engenharia Simultânea de Sistemas no desenvolvimento de um sistema de informação para o processo de AIT de satélites	117

A.2	Estratégia de integração de MBSE com sistemas de informação que apoiam processos de ciclo de vida de satélites	132
A.3	Simultaneous and collaborative development of satellites and Information Systems to support their lifecycle processes	149
A.4	Análise de <i>stakeholders</i> do sistema espacial como fonte de requisitos para os Sistemas de Informação que darão suporte aos processos do seu ciclo de vida	163
A.5	Iniciativas de adesão à MBSE na área espacial	174
A.6	The relationship between the model based systems engineering models and information systems to support space products lifecycle processes . .	185

1 INTRODUÇÃO

Adotar sistemas de informação para automatizar os processos de negócio das organizações tem o potencial de melhorar a eficiência desses processos. Conforme definido por (ROSING et al., 2015), um processo de negócio é um conjunto de atividades ou tarefas relacionadas executadas para entregar um resultado esperado, como um produto ou um serviço. Nas diversas etapas pelas quais um satélite passa, desde a concepção até o descarte, conjuntos de atividades são realizados para entregar um resultado esperado, ou seja, existem processos de negócio que podem se beneficiar de um sistema de informação (SI).

A Engenharia de Sistemas Espaciais é o domínio no qual os satélites são desenvolvidos e a Engenharia de Software é o domínio no qual os sistemas de informação são desenvolvidos.

A Engenharia de Sistemas tem um leque de abordagens, técnicas e ferramentas para o desenvolvimento de sistemas complexos e a engenharia de software também tem um leque de abordagens, técnicas e ferramentas para o desenvolvimento dos diversos tipos de software, incluindo os sistemas de informação.

Embora tradicionalmente esses dois assuntos sejam abordados separadamente, a questão levantada nesta pesquisa é que, olhando para o processo de engenharia de sistemas espaciais, pode-se observar que muitas informações levantadas pelo engenheiro de sistemas para desenvolver um produto espacial também são necessárias e levantadas pelo engenheiro de software, no domínio da Engenharia de Software, para desenvolver os sistemas de informação que darão suporte aos processos de ciclo de vida desse produto. Isso sugere a possibilidade de trabalho simultâneo e colaborativo entre os dois domínios.

Assumindo o uso da Engenharia Simultânea de Sistemas (SCE) e da Engenharia de Sistemas baseada em modelos (MBSE) para a engenharia de um satélite, nesta pesquisa foi estudado o processo de Engenharia de Sistemas Espaciais para identificar como o trabalho do engenheiro de software pode ser inserido nele.

O trabalho é exemplificado com o caso do satélite Amazonia 1, desenvolvido, montado, integrado, testado e operado pelo Instituto Nacional de Pesquisas Espaciais (INPE). O processo de ciclo de vida utilizado para estudo foi o processo de Montagem, Integração e Testes (AIT, sigla do termo em inglês) desse satélite, que ocorreu no Laboratório de Integração e Testes (LIT) do INPE.

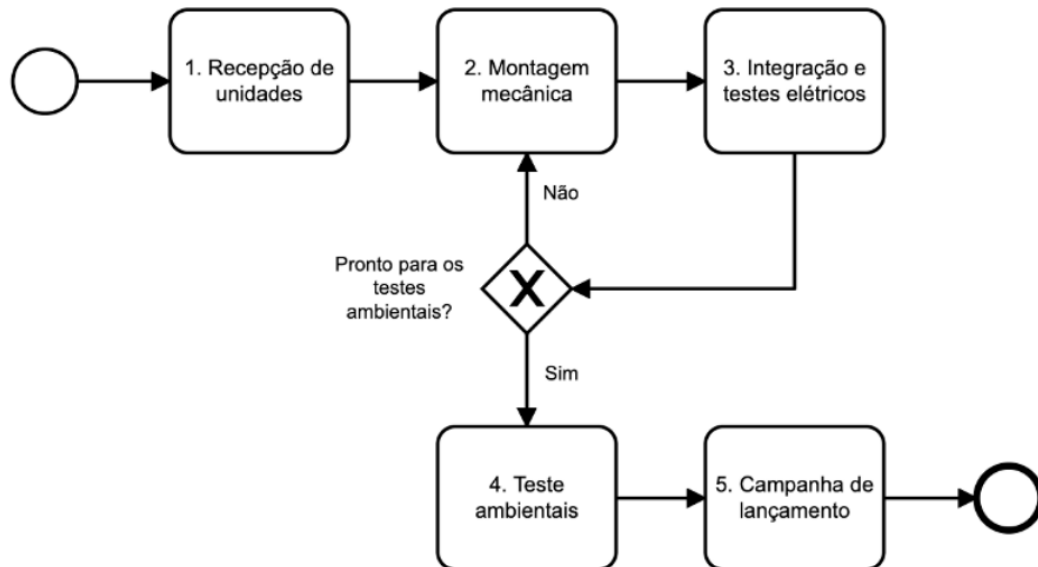
1.1 Relevância de sistemas de informação para apoiar o processo de AIT de satélites

As atividades de AIT são uma sequência lógica e inter-relacionada de eventos, cujo objetivo principal é alcançar um alto grau de confiança de que o satélite está em conformidade com os parâmetros de desempenho especificados (SILVA; LOUREIRO, 2011). AIT inclui a montagem e integração dos vários subsistemas e a realização de testes funcionais e ambientais no satélite como um sistema (SILVA; LOUREIRO, 2011).

No processo de AIT, é realizado um conjunto de atividades para que um satélite montado, integrado e testado seja entregue. Embora a AIT de satélites tenha uma sequência lógica e quase padronizada, cada organização a executa com suas particularidades, de acordo com sua cultura e estratégias, o que justifica o desenvolvimento de um sistema de informação customizado para dar suporte a esse processo.

Um processo típico de AIT de satélites é apresentado na Figura 1.1:

Figura 1.1 - Processo típico de AIT.



Fonte: Silva (2023).

Cada fase desse processo inclui uma série de atividades que envolvem diversos equipamentos, pessoas, tarefas e informações, sendo, portanto, um processo complexo.

No LIT/INPE, por exemplo, muitas dessas informações estão contidas em documentos. Os principais documentos utilizados no processo de AIT do LIT/INPE são apresentados na Tabela 1.1.

Tabela 1.1 - Principais objetivos dos principais documentos de AIT.

Especificação de teste	Detalha os requisitos de teste. Deve conter itens como objetivos dos testes, equipamentos de apoio necessários, condições, sequência de atividades, critérios de sucesso, organização e responsabilidades, relacionamento com atividades de garantia de produto e cronograma (BRANCO, 2014 apud Burger (2018)).
Requisitos de AIT	Discrimina os requisitos de AIT para cada atividade a ser desenvolvida durante a AIT do satélite (SILVA, 2011a apud Burger (2018)).
Plano de AIT	Organiza as atividades de AIT da maneira mais eficiente em termos de cronograma e orçamento. Avalia se o sistema/subsistema atende a todos os requisitos funcionais e de desempenho. Certifica a realização de todos os testes ambientais obrigatórios para aceitação do sistema (SILVA, 2011a apud Burger (2018)).
Plano de Garantia de Qualidade de AIT	Organiza e controla as atividades de AIT, fornece suporte para revisões de projetos, atividades relacionadas a testes de satélites, sistemas viabilizadores de teste e atividades durante a campanha de lançamento (SILVA, 2009a apud Burger (2018)).
Procedimentos	Os procedimentos descrevem instruções passo a passo para cada atividade de teste (derivadas das especificações do teste) (BRANCO, 2014 apud Burger (2018)).
Relatórios	Os relatórios contêm informações sobre os resultados dos testes, enfatizando o cumprimento dos requisitos correspondentes para seu fechamento na banca de controle de verificação (VCB) (BRANCO, 2014 apud Burger (2018)).

Fonte: Traduzida de Burger (2018).

A equipe de AIT do LIT/INPE utiliza uma série de ferramentas para ajudá-la a gerenciar esses documentos, as diversas tarefas que devem realizar e todas as informações que fluem por todo o processo de AIT. Essas ferramentas incluem sistemas de gerenciamento de documentos, planilhas e banco de dados locais. São ferramentas desconectadas, dedicadas a pontos específicos do processo, o que dificulta o acesso à informação e exige grande esforço da equipe para manter a consistência das informações.

Um sistema de informação (SI) dedicado ao processo de AIT tem a função de gerenciar e prover acesso às informações que fluem de ponta a ponta, mantendo-as em um banco de dados único e compartilhado entre os diversos envolvidos no processo.

Os sistemas de informação de interesse deste trabalho não devem ser confundidos com outros sistemas de software dedicados a tarefas específicas do processo de AIT, como os sistemas para envio de telecomando e recebimento de telemetrias do satélite e para monitoração e comando de GSEs (*ground support equipment*). Esses cumprem suas funções pontuais, enquanto o SI tem a função de gerenciar as informações do processo como um todo, podendo, inclusive, interfacear com esses outros sistemas, tanto para prover, como para obter informações e disponibilizá-las em qualquer ponto do processo.

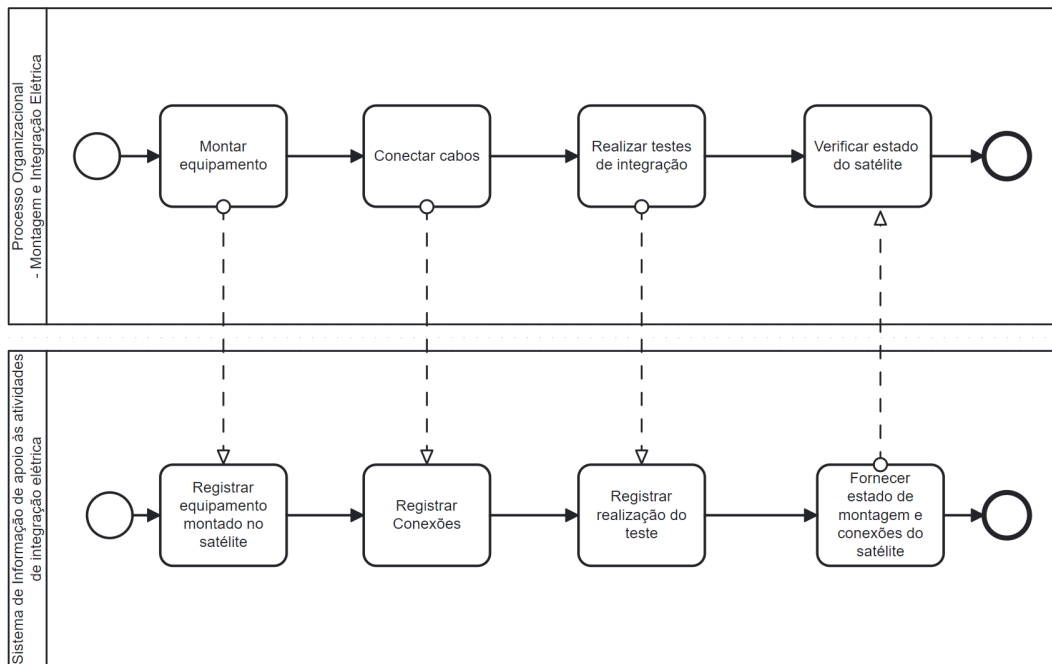
A Figura 1.2 ilustra o papel de um SI durante a realização de atividades de testes de integração elétrica em um satélite. Esses são testes realizados após a interconexão elétrica entre os equipamentos para verificar se eles estão atendendo aos requisitos.

Cada atividade do processo de AIT gera ou requer informação quando executada. São exemplos de informações geradas nas atividades da Figura 1.2:

- Montar equipamento: equipamento que foi montado, data da montagem, responsável pela montagem;
- Conectar cabos: conector, data da conexão, responsável pela conexão;
- Realizar testes de integração: data do teste, procedimento utilizado, executores, resultados e intercorrências.

São exemplos de informações requeridas pela atividade “Verificar estado do satélite”: equipamentos montados no satélite, conexões estabelecidas.

Figura 1.2 - Papel de um sistema de informação durante testes de integração elétrica em um satélite.



O papel do SI é apoiar o processo de AIT armazenando, processando e provendo essas informações. Um SI desenvolvido de forma customizada está interligado a todo processo de AIT, disponível tanto para o registro, como para o processamento e a recuperação de informação em cada ponto do processo.

O uso de um SI em substituição ao uso de diversas ferramentas desconectadas traz as vantagens de:

- Eliminar retrabalho, pois a mesma informação não precisa ser registrada mais de uma vez pelos diversos interessados;
- Eliminar inconsistências, pois a informação uma vez inserida é apenas referenciada e não repetida em outros pontos do processo;
- Facilitar o compartilhamento de informações (informações ficam disponíveis em tempo real para toda equipe, com controle de acesso, via interface do SI);
- Processar dados automaticamente, o que elimina uma série de atividades

humanas;

- Permitir a automatização de tarefas, o que minimiza a ocorrência de erros humanos;
- Criar uma base de conhecimento disponível para a organização, independentemente da rotatividade de pessoal, que favorece, entre outras coisas, o reuso das informações do processo para AIT de satélites similares.

1.2 Motivação

Embora uma série de informações provenientes do esforço do engenheiro de sistemas para desenvolver um satélite seja também de interesse do engenheiro de software que trabalha para desenvolver os sistemas de informação que irão apoiar os processos do ciclo de vida desse satélite, as atividades de desenvolvimento do sistema espacial e dos sistemas de informação são tradicionalmente realizadas de forma separada.

Tanto na Engenharia de Sistemas, como na Engenharia de Software, existe um processo de compreensão e definição dos requisitos do sistema. O Conselho Internacional de Engenharia de Sistemas (INCOSE), destaca que esse processo constitui um esforço significativo dentro do processo do desenvolvimento do sistema (INCOSE, 2015).

No caso do desenvolvimento de satélites, no domínio da Engenharia de Sistemas esse esforço inclui a interação com os diversos *stakeholders* do sistema espacial para entendimento das suas necessidades e expectativas quanto ao produto a ser desenvolvido. O engenheiro de sistemas encarregado de desenvolver um satélite realiza uma série de atividades para entendimento dos requisitos, um esforço do qual as informações resultantes também interessam ao engenheiro de software.

O fato é que o resultado do esforço realizado para a engenharia do sistema espacial tem muito significado também para a especificação dos sistemas de informação. Trabalhando isoladamente, o engenheiro de software despenderia um esforço também significativo para explorar os mesmos processos e cenários que são explorados no domínio da Engenharia de Sistemas. Unir os domínios da Engenharia de Sistemas e da Engenharia de Software significa unir esforços e evitar retrabalho.

Este trabalho é motivado pelo potencial de que o esforço realizado no domínio da Engenharia de Sistemas para desenvolver um produto espacial possa ser aproveitado no domínio da Engenharia de Software para desenvolver sistemas de informação para

dar suporte aos processos do ciclo de vida desse produto.

1.3 Objetivo

1.3.1 Objetivo geral

O objetivo geral desta pesquisa é propor um método para integrar o desenvolvimento dos sistemas de informação que darão suporte aos processos do ciclo de vida de satélites ao processo de Engenharia de Sistemas Espaciais, de forma que o esforço realizado no domínio da Engenharia de Sistemas seja aproveitado no domínio da Engenharia de Software.

A principal pergunta deste estudo é: Como o trabalho do engenheiro de software, que visa construir os sistemas de informação que darão suporte aos processos do ciclo de vida do satélite, pode ser integrado ao trabalho do engenheiro de sistemas, que visa construir esse satélite, de forma que o esforço realizado no domínio da Engenharia de Sistemas possa ser aproveitado no domínio da Engenharia de Software?

1.3.2 Objetivos específicos

- a) Identificação de oportunidades: identificar, nas abordagens de Engenharia de Sistemas e nos trabalhos já publicados nessa área, oportunidades para integrar o desenvolvimento de sistemas de informação com o desenvolvimento de satélites.
- b) Avaliação de viabilidade: avaliar a viabilidade de integrar o desenvolvimento de sistemas de informação com o desenvolvimento de satélites.
- c) Proposta de um método: propor um método que integre o desenvolvimento de sistemas de informação com o desenvolvimento de satélites.
- d) Aplicação do método: exemplificar a aplicação do método proposto com um caso real de desenvolvimento de sistema de informação para apoiar um processo do ciclo de vida de um satélite.
- e) Avaliação do método por profissionais da área de desenvolvimento de sistemas de informação: submeter o método proposto a uma avaliação por desenvolvedores de sistemas de informação.
- f) Análise crítica do método: fazer uma análise crítica do método proposto.

1.4 Metodologia e ferramentas

Esta seção situa esta pesquisa de acordo com as formas de classificação apresentadas por SILVA e MENEZES (2005) e Gil (2022), apresenta a forma como ela foi conduzida e as ferramentas utilizadas.

1.4.1 Classificação da pesquisa

De acordo com SILVA e MENEZES (2005), uma pesquisa pode ser classificada sob o ponto de vista da sua natureza e da forma de abordagem do problema. De acordo com Gil (2002), uma pesquisa pode ser classificada sob o ponto de vista de seus objetivos e dos procedimentos técnicos. A Tabela 1.2 mostra as classificações apresentadas por esses autores e situa esta pesquisa de acordo com elas. As classificações nas quais esta pesquisa se enquadra aparecem em destaque na tabela.

Tabela 1.2 - Classificação da pesquisa.

Ponto de vista	Natureza SILVA e MENEZES (2005)	Abordagem do problema SILVA e MENEZES (2005)	Objetivos Gil (2002)	Procedimentos técnicos Gil (2002)
Tipo de pesquisa	Básica	Quantitativa	Exploratória	Bibliográfica
	Aplicada	Qualitativa	Descritiva	Documental
			Explicativa	Experimental
				Levantamento
				Estudo de caso
				Pesquisa Expost-Facto
				Pesquisa-ação
			Pesquisa participante	

Conforme apresentado na Tabela 1.2, esta é uma pesquisa:

- Aplicada: pois tem objetivo de gerar conhecimentos para aplicação prática nas áreas de Engenharia de Sistemas e Engenharia de Software, dirigidos ao problema específico de desenvolvimento de sistemas de informação para apoio aos processos do ciclo de vida de satélites;
- Qualitativa: pois os dados são interpretados de forma subjetiva, não sendo

possível traduzir a interpretação em números;

- Exploratória: pois, a fim de buscar mais familiaridade com o problema, foram feitos levantamentos bibliográficos nas áreas de Engenharia de Sistemas e de Engenharia de Software, além de um trabalho de reflexão e discussão com especialistas, especialmente com o orientador da pesquisa e com a equipe de AIT do LIT;
- Bibliográfica: pois foram realizados levantamentos bibliográficos, como citado no item anterior;
- Estudo de caso: pois sua realização incluiu o caso do satélite Amazonia 1 como modelo para exemplificar o processo de engenharia de um satélite e o método proposto.

1.4.2 Condução da pesquisa e ferramentas utilizadas

Partindo do fato de que muitas informações provenientes do esforço para engenharia de um satélite são também relevantes para o desenvolvimento de sistemas de informação para apoiar os processos do ciclo de vida desse satélite, neste trabalho foi estudado o processo de Engenharia de Sistemas Espaciais com o intuito de identificar como o trabalho do engenheiro de software que desenvolve os sistemas de informação pode ser inserido nele.

Dentre as diversas abordagens para a engenharia de sistemas espaciais, a autora optou por trabalhar com a Engenharia Simultânea de Sistemas (SCE) e com a Engenharia de Sistemas Baseada em Modelos (MBSE).

A SCE foi escolhida por ser uma abordagem que se mostra mais favorável à integração das atividades do engenheiro de software ao processo de engenharia de sistemas, por antecipar os requisitos dos processos do ciclo de vida do produto para o início do projeto e por abordar também o desenvolvimento das organizações que implementam os processos do ciclo de vida.

A MBSE foi escolhida por ser uma tendência para o desenvolvimento de sistemas espaciais (SILVA; LOUREIRO, 2020) e com potencial para soluções automatizadas, característica do uso de modelos, além do fato de o uso de modelos ser familiar aos engenheiros de software, o que também favorece o objetivo da pesquisa.

Os seguintes passos foram seguidos para a realização desta pesquisa:

- Revisão de literatura para identificação de trabalhos relevantes e oportunidades de pesquisa.
 - A ferramenta utilizada para a organização da pesquisa, seleção e classificação dos artigos foi a Parsifal (PARSIFAL, 2023).
- Estudo das abordagens MBSE e SCE.
- Aplicação, suficiente para exemplificação, das abordagens MBSE e SCE para engenharia de um satélite e para estudo de um dos processos do ciclo de vida desse satélite e da organização que realizou esse processo.
 - O satélite utilizado como exemplo foi o Amazonia 1. A escolha desse satélite se deu pelo fato de que informações sobre ele estavam acessíveis para a autora em nível suficiente para o desenvolvimento desta pesquisa.
 - O processo de ciclo de vida escolhido para estudo foi o de AIT, pelo fato de que informações sobre ele estavam mais acessíveis para a autora do que sobre os demais processos.
 - Seguindo as atividades propostas pela SCE, foram construídos modelos do satélite Amazonia 1 e do LIT, organização que realizou o processo de AIT desse satélite (em nível de detalhe suficiente para a realização da pesquisa).
 - As informações para construção dos modelos requeridos pela aplicação das abordagens foram obtidas de documentos disponíveis na biblioteca do LIT e de entrevistas com integrantes da equipe que realizou o processo de AIT do satélite Amazonia 1.
 - A ferramenta utilizada para a construção dos modelos foi a IBM Rhapsody (IBM, 2023b) e a linguagem utilizada para modelagem foi a SysML.
- Modelagem de um sistema de informação para atender às necessidades do processo de AIT estudado (em nível de detalhe suficiente para a realização da pesquisa).
 - Os modelos foram construídos de acordo com o paradigma orientado a objetos.
 - A ferramenta utilizada para a construção dos modelos foi a IBM Rhapsody (IBM, 2023b) e a linguagem utilizada para modelagem foi a UML.

- A partir dos exemplos da aplicação da SCE e da MBSE, foram analisadas as influências mútuas entre as informações e modelos de interesse da engenharia de sistemas para a construção do satélite e as informações e modelos de interesse da engenharia de software para construção dos sistemas de informação.
- Observadas as influências, foi desenvolvido e proposto um método para a realização da engenharia de um satélite de modo a suportar também os projetos dos sistemas de informação de apoio aos processos do ciclo de vida desse produto.
- o método proposto foi submetido à avaliação por profissionais da área de engenharia de software, com experiência em desenvolvimento de sistemas de informação, e foi revisado após o *feedback* dos avaliadores.
 - o método foi apresentado aos avaliadores por meio de um vídeo explicativo gravado pela autora.
 - o método de avaliação foi um questionário elaborado pela autora.
 - a ferramenta para elaboração e resposta do questionário foi a Google Forms.

1.5 Organização da tese

Os capítulos restantes desta tese estão organizados da seguinte maneira:

- Capítulo 2: apresenta a fundamentação teórica desta pesquisa, abordando os temas Engenharia de Software, UML (Linguagem de Modelagem Unificada), processos de negócio e sistemas de informação, ciclo de vida de satélites, SCE (Engenharia Simultânea de Sistemas), MBSE (Engenharia de Sistemas Baseada em Modelos) e SysML (Linguagem de Modelagem de Sistemas), .
- Capítulo 3: apresenta a revisão de literatura realizada, abordando os temas “MBSE na área espacial” e “sistemas de informação no contexto do ciclo de vida de produtos complexos”.
- Capítulo 4: demonstra a viabilidade de integrar o desenvolvimento de sistemas de informação no processo da SCE.

- Capítulo 5: apresenta a proposta de um método para integrar o desenvolvimento de sistemas de informação com o desenvolvimento de sistemas espaciais.
- Capítulo 6: exemplifica a aplicação do método proposto com o caso do satélite Amazonia 1, demonstrando a especificação de um SI para apoiar o processo de AIT desse satélite e a integração de modelos do satélite com o modelo de dados do SI. Apresenta também uma avaliação do método proposto, realizada por profissionais da área de desenvolvimento de sistemas de informação.
- Capítulo 7: discute os resultados desta pesquisa comparando o método proposto com as abordagens da fundamentação teórica, ressaltando as contribuições frente ao estado da arte dos temas pesquisados; comentando sobre a utilização do método posterior ao desenvolvimento do sistema espacial; fazendo uma análise crítica do método proposto; apresentando as limitações desta pesquisa; além de apresentar outras considerações gerais.
- Capítulo 8: apresenta as conclusões quanto ao atendimento aos objetivos e as contribuições deste trabalho para a área de Engenharia de Sistemas Espaciais. Apresenta também sugestões para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Engenharia de software

A Engenharia de Software é a disciplina que se preocupa com todos os aspectos da produção de software, desde os estágios iniciais da especificação do sistema até a sua manutenção, quando o sistema já está em uso (SOMMERVILLE, 2018).

Existem vários tipos diferentes de sistemas de software, desde sistemas embarcados simples até sistemas de informação complexos. Diferentes tipos de software exigem abordagens diferentes de desenvolvimento, diferentes métodos e técnicas, porém todos precisam de engenharia de software. (SOMMERVILLE, 2018)

As atividades fundamentais da Engenharia de Software, segundo Sommerville (2018) são: especificação, desenvolvimento, validação e evolução do software.

Para guiar o trabalho da Engenharia de Software existem modelos de processo de software que definem o fluxo de todas as atividades, ações e tarefas, o grau de interação, os artefatos e a organização do trabalho a ser feito (PRESSMAN; MAXIM, 2016). O modelo de processo de software a ser utilizado vai depender do tipo de desenvolvimento de software que será realizado, não existe um modelo de processo aplicável a todos os tipos.

Entre os modelos de processo de software estão: cascata, incremental, evolutivo e integração e configuração.

O modelo em cascata sugere uma abordagem sequencial e sistemática para a realização das atividades de desenvolvimento de software, que começa com a especificação de requisitos, passando pelo planejamento, modelagem, construção, disponibilização e culmina no suporte contínuo do software concluído (PRESSMAN; MAXIM, 2016).

O modelo incremental intercala as atividades e o sistema é desenvolvido como uma série de versões (incrementos) que vão acrescentando funcionalidade à versão anterior (PRESSMAN; MAXIM, 2016). No modelo incremental requisitos básicos são atendidos em uma primeira versão e recursos e funcionalidades adicionais são entregues nas versões seguintes (PRESSMAN; MAXIM, 2016).

O modelo evolucionário, ou evolutivo, é iterativo e possibilita desenvolver versões cada vez mais completas a cada iteração (PRESSMAN; MAXIM, 2016). Um primeiro incremento é entregue com base em requisitos já compreendidos e outros incrementos

vão sendo entregues a medida que os detalhes vão sendo esclarecidos e definidos. É um modelo projetado para desenvolver um produto de software que cresce e muda (PRESSMAN; MAXIM, 2016).

Já o modelo de integração e configuração se concentra na configuração de componentes disponíveis ou sistemas reusáveis, para utilização em um novo contexto, e na utilização deles em um sistema (SOMMERVILLE, 2018).

Independentemente do modelo adotado para o desenvolvimento, dentre as atividades da Engenharia de Software, a especificação, ou Engenharia de Requisitos, é destacada como crucial por vários autores, como Sommerville (2018), Zowghi e Coulin (2005) e Castro et al. (2002), já que essa atividade lida com a difícil tarefa de projetar o software certo para o cliente.

As três atividades principais da Engenharia de Requisitos são a elicitação e análise de requisitos, a especificação de requisitos e a validação de requisitos (SOMMERVILLE, 2018).

A atividade de elicitação e análise trata-se da descoberta dos requisitos por meio da interação com *stakeholders* (SOMMERVILLE, 2018) e é atividade mais afetada pela proposta deste trabalho.

Quanto aos paradigmas para o desenvolvimento de sistemas, o paradigma da orientação a objetos, que é utilizado neste trabalho, é amplamente utilizado na engenharia de software (PRESSMAN; MAXIM, 2016).

Na orientação a objetos a perspectiva funcional e a perspectiva de dados são unificadas. O mundo real é modelado como um conjunto de objetos que podem relacionar-se uns com os outros. Um objeto contém tanto dados como comportamentos (MELO, 2011).

2.2 UML

A UML é uma linguagem utilizada para modelar softwares baseados no paradigma da orientação a objetos que consagrou-se internacionalmente como a linguagem-padrão de modelagem adotada pela indústria de Engenharia de Software (GUEDES, 2018).

A UML na sua versão 2.5 tem 14 diagramas, alguns para modelar a estrutura e outros para modelar o comportamento dos sistemas (OBJECT MANAGEMENT GROUP (OMG), 2017).

Entre os diagramas estruturais está o diagrama de Classes e entre os diagramas que modelam o comportamento está o Diagrama de Casos de Uso (MELO, 2011).

Uma Classe UML é um modelo para instanciar objetos, especifica as características da estrutura (atributos) e do comportamento (operações) desses objetos. Objetos de uma Classe contêm valores para cada atributo da Classe, de acordo com as características do atributo, por exemplo seu tipo e multiplicidade. Um diagrama de Classes modela a estrutura das Classes e o relacionamento entre elas (OBJECT MANAGEMENT GROUP (OMG), 2017).

Um Caso de Uso descreve a sequência de eventos de um ator que usa um sistema para completar um processo (JACOBSON, 1992). Atores são pessoas, grupos ou outros sistemas que interagem com o sistema de software (MELO, 2011). Um diagrama de Casos de Uso apresenta uma visão externa geral das funcionalidades que o sistema deverá oferecer (GUEDES, 2018) por meio da representação dos Casos de Uso e do relacionamento entre eles e com os atores. (MELO, 2011).

2.3 Processos de negócio e sistemas de informação

Os sistemas de informação de interesse deste trabalho são os sistemas de software destinados a apoiar processos de negócio das organizações.

Processo de negócio é um conjunto de atividades ou tarefas relacionadas que são executadas para entregar um resultado esperado, como um produto ou um serviço (ROSING et al., 2015).

Um sistema de informação, conforme definido por Sommerville (2018), é um sistema cuja principal finalidade é gerenciar e prover acesso a um banco de dados de informações.

Na definição de Stair e Reynolds (2017), sistema de informação é um conjunto de componentes inter-relacionados que coleta, processa, armazena e dissemina dados e informações. Sistemas de informação baseados em computador incluem hardware, software, redes, pessoas e procedimentos (STAIR; REYNOLDS, 2017).

Embora o software não seja o único componente de um sistema de informação, ele é seu componente-chave.

Existem dois tipos de produtos de software, os produtos genéricos e os softwares personalizados. Os produtos genéricos são desenvolvidos por uma organização de de-

envolvimento de software e vendidos no mercado para diversos clientes, enquanto os softwares personalizados são feitos sob medida para um determinado cliente (SOMMERVILLE, 2018).

Sistemas de informação podem se enquadrar nas duas categorias. Quando sua função é apoiar processos de negócio específicos de uma organização, é mais adequado que sejam personalizados, projetados tendo como guia as políticas e regras organizacionais.

Embora a maioria dos sistemas de prateleira permita alguma customização, como a escolha de módulos ou recursos, um sistema desenvolvido especificamente para o processo de negócio da própria organização tem maior potencial de atender suas necessidades e obter vantagem competitiva (STAIR; REYNOLDS, 2017).

Um sistema de informação adequado facilita o acesso às informações, elimina ou minimiza possíveis inconsistências, assim como a realização da mesma tarefa por mais de uma vez por diferentes interessados na mesma informação ao longo dos processos de negócio das organizações.

2.4 Ciclo de vida satélites

De acordo com a EUROPEAN SPACE AGENCY (ESA) (2010) projetos espaciais são normalmente divididos em 7 fases:

- Fase 0 (zero): Análise de missão
- Fase A: Análise de viabilidade
- Fase B: Definição preliminar do projeto
- Fase C: Definição detalhada do projeto
- Fase D: Produção e qualificação
- Fase E: Operação
- Fase F: Descarte

O processo de AIT de satélites acontece durante a Fase D do projeto.

2.5 SCE

SCE é uma abordagem que integra Engenharia de Sistemas e Engenharia Simultânea para o desenvolvimento de produtos complexos (LOUREIRO, 1999). Ela foi proposta em 1999 por Loureiro (1999) e veio evoluindo ao longo dos anos de acordo com as lições aprendidas com sua aplicação em diversos sistemas complexos (LOUREIRO et al., 2018). A abordagem SCE, em sua versão mais atual, implementa o framework apresentado na Figura 2.1, que é denominado *Total View Framework*.

Na Figura 2.1, a dimensão de análise inclui os subprocessos do processo de Engenharia de Sistemas que são aplicados aos elementos de cada camada da estrutura de decomposição do produto, representados na dimensão de estrutura. A dimensão de integração contém os elementos que devem ser integrados na solução sistema. O *Total View Framework* assume que os processos de Engenharia de Sistemas podem ser aplicados tanto para o desenvolvimento do produto, quanto das organizações que executam os processos do ciclo de vida desse produto (LOUREIRO et al., 2018). É a dimensão de integração que usa a engenharia simultânea (LOUREIRO et al., 2018).

A Engenharia Simultânea é uma abordagem que antecipa os requisitos dos processos do ciclo de vida para os estágios iniciais do desenvolvimento do produto, visando o desenvolvimento simultâneo do produto e das organizações que estão dentro do escopo do esforço de desenvolvimento. Porém, tradicionalmente a Engenharia Simultânea é aplicada somente em níveis muito baixos de abstração para o projeto de peças e seus respectivos processos de ciclo de vida. A SCE expande a engenharia simultânea para todas as camadas da dimensão de estrutura (LOUREIRO et al., 2018).

2.5.1 Atividades propostas pela SCE

Considerando a divisão proposta pela ESA (EUROPEAN SPACE AGENCY (ESA), 2010) para projetos de sistemas espaciais, as atividades da SCE ocorrem dentro das fases 0 a C.

Com base em uma declaração de missão, a SCE começa identificando necessidades e medidas de efetividade e, a partir disso, uma solução de missão. Em um próximo passo, são identificados os processos do ciclo de vida do sistema de interesse a ser desenvolvido. Cada processo do ciclo de vida pode ser ainda decomposto em cenários.

Figura 2.1 - *Total View Framework*.



Fonte: Traduzida e adaptada de Loureiro et al. (2018).

Para cada processo ou cenário do processo do ciclo de vida, requisitos de sistema são derivados através das análises de *Stakeholders* e de Requisitos e das análises de contexto Funcional e de Implementação, tanto para o produto como para a organização que implementa o processo do ciclo de vida.

Dos requisitos funcionais e não funcionais do sistema são derivadas a arquitetura funcional e a arquitetura de implementação, tanto do produto como da organização. Para cada elemento da arquitetura do sistema são identificados requisitos e medidas de performance e então, chega-se ao projeto detalhado do sistema.

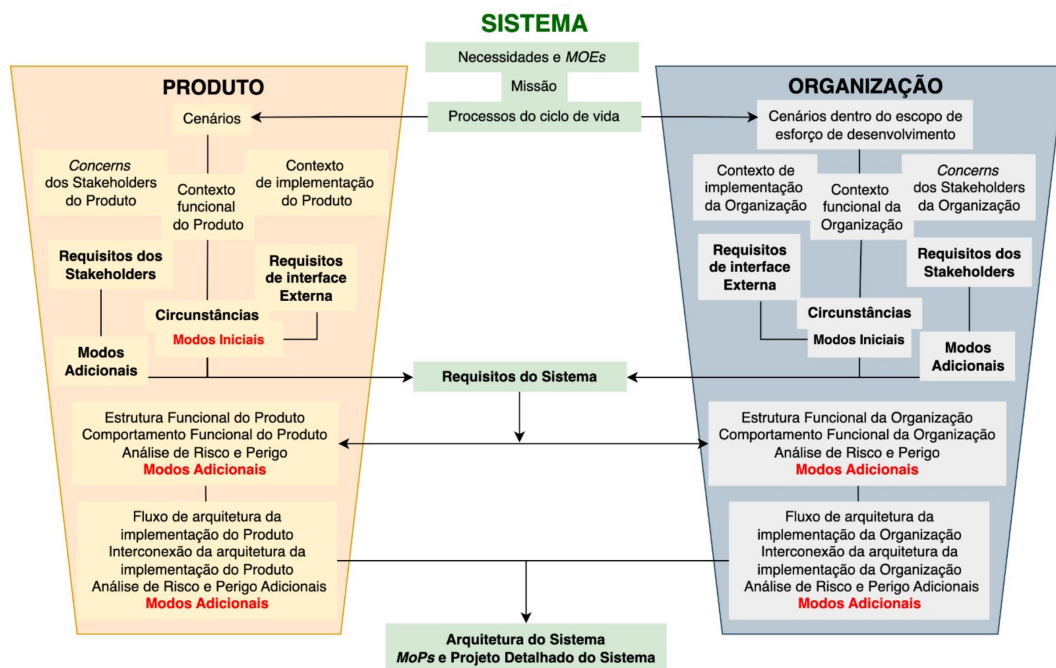
As atividades propostas pela SCE são ilustradas na Figura 2.2.

2.6 MBSE

A Engenharia de Sistemas Baseada em Modelos, ou MBSE, é uma opção para projetar sistemas complexos em substituição à abordagem tradicional baseada em documentos.

Com a abordagem baseada em documentos, os artefatos da engenharia de sistemas, como conceito de operação (ConOps), especificações de requisitos, matrizes de rastreabilidades e verificação de requisitos, especificações de casos de testes, entre outros, são produzidos manualmente pelos engenheiros de sistemas na forma de documentos de texto, planilhas, diagramas e apresentações de forma desarticulada (DELLIGATTI, 2013). O gerenciamento da configuração desses artefatos é feito por meio de repositórios também desarticulados, o que torna essa tarefa trabalhosa e cara (DELLIGATTI, 2013).

Figura 2.2 - Método SCE.



Fonte: Traduzida e adaptada de Loureiro et al. (2018).

Na MBSE, modelos são a principal fonte de informação e o principal meio de troca de informação entre os envolvidos no processo de engenharia, substituindo nessas

funções os documentos utilizados na abordagem tradicional de engenharia de sistemas (FRIEDENTHAL et al., 2014). Os mesmos artefatos gerados na abordagem baseada em documentos podem ser gerados automaticamente a partir dos modelos utilizando as próprias ferramentas de modelagem, porém o modelo do sistema serve como um repositório central para decisões de projeto (DELLIGATTI, 2013).

Modelos são construídos para representar requisitos, estrutura e comportamento dos sistemas (FRIEDENTHAL et al., 2014).

Como benefícios da MBSE, são citados por Friedenthal et al. (2014): a melhoria na comunicação, a redução no risco de desenvolvimento, o aprimoramento da qualidade, o aumento da produtividade e a melhoria na transferência de conhecimento.

A MBSE se apoia em três pilares: linguagem de modelagem, método de modelagem e ferramenta de modelagem.

Linguagem de modelagem: Uma linguagem de modelagem define os tipos de elementos que você pode colocar em seu modelo e os relacionamentos permitidos entre eles (DELLIGATTI, 2013). Linguagens de modelagem gráfica definem ainda o conjunto de notações que você pode usar para exibir os elementos e relacionamentos em diagramas (DELLIGATTI, 2013).

Método de modelagem: um roteiro, um conjunto documentado de tarefas, a ser executado pela equipe de modelagem para criar um modelo de sistema. O método garante que todos da equipe estejam construindo o modelo do sistema de forma consistente e trabalhando em direção a um ponto final comum (DELLIGATTI, 2013). O método é definido pela equipe conforme escopo e objetivo do sistema a ser modelado. São exemplos de métodos de modelagem: INCOSE *Object-Oriented System Engineering Method* (OOSEM); *Weilkiens System Modeling* SYSMOD method; IBM *Telelogic Harmony-SE*.

Ferramentas de Modelagem: São ferramentas projetadas para cumprir as regras de uma ou mais linguagens de modelagem (DELLIGATTI, 2013). Ferramentas de modelagem diferem de ferramentas de diagramação como Visio, Schematic, SmartDraw, ProcessOn e outras. Com ferramentas de diagramação criamos diagramas, formas, porém não há nenhum modelo subjacente a esses diagramas que garanta a consistência automatizada entre eles (DELLIGATTI, 2013). Já nas ferramentas de modelagem, criamos modelos, ou seja, um conjunto de elementos e seus relacionamentos e opcionalmente podemos criar um conjunto de diagramas para visualizar o modelo

criado (DELLIGATTI, 2013).

A Engenharia de Sistemas espaciais é uma tarefa desafiadora. Considerando os potenciais benefícios da MBSE, produtos espaciais, por serem produtos complexos e com altos custos envolvidos em seu desenvolvimento, são bons candidatos para aplicação dessa abordagem.

2.7 SysML

A linguagem de modelagem é um dos três pilares da MBSE. Modelos podem ser criados em várias linguagens, sob a escolha da equipe. Uma das linguagens mais utilizadas atualmente é a SysML.

SysML é uma linguagem de modelagem gráfica de uso geral que pode ser base para diferentes métodos da MBSE (INCOSE, 2015).

Sendo uma linguagem de modelagem gráfica, a SysML define o conjunto de notações que podem ser usadas para exibir os elementos e relacionamentos em diagramas (DELLIGATTI, 2013).

A SysML possui nove tipos de diagramas que suportam a especificação, análise, design, verificação e validação de sistemas (IBM, 2012). Esses diagramas não se limitam apenas a implementação dos sistemas, também é ferramenta para comunicar o projeto aos usuários desses sistemas (DELLIGATTI, 2013).

Os nove diagramas se agrupam em diagramas de comportamento, de estrutura e de requisitos (FRIEDENTHAL et al., 2009). A seguir, são descritos os diagramas utilizados neste trabalho.

Entre os diagramas de estrutura, estão o Diagrama de Definição de Blocos (BDD) e o Diagrama de Blocos Internos (IBD). Esses dois diagramas são utilizados para representar a estrutura do sistema (FRIEDENTHAL et al., 2009).

O bloco é a unidade básica de estrutura em SysML e pode ser usado para representar hardware, software, instalações, pessoal ou qualquer outro elemento do sistema. Tanto a estrutura, quanto o comportamento desses elementos são representados no Bloco (FRIEDENTHAL et al., 2009).

Um BDD foca na definição e descreve a hierarquia do sistema (IBM, 2012). O IBD foca no uso (FRIEDENTHAL et al., 2014) e descreve a estrutura interna de um sistema em termos de suas partes, portas e conectores. Enquanto um Bloco é a

definição de um elemento, o uso ou instância desse elemento é chamado Parte (IBM, 2012).

Entre os diagramas de comportamento está o Diagrama de Atividades. O Diagrama de Atividades representa o fluxo de dados e o controle entre as atividades (FRIEDENTHAL et al., 2009).

A SysML inclui também o Diagrama de Requisitos, que permite uma construção gráfica para representar requisitos baseados em texto e relacioná-los com outros elementos do modelo. Esse diagrama captura hierarquias e derivação de requisitos, além de fornecer uma ponte entre as ferramentas típicas de gerenciamento de requisitos e os modelos de sistema (FRIEDENTHAL et al., 2009).

3 REVISÃO DE LITERATURA

A revisão de literatura foi realizada com o objetivo de levantar o que se tem discutido no meio científico sobre dois temas específicos relacionados com esta pesquisa:

1. MBSE na área espacial;
2. Sistemas de informação no contexto do ciclo de vida de produtos complexos.

Também buscou-se encontrar estudos sobre um possível relacionamento entre esses dois temas.

As buscas foram realizadas nas bases Web of Science e Scopus em 2 de agosto de 2021 para embasar este trabalho e foram revistas em 17 de setembro de 2023 para uma atualização de resultados. A janela temporal para o tema 1 foi de 10 anos e para o tema 2 não houve limitação de data, já que a quantidade de artigos publicados sobre o tema 2 é pequena.

Para o tema 1, além das bases citadas, também foram consultadas publicações em sites e blogs oficiais da NASA (National Aeronautics and Space Administration), ESA (European Space Agency), Thales Alenia Space, Airbus Defense and Space e publicações do INPE.

3.1 MBSE na área espacial

3.1.1 Palavras-chave e critérios de exclusão

Para o tema 1, foi utilizada a palavra-chave “MBSE”, e suas variações de grafia, combinada com os seguintes termos, e suas variações de grafia, referentes a produtos espaciais:

- satellite
- spacecraft
- space product
- space system

Para análise dos artigos, foram excluídos dos resultados os artigos que se encaixam nos seguintes critérios:

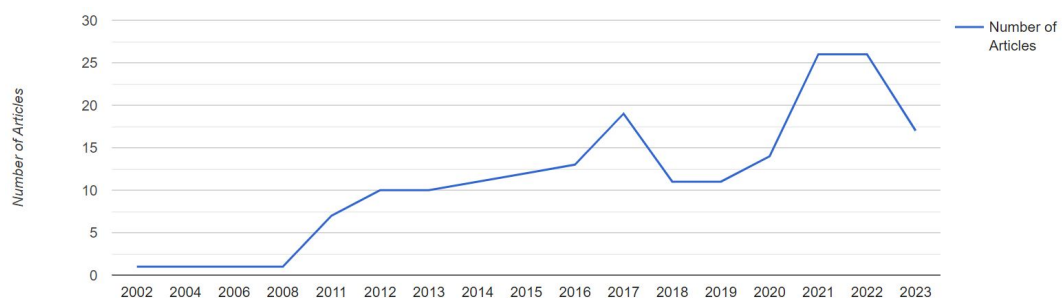
- apesar do termo MBSE ser mencionado no artigo, tem um papel menos relevante, ou seja, o foco do artigo é outro.
- o artigo foi publicado há mais de 10 anos.

Ao todo foram 160 artigos considerados pertinentes e aceitos para análise a fim de entender o estado da arte desse tema de pesquisa, que é apresentado na Subseção 3.1.2.

3.1.2 Estado da arte do tema “MBSE na área espacial”

A Figura 3.1 apresenta uma bibliometria que desconsidera o critério da data para dar uma visão geral da quantidade de publicações ao longo dos anos. Esse gráfico aponta o crescente interesse pelo tema.

Figura 3.1 - Quantidade de trabalhos sobre o tema “MBSE na área espacial” - bases Scopus e Web of Science em 17/09/2023.



Compilação dos resultados das pesquisas realizadas em 17/09/2023 nas bases Scopus e Web of Science (*trabalhos presentes nas duas bases foram considerados apenas uma vez; *pesquisa sem limitação de data).

Quanto às publicações dos últimos 10 anos, os 160 artigos analisados podem ser agrupados em 5 categorias de interesse de pesquisa apresentadas no gráfico da Figura 3.2, que ilustra o estado da arte desse tema.

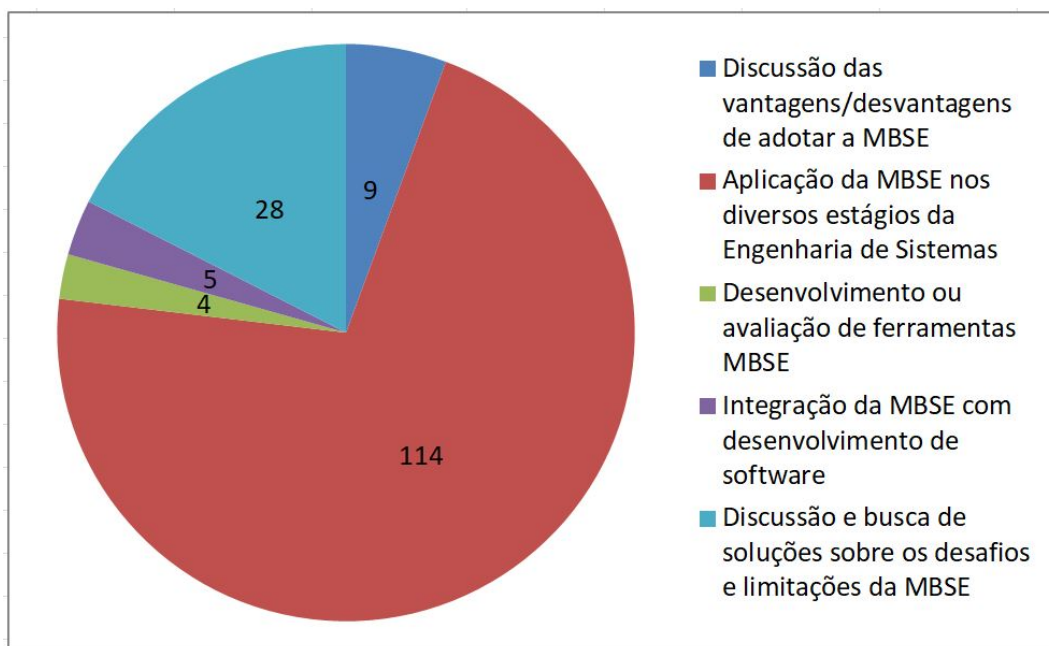
Além das publicações no meio acadêmico, a relevância do tema também é revelada pelas iniciativas de adesão à MBSE pelos principais atores da Engenharia Espacial.

Visando identificar essas iniciativas, a autora realizou e publicou uma pesquisa bibliográfica (SILVA; LOUREIRO, 2020) que incluiu, além de artigos científicos de revistas e congressos, publicações em *sites* e *blogs* oficiais de duas das principais

agências espaciais, NASA e ESA, e em duas das principais indústrias envolvidas no ramo espacial, Thales Alenia Space e Airbus Defense and Space. Iniciativas no INPE também foram investigadas.

Nas subseções seguintes, são apresentadas as iniciativas em cada uma dessas organizações, podendo ser observado um cenário no qual o uso da MBSE mostra-se uma tendência para a Engenharia de Sistemas Espaciais.

Figura 3.2 - Estado da arte do tema “MBSE na área espacial”.



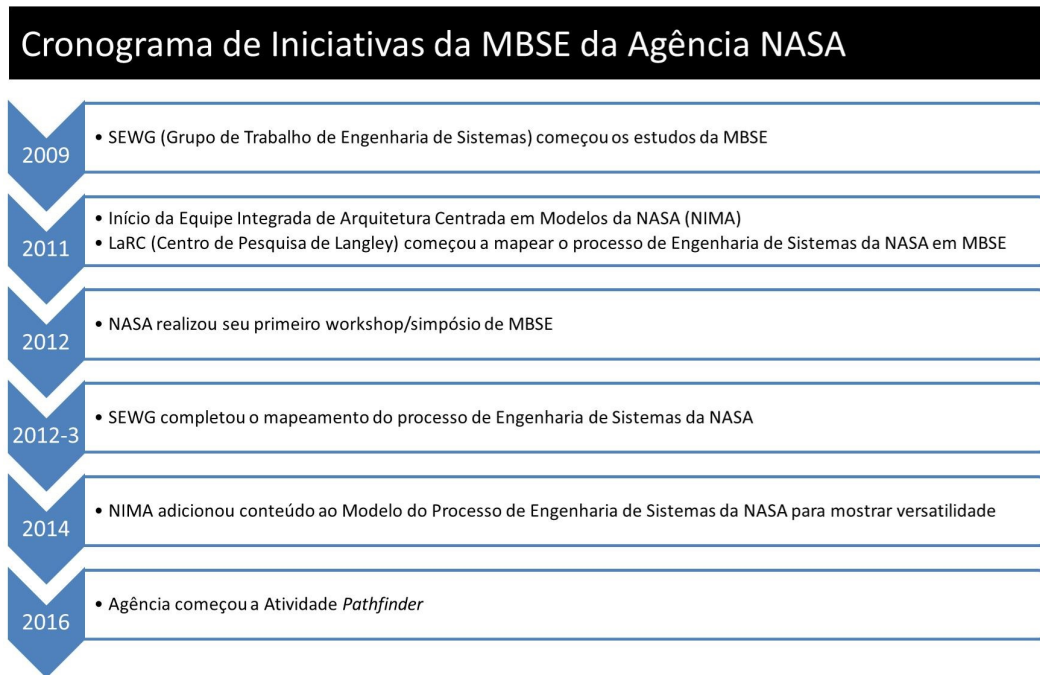
3.1.2.1 Iniciativas de adesão à MBSE na NASA

A Figura 3.3 apresenta em uma linha do tempo as iniciativas da NASA relativas à MBSE de 2009 a 2016.

O avanço do uso da MBSE nas aplicações da NASA e o desejo de capturar lições aprendidas para guiar os próximos passos motivaram o estabelecimento, em 2016, de um estudo chamado MBSE *Pathfinder* (WEILAND; HOLLADAY, 2017). Conduzido pelo Centro de Engenharia e Segurança da NASA, esse estudo avaliou a facilidade com que a MBSE poderia ser implantada e aplicada em quatro áreas de interesse. Quatro equipes trabalhando em paralelo por oito meses demonstraram potenciais

benefícios da utilização da MBSE para seus *stakeholders* e para a NASA como um todo (HOLLADAY et al., 2019), como mostrado na Tabela 3.1.

Figura 3.3 - Iniciativas de adesão à MBSE na NASA - linha do tempo.



Fonte: Traduzida de Parrott (2016).

A abordagem adotada para o MBSE *Pathfinder* forneceu uma série de lições aprendidas e construiu uma comunidade de usuários forte e colaborativa (WEILAND; HOLLADAY, 2017). Em 2017 foi estabelecida a parte 2 do MBSE *Pathfinder* com os objetivos de implementar as lições aprendidas na parte 1, focar no desenvolvimento de produtos de engenharia de sistemas ao longo do ciclo de vida, prover integração com várias ferramentas analíticas, como CAD, e demonstrar e comunicar valor para os projetos e Centros (HOLLADAY et al., 2019).

Tabela 3.1 - Benefícios da MBSE demonstrados pelas equipes do Pathfinder 1.

Área de interesse	Benefícios demonstrados
Uso e reuso da arquitetura da missão para uma campanha de missões humana-para-Marte.	A equipe demonstrou como os princípios de desenvolvimento ágil (semelhantes aos usados para engenharia de software) podem ser usados para criar modelos de sistema.
Fabricação aditiva para desenvolvimento de motores de foguetes.	A equipe gerou automaticamente requisitos e especificações de desempenho atualizados para permitir que os engenheiros rastreiem com mais facilidade as rápidas mudanças do sistema inerentes ao processo de fabricação aditiva.
Projeto de elemento de missão de um <i>Mars lander</i> (veículo para pousar em Marte).	A equipe gerou um subconjunto do pacote de revisão técnica para uma Revisão Preliminar de projeto.
Sombreamento do fluxo da missão de um projeto de foguete de sondagem.	A equipe modelou com sucesso as informações da Revisão de Projeto de uma missão anterior, descobrindo informações ausentes e inconsistências nos planos de teste.

Fonte: Adaptada de [Holladay et al. \(2019\)](#).

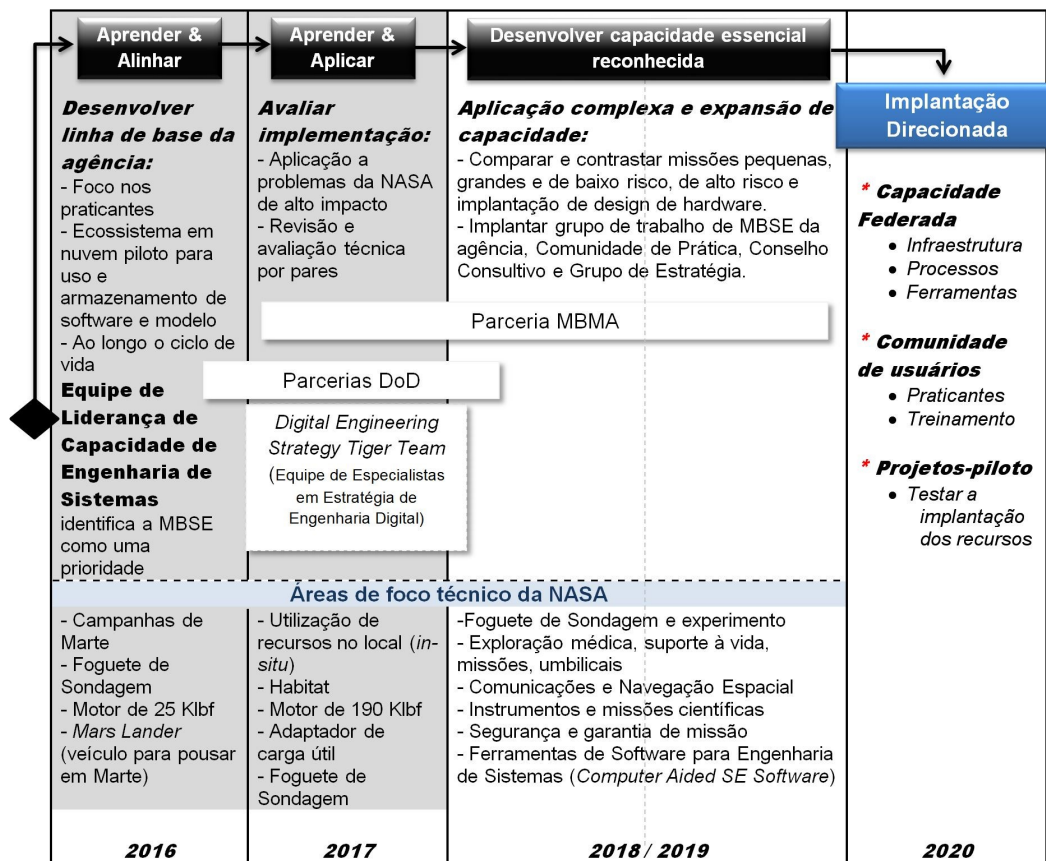
Como na Parte 1 do MBSE *Pathfinder*, na Parte 2 os participantes também foram organizados em equipes virtuais representando vários centros da NASA. Dessa vez foram seis equipes em seis diferentes áreas de interesse. Quatro equipes continuaram o trabalho em relação ao ano anterior e duas novas equipes com dois novos tópicos foram acrescentadas: a equipe do ISRU (*in-situ resource utilization*) e Arquitetura de Habitat Integrada e a equipe do Adaptador de Carga SLS ([HOLLADAY et al., 2019](#)).

Após essas duas experiências, a equipe de liderança do MBSE Pathfinder mudou o nome da iniciativa para Iniciativa de Infusão e Modernização de Engenharia de Sistemas Baseada em Modelo (MIAMI), a fim de distinguir seu escopo expandido do trabalho do MBSE Pathfinder de 2016 e 2017. O foco mudou de “preparação para transição” para “implementação”. Vários programas e projetos na NASA passaram a aplicar a MBSE ([HOLLADAY et al., 2019](#)).

[Holladay et al. \(2019\)](#) apresenta uma figura (Figura 3.4) com um resumo do planejamento para implantação da MBSE na NASA de 2016 a 2020. Na parte superior da figura estão as *tag lines* para cada ano e na metade superior são mostradas breves descrições da abordagem de implementação. As áreas do MBSE *Pathfinder*, na seção inferior, indicam a estreita relação com as missões reais da NASA. O centro da figura destaca parcerias com outras organizações. A parceria MBMA (*Model-based*

Mission Assurance) é realizada com o Escritório de Segurança e Garantia de Missão da NASA, e a parceria DoD e *Digital Engineering Strategy Tiger Team* são parcerias com o Departamento de Defesa. O marco em 2020 é uma implantação direcionada de infraestrutura, processos e ferramentas, comunidade de usuários e projetos piloto.

Figura 3.4 - Resumo do planejamento de cinco anos (2016 a 2020) da NASA para implantação da MBSE.



Fonte: Traduzida de Holladay et al. (2019).

3.1.2.2 Iniciativas de adesão à MBSE na ESA

Na ESA a MBSE foi implementada como uma inovação dentro da missão, também inovadora, e.Deorbit, uma missão para remover detritos espaciais da órbita Sol-Síncrona entre 600 e 800km de altitude.

Considerada uma missão complexa envolvendo requisitos de robótica, orientação,

navegação e controle, teve início em 2012, tendo um primeiro modelo, um modelo da arquitetura física, sido implementado durante a Pré-Fase A no CDF (*Concurrent Design Facility*) (WOLAHAN; BIESBROEK, 2017), uma instalação de ponta onde especialistas de várias disciplinas aplicam métodos de engenharia simultânea ao projeto de futuras missões espaciais (EUROPEAN SPACE AGENCY (ESA), 2020).

Após o estudo de viabilidade interna, contratos industriais da Fase A foram firmados com três grandes Integradores de Sistemas europeus, Airbus Defense and Space, OHB e Thales Alenia Space. Como parte da Fase A, foi solicitado aos contratados usar MBSE para modelar as arquiteturas físicas e funcionais (JESSICA, 2017).

Após a conclusão bem-sucedida da Revisão Preliminar de Requisitos, foi recomendado estender o uso da MBSE também ao nível do sistema (JESSICA, 2017). A Fase B1 começou em 2015 e terminou em 2017. Nessa fase foram feitos contratos com a Airbus e OHB, que foram incentivadas pela ESA a aplicar a MBSE sempre que possível. Cada contratado adotou diferentes ferramentas e metodologias para aplicação da MBSE, porém com os mesmos propósitos (JESSICA, 2017):

- Analisar o problema e definir seus limites;
- Derivar e gerenciar os requisitos;
- Identificar os recursos/funções exigidos pelo sistema;
- Desenvolver a arquitetura funcional, lógica e física;
- Seguir os métodos de verificação/validação necessários;
- Iniciar simulações diretamente do modelo MBSE;
- Estabelecer uma troca de dados de “verdade única” no nível do sistema.

Esses foram os primeiros exemplos da ESA dos estudos da Fase B1, usando a MBSE, incluindo a arquitetura física da plataforma e a carga útil, juntamente com a arquitetura funcional e lógica da missão, com links diretos para algumas das ferramentas usadas para simulações (JESSICA, 2018).

3.1.2.3 Iniciativas de adesão à MBSE na Thales Alenia Space

A Thales Alenia Space é uma empresa que fornece soluções de alta tecnologia para telecomunicação, navegação, observação da Terra, gerenciamento ambiental, exploração, ciência e infraestrutura orbital (THALESGROUP, 2020).

Em 2005, a Thales começou a investir na MBSE visando a melhoria do seu processo de engenharia. Investiu fortemente nos aspectos metodológicos e de ferramentas, tendo como parte dos resultados o método de desenvolvimento baseado em modelo chamado Arcadia e uma ferramenta de suporte a esse método, chamada Capella, que foi disponibilizada em 2014 em todas as unidades da Thales pelo mundo (BONNET et al., 2016).

A solução Arcadia/Capella é inspirada nos conceitos da SysML, porém com meios de expressão reduzidos. Essa simplificação é possível devido ao seu escopo mais preciso. Arcadia/Capella não cobre todo o espectro das atividades de design, o foco está no projeto de arquitetura (justificativa de componentes/interfaces por meio da análise funcional, avaliação não funcional antecipada da arquitetura e preparação das atividades de integração e validação), excluindo modelagem comportamental de baixo nível ou simulação (BONNET et al., 2016) e (ECLIPSE, 2017).

A simplificação ou especialização dos conceitos provenientes da SysML foi pensada visando uma curva de aprendizado facilitada (ECLIPSE, 2017) que pudesse atingir os engenheiros de sistemas que não possuem experiência em engenharia de software, já que as origens da SysML na orientação-a-objetos mostraram-se claramente um obstáculo para sua adoção pelos engenheiros de sistemas da Thales que não eram familiares com o mundo de desenvolvimento de software (BONNET et al., 2016).

O objetivo final da Arcadia/Capella não é ter especialistas em modelagem, mas sim promover uma mudança cultural, direcionada para a MBSE, nas práticas dos engenheiros de sistemas (BONNET et al., 2016) e (ECLIPSE, 2017).

Podemos dizer que alguns conceitos da solução Arcadia/Capella são especializações de conceitos da SysML para melhor representar a realidade da Thales. Foi feita uma interpretação da especificação SysML de forma a prover os engenheiros de todos os meios de expressão que eles precisassem, porém evitando sobrecarregá-los com complexidade desnecessária, ajudando-os a seguir as metodologias da Thales e unificando a maneira como as arquiteturas de software e de sistemas são modeladas na Thales (CALIO et al., 2016).

3.1.2.4 Iniciativas de adesão à MBSE na Airbus Defence and Space

Segundo Sharples (2018), a Airbus Defense and Space tem uma rigorosa abordagem de MBSE que é utilizada em todos os seus projetos. E, segundo Gregory et al. (2020), a Airbus liderou vários projetos buscando desenvolver técnicas de MBSE e aplicá-

las a projetos do mundo real, sendo o mais significativo deles o desenvolvimento do modelo em SysML da missão e.Deorbit da ESA.

Em 2012 a Airbus sentiu a necessidade de uma ferramenta para dar suporte aos novos métodos baseados nos princípios da MBSE. Anteriormente, contava com ferramentas comerciais prontas disponíveis, que considerava bem estabelecidas e com bons resultados, porém complexas o suficiente para limitar sua implantação. A nova ferramenta deveria ser essencialmente flexível o suficiente para se adaptar ao vocabulário e aos conceitos da Engenharia de Sistemas da Airbus e para se ajustar facilmente às necessidades em evolução, à medida que os processos e métodos amadureciam gradualmente (MAGNET, 2016).

Para suprir essa necessidade, a Airbus desenvolveu e implantou uma ferramenta customizada, baseada na ferramenta de código aberto Eclipse Papyrus (ECLIPSE, 2019) que fornece, entre outras coisas, suporte completo a SysML (MAGNET, 2016).

A Papyrus ofereceu a capacidade de criar rapidamente uma ferramenta personalizada que restringe alguns dos recursos gerais da SysML permitindo que os usuários trabalhem apenas com conceitos específicos do domínio. A linguagem SysML foi, portanto, personalizada para um formato mais facilmente compreensível pelos projetistas (MAGNET, 2016).

A ferramenta desenvolvida, chamada FAST, dá suporte às análises funcional e operacional, utilizando os diagramas de Atividades e de Blocos Internos da SysML fornecidos nativamente pela Papyrus. FAST permite a definição das funções necessárias para operar o sistema, bem como sua decomposição funcional e suas interfaces (MAGNET, 2016).

A extensibilidade geral da Papyrus também permitiu a inclusão de outras ferramentas, como verificações internas de consistência e geração automática de documentos (MAGNET, 2016).

3.1.2.5 Iniciativas de adesão à MBSE no INPE

No INPE, uma metodologia de Engenharia de Sistemas Baseada em Modelos foi adotada na missão EQUARS (*Equatorial Atmosphere Research Satellite*) em desenvolvimento (HOFFMANN et al., 2018). EQUARS é uma missão que tem como objetivo “promover o avanço do conhecimento científico em Aeronomia Equatorial, com ênfase no entendimento dos fenômenos físicos que perturbam o comportamento

médio do plasma ionosférico” (HOFFMANN et al., 2018).

Além da aplicação prática na missão EQUARS, vários projetos desenvolvidos no INPE foram e estão sendo utilizados como base para pesquisas envolvendo a MBSE. Algumas dessas pesquisas são citadas a seguir:

- Coicev e Loureiro (2019): apresentam uma proposta de utilização de MBSE e SysML aplicadas a um estudo de caso de análise de um componente de um Equipamento de Suporte Elétrico de Solo (EGSE) típico utilizado na AIT de satélites com o objetivo de descrever o fluxo de processos utilizado nas análises, fornecendo um pano de fundo metodológico para a aplicação da notação SysML de forma prática para desenvolvimento de EGSEs.
- Franco (São José dos Campos, 2018): propõe um modelo de referência da interface entre o satélite e o veículo lançador, em linguagem SysML, para servir como base a futuros esforços na utilização da MBSE para definição e controle dessa interface.
- Aquino et al. (2018): realizaram um projeto de Iniciação Científica que teve como objetivo a implementação da MBSE para o nano satélite AESP-14, primeira plataforma CubeSat brasileira, desenvolvido pelo ITA (Instituto Tecnológico de Aeronáutica) com apoio do INPE.
- Burger (2018): apresenta em sua tese de doutorado um *framework* conceitual que considera o uso da MBSE para fornecer entradas para o planejamento da Montagem, Integração e Testes (AIT) de satélites.

3.2 Sistemas de informação no contexto do ciclo de vida de produtos complexos

3.2.1 Palavras-chave e critérios de exclusão

Para o tema 2, a *string* de busca foi construída cruzando o termo “information system(s)” (e seus termos alternativos) com o termo “complex product(s) life cycle” (e seus termos alternativos).

Os termos alternativos para “information system” são listados a seguir:

- “software tool(s)” e “software system(s)”: termos mais genéricos, porém com potencial de retornar resultados relevantes para este trabalho, já que

sistemas de informação também costumam ser referenciados como ferramentas de software ou sistemas de software, apesar de uma ferramenta, ou sistema, de software não ser seu único componente e de os termos englobarem também vários outros tipos de sistemas.

- “product lifecycle management” (PLM): termo que se refere à atividade empresarial de gerenciar, da maneira mais eficaz, os produtos de uma empresa durante todo o seu ciclo de vida (STARK, 2016). Esse termo também tem potencial de retornar resultados relevantes, pois softwares são desenvolvidos e utilizados para esse fim.
- “configuration management”: termo que se refere a atividade de gerenciar mudanças e que comumente envolve sistemas de software. Quando associado aos termos referentes ao ciclo de vida de produtos complexos também tem potencial de retornar resultados de interesse para este trabalho.

Os termos alternativos para “complex product(s) life cycle” são listados a seguir (quando aplicável, foram utilizados também o plural e variações de grafia). Termos específicos da área espacial foram acrescentados à lista de alternativas para ampliar as possibilidades de resultados relevantes, especialmente relevantes para a área espacial. A palavra mais específica “development” foi utilizada como variação à “life cycle” para ampliar as possibilidades de resultados, especialmente relacionados ao processo de desenvolvimento dos produtos complexos.

- “complex product development”
- “space system development”
- “space system life cycle”
- “satellite development”
- “satellite life cycle”
- “space product development”
- “space product life cycle”
- “spacecraft development”
- “spacecraft life cycle”

Foram excluídos já na *string* de busca, os resultados contendo “*geographic information system*”, e plural, termo que se refere a sistemas de informação geográficos, que não se encaixam no interesse deste trabalho.

Também foram excluídos dos resultados os artigos que se encaixam nos seguintes critérios:

- discorrem sobre softwares embarcados (não é o tipo de sistema de software de interesse deste trabalho)
- discorrem sobre softwares de simulação (não é o tipo de sistema de software de interesse deste trabalho)
- apesar de citar as palavras-chave da pesquisa, o estudo não tem foco no sistema de informação

Ao todo foram 18 artigos, considerados pertinentes e aceitos para análise a fim de entender o estado da arte desse tema de pesquisa, que é apresentado na Subseção 3.2.2.

3.2.2 Estado da arte do tema “sistemas de informação no contexto do ciclo de vida de produtos complexos”

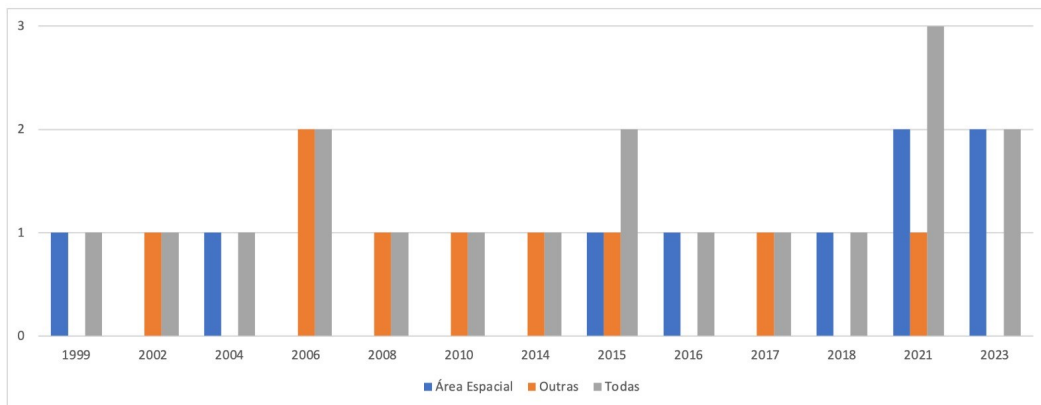
Dos 18 artigos dentro dos critérios para análise, 2 são da própria autora já abordando o tema desta pesquisa de doutorado, [Silva e Loureiro \(2023\)](#) e [Silva e Loureiro \(2018\)](#). Os demais artigos podem ser agrupados em 4 categorias, de acordo com seu foco, que apontam as principais áreas de estudo nesse tema ao longo dos anos:

- compartilhamento/integração de dados/recursos/ferramentas durante o processo de ciclo de vida do produto ([JIANG; MAIR, 2002](#)), ([CHIANG et al., 2006](#)), ([MALETZ et al., 2008](#)), ([XU et al., 2014](#)), ([SKOBELEV et al., 2015](#)), ([BERNABEI et al., 2015](#)), ([VIOLANTE et al., 2017](#)), ([XUEMEI; XIAOLANG, 2021](#)) , ([CZIEP et al., 2021](#))
- gerenciamento de projeto ([ZHAO; YIN, 2006](#)), ([ZHAO et al., 2010](#))
- gestão de documentação ([PRATOMO et al., 2021](#))
- ferramentas dedicadas a atividades específicas dos processos do ciclo de vida do produto

- processo de verificação e testes (REID et al., 2016), (FOX et al., 2023)
- processo de desenvolvimento (LOUREIRO; LEANEY, 1999), (FUJIKAWA; OGASAWARA, 2004)

Na Figura 3.5 é apresentada uma bibliometria que aponta a distribuição dos trabalhos no tempo, incluindo os trabalhos da autora, e se são ou não da área espacial.

Figura 3.5 - Quantidade de trabalhos sobre o tema “sistemas de informação no contexto do ciclo de vida de produtos complexos” - bases Scopus e Web of Science em 17/09/2023.



Compilação dos resultados das pesquisas realizadas em 17/09/2023 nas bases Scopus e Web of Science (*trabalhos presentes nas duas bases foram considerados apenas uma vez; *pesquisa sem limitação de data).

3.3 Trabalhos correlatos

Dos trabalhos analisados, 3, assim como este, envolvem a MBSE e o desenvolvimento de software.

A principal diferença entre este trabalho e os 3 trabalhos correlatos está no tipo de software alvo. O foco deste trabalho é a integração do desenvolvimento de SI com a MBSE, enquanto Halvorson e Thomas (2022) e Guo et al. (2014) tratam da integração do desenvolvimento de softwares embarcados com a MBSE e Fischer et al. (2017) focam no desenvolvimento de uma ferramenta para MBSE.

A Tabela 3.2 resume semelhanças e diferenças entre os trabalhos. Podemos observar nessa tabela, a SysML como a linguagem comum entre 3 dos 4 trabalhos comparados e a variação das ferramentas de modelagem utilizadas.

Tabela 3.2 - Tabela comparativa de trabalhos correlatos.

	Este trabalho	Halvorson e Thomas (2022)	Guo et al. (2014)	Fischer et al. (2017)
Tipo de relação com a MBSE	Especificação de SI a partir de modelos da MBSE; link entre a MBSE e a base de dados do SI.	Geração de software embarcado como um artefato MBSE.	Geração automática de software embarcado a partir de modelos de simulação conectados à modelos da MBSE.	Desenvolvimento de ferramenta para MBSE
Tipo de software	Sistema de informação	Software embarcado	Software embarcado	Ferramenta para MBSE
Linguagem MBSE	SysML	SysML	SysML	GSEL (generic system engineering language)
Diagrama(s) explorado(s)	Requisitos, BDD, IBD	BDD, IBD	Requisitos, paramétrico	O nome não foi especificado.
Ferramenta MBSE	IBM Rhapsody	MagicDraw	Artisan Studio	Virtual Satellite
Caso de estudo	Satélite de médio porte	CubeSat	Nano satélite	Plataforma multimissão

Comparando este trabalho com as propostas de Halvorson e Thomas (2022) e Guo et al. (2014) que trabalham para integrar na MBSE o desenvolvimento de software embarcado, é preciso destacar que a natureza do software alvo implica na necessidade de estratégias de integração com grandes diferenças. O software embarcado vai no hardware do produto em desenvolvimento na MBSE, está diretamente ligado à implementação do produto, faz parte do produto e pode estar diretamente ligado aos modelos do produto. Quanto ao SI para apoio aos processos do ciclo de vida do produto, embora modelos da implementação do produto também sejam de seu interesse, está fora do produto e dentro da organização que implementa o processo do ciclo de vida apoiado. As abordagens para integração desses dois tipos de software na MBSE são necessariamente diferentes.

4 DEMONSTRAÇÃO DA VIABILIDADE DE INTEGRAR O DESENVOLVIMENTO DE SISTEMAS DE INFORMAÇÃO NO PROCESSO DA SCE

O engenheiro de software encarregado de desenvolver um sistema de informação, para qualquer finalidade, precisa se engajar o suficiente com os *stakeholders* desse futuro sistema para entender seus requisitos e entregar um produto que satisfaça as expectativas.

Considere, por exemplo, um sistema de informação para processamento de pedidos. Para desenvolver esse sistema, o engenheiro de software precisa trabalhar para ter respostas para perguntas como: por quais etapas o pedido passa dentro do fornecedor até ser entregue ao cliente? Como é organizado o estoque de produtos? Por corredores? Armários? Prateleiras? Quais são as formas possíveis de embalar os produtos? Como os produtos serão identificados? Etiquetas? Código de barras? QR code? Quais são os métodos de entrega utilizados pelo fornecedor?

Da mesma forma que no domínio da Engenharia de Software, o engenheiro de sistemas com a missão de desenvolver um satélite precisa se envolver o suficiente com os *stakeholders* do sistema espacial para entender suas necessidades e requisitos e satisfazer suas expectativas.

Considere, por exemplo, um satélite para observação da Terra que fornecerá imagens do planeta. Que resolução de imagem é necessária? Qual é a área de cobertura necessária? Com que frequência o satélite deve visitar o mesmo ponto da Terra? Qual infraestrutura está disponível para verificação e validação do sistema? Quais características o satélite precisa ter para ser verificado e validado nessa infraestrutura? Qual será o veículo lançador? Como o satélite será transportado para a plataforma de lançamento? Essas são algumas das perguntas para as quais o engenheiro de sistemas vai trabalhar para buscar as respostas.

Como apresentado na Seção 2.5, a SCE propõe um processo a ser percorrido pelo engenheiro de sistemas para chegar a essas respostas e a um projeto detalhado do sistema.

Considerando o desenvolvimento de um satélite, seguindo a proposta da SCE, o engenheiro de sistemas precisa explorar os processos do ciclo de vida desse satélite para especificá-lo e projetá-lo. O engenheiro de software que está trabalhando em um sistema de informação para dar suporte aos processos do ciclo de vida desse

satélite também precisará explorar esses mesmos processos, embora com finalidade distinta. Enquanto o engenheiro de sistemas trabalha para desenvolver o satélite, o engenheiro de software trabalha para desenvolver os sistemas de informação.

Essa necessidade de explorar os mesmos processos é o primeiro indício da viabilidade de integrar o desenvolvimento dos sistemas de informação ao processo da SCE.

Para demonstrar pontualmente que o trabalho do engenheiro de software pode ser inserido no processo de engenharia de satélites, neste capítulo percorremos o processo da SCE. Atividades propostas por essa abordagem de engenharia de sistemas são exemplificadas e é apontada a viabilidade do trabalho conjunto do engenheiro de software em cada uma delas.

Para o apontamento das atividades do engenheiro de software, foi assumido um desenvolvimento baseado no paradigma da orientação a objetos e o uso da linguagem UML.

O satélite utilizado como exemplo é o AMAZÔNIA 1, desenvolvido e operado pelo INPE.

A Seção 4.1 apresenta a declaração de missão e os processos do ciclo de vida do satélite AMAZONIA 1.

A Seção 4.2 exemplifica a análise de *stakeholders* do produto e da organização e a análise de requisitos, demonstrando que essas atividades também atuam como fonte de requisitos para os sistemas de informação e que o engenheiro de software pode trabalhar nelas simultaneamente com o engenheiro de sistemas.

A Seção 4.3 exemplifica a análise funcional da organização e do produto, demonstrando que a análise funcional da organização também atua como fonte de requisitos para os sistemas de informação e que, por sua vez, os modelos funcionais do produto não exercem influência significativa nos sistemas de informação.

A Seção 4.4 demonstra a relação entre as arquiteturas de implementação do produto e de implementação da organização, modeladas pelo engenheiro de sistemas, e a estrutura de dados do sistema de informação, modelada pelo engenheiro de software.

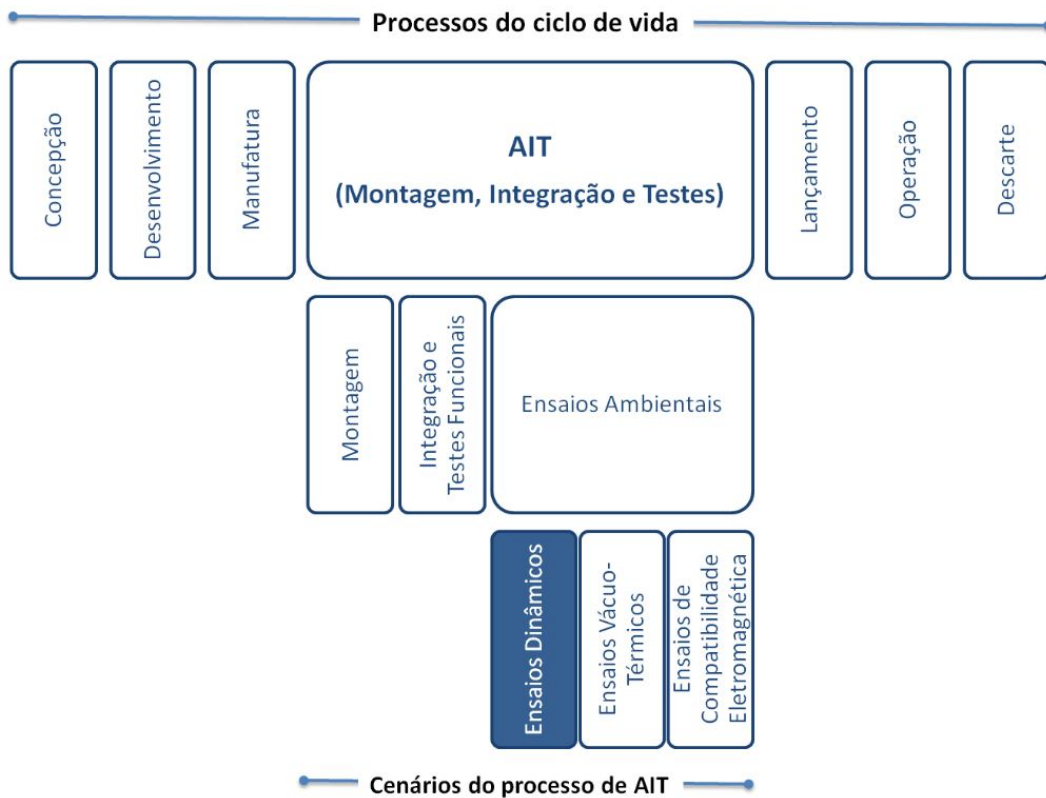
4.1 Definição da missão e processos do ciclo de vida do satélite

Tomando o satélite AMAZONIA 1 como exemplo, nesta seção é apresentada sua missão e os processos do seu ciclo de vida.

AMAZONIA 1 é o primeiro satélite de observação da Terra, estabilizado em três eixos, inteiramente projetado, montado, integrado, testado e operado pelo Brasil. A missão do AMAZONIA 1 é gerar imagens do planeta e fornecer dados para monitoramento ambiental, principalmente na Amazônia (INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS (INPE), 2022a).

A Figura 4.1 mostra os processos do ciclo de vida do satélite, destacando o processo de Montagem, Integração e Testes (AIT) e também o cenário de ensaios dinâmicos, que são usados para exemplificar os próximos passos.

Figura 4.1 - Processos do ciclo de vida do satélite AMAZONIA 1.



O cenário de Ensaios Dinâmicos é um desdobramento do cenário de Ensaios Ambientais, no qual o satélite é exposto a condições controladas em laboratório com o objetivo de verificar e garantir a sua capacidade de suportar as condições ambientais em todas as fases de sua vida útil, desde o momento do lançamento até o término previsto de sua operação em órbita. No cenário de Ensaios Dinâmicos, o satélite

passa por ensaios de vibração e ensaios acústicos (INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS (INPE), 2022b).

As atividades de AIT para o AMAZÔNIA 1 foram realizadas no Laboratório de Integração e Testes (LIT) do Instituto Nacional de Pesquisa Espaciais (INPE). O LIT é, portanto, a organização que implementa o processo de AIT do satélite AMAZÔNIA 1.

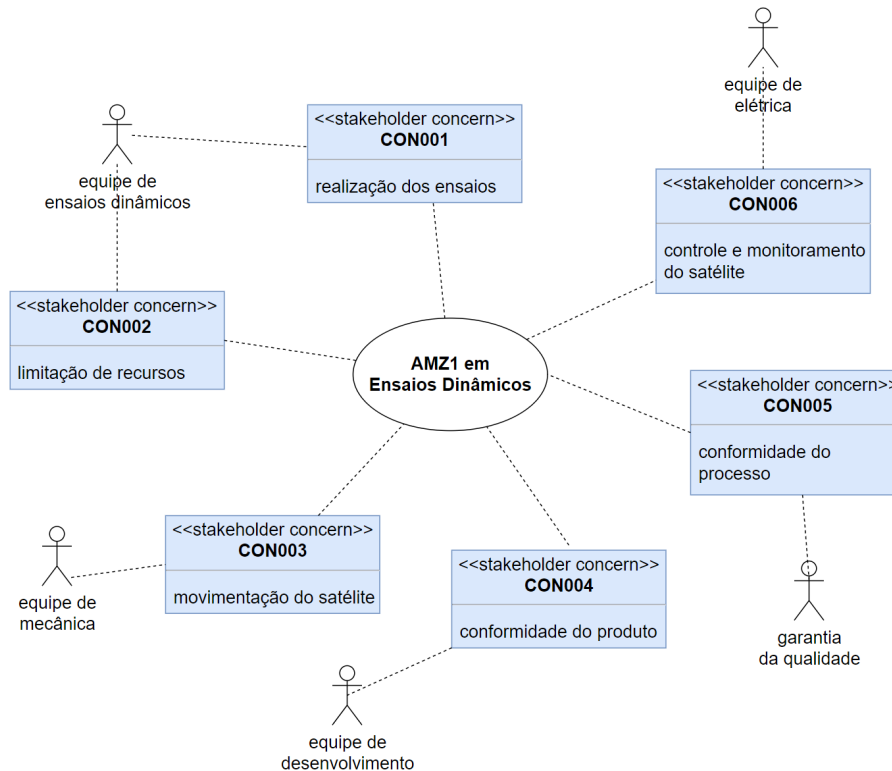
4.2 Análises de *stakeholders* e de requisitos e suas influências nos sistemas de informação

Para a Análise de *Stakeholders*, a SCE propõe que para cada processo ou cenário do ciclo de vida do sistema sejam identificados os *stakeholders* do produto e da organização e seus *concerns* (desejos, vontades, objetivos, interesses, preocupações). *Stakeholders* do produto são aqueles que afetam ou são afetados pelo produto naquele processo ou cenário. *Stakeholders* da organização são aqueles que afetam ou são afetados pela organização que implementa aquele cenário.

A Figura 4.2 apresenta, em um diagrama de requisitos da SysML, exemplos de *stakeholders* do produto AMAZONIA 1 no cenário de Ensaio Dinâmico e de seus *concerns*.

Nesse cenário, uma das preocupações da equipe mecânica é movimentar e posicionar o satélite nos vibradores, a equipe de ensaios dinâmicos tem o objetivo de realizar os ensaios previstos para o satélite, além de preocupações com a limitação de recursos para a realização dos ensaios, enquanto a equipe elétrica precisa de condições adequadas para acompanhar o estado do satélite durante os ensaios e interagir com ele caso algo saia fora do esperado. Por sua vez, a equipe de garantia da qualidade e a equipe de desenvolvimento acompanham de perto todas as atividades. A primeira com a preocupação de assegurar que os processos ocorram de acordo com os requisitos definidos nos procedimentos da qualidade e a segunda com a preocupação de garantir que o produto esteja de acordo com o especificado no projeto de engenharia.

Figura 4.2 - *Stakeholders* do produto e seus *concerns*.



A Figura 4.3 apresenta, em um diagrama de requisitos da SysML, exemplos de *stakeholders* do LIT, organização que implementa o processo de AIT, e de seus *concerns*.

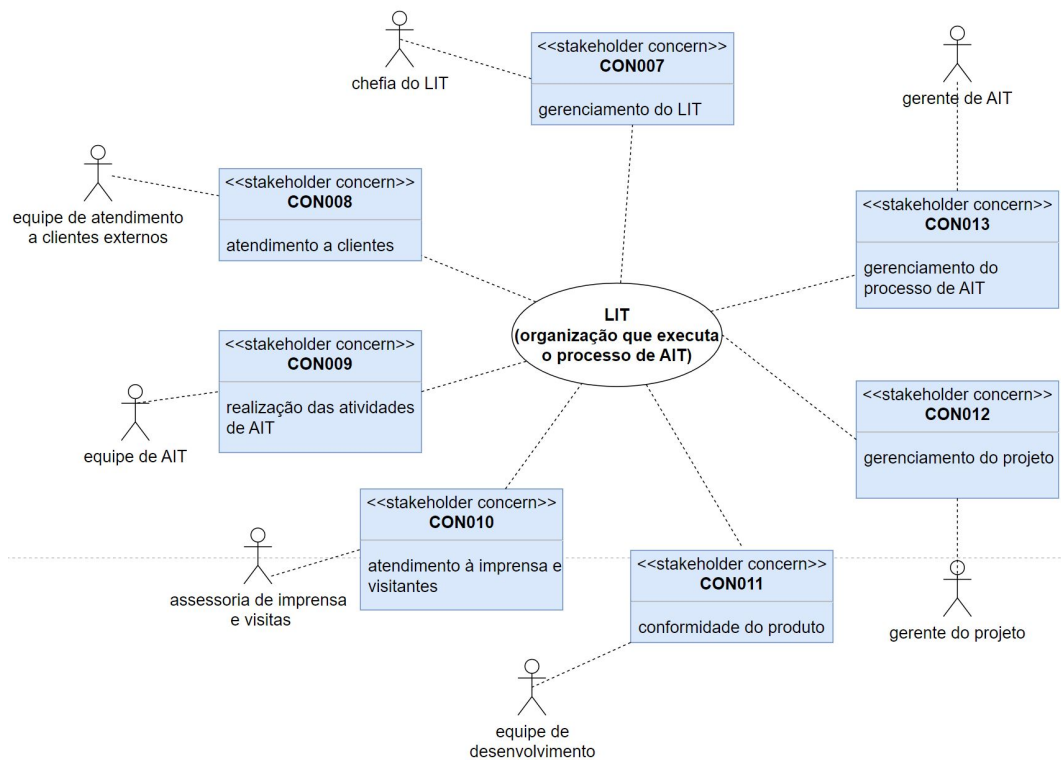
Além dos *concerns* da equipe de AIT, durante esse processo há preocupações tanto com a gerenciameto das atividades de AIT, quanto com o gerenciameto do projeto do satélite como um todo e com o gerenciameto do próprio laboratório. Todos os responsáveis por essas atividades afetam e são afetados pela organização; eles têm, portanto, interesses a serem considerados pelo engenheiro de sistemas.

Ainda na Figura 4.3, temos como *stakeholders* a equipe de assessoria de imprensa e visitas, que também precisa fazer uso do laboratório durante a AIT de satélites e a equipe de atendimento a clientes externos. Além de atender aos programas espaciais, o LIT também disponibiliza os seus meios de ensaio e testes ao setor produtivo e a sociedade técnico-científica para a qualificação de produtos, equipamentos e novas tecnologias. A equipe responsável pelo atendimento aos clientes externos precisa

também fazer uso dos meios de ensaio e teste e, portanto, precisam ter seus interesses considerados pelo engenheiro de sistemas.

A equipe de desenvolvimento, com o interesse em garantir a conformidade do produto, também afeta e é afetada pela organização de AIT, pois há uma interação constante entre ela e o LIT durante a AIT de satélites.

Figura 4.3 - *Stakeholders* da organização e seus *concerns*.



Na continuidade do processo de engenharia proposto pela SCE, os *concerns* são desdobrados em necessidades e então em requisitos de *stakeholder*. Os requisitos de *stakeholder* descrevem o que os *stakeholders* devem ser capazes de fazer (*capabilities*) ou restrições que eles impõe ao sistema (*constraints*). Enquanto as necessidades são declarações mais genéricas, os requisitos de *stakeholder* devem ser escritos cuidadosamente para serem simples, concisos, compreensíveis e verificáveis.

A Figura 4.4 e a Figura 4.5 ilustram esse procedimento em diagramas de requisitos da SysML.

Figura 4.4 - Desdobramento de *concerns* em requisitos de *stakeholder* (*stakeholders* do produto).

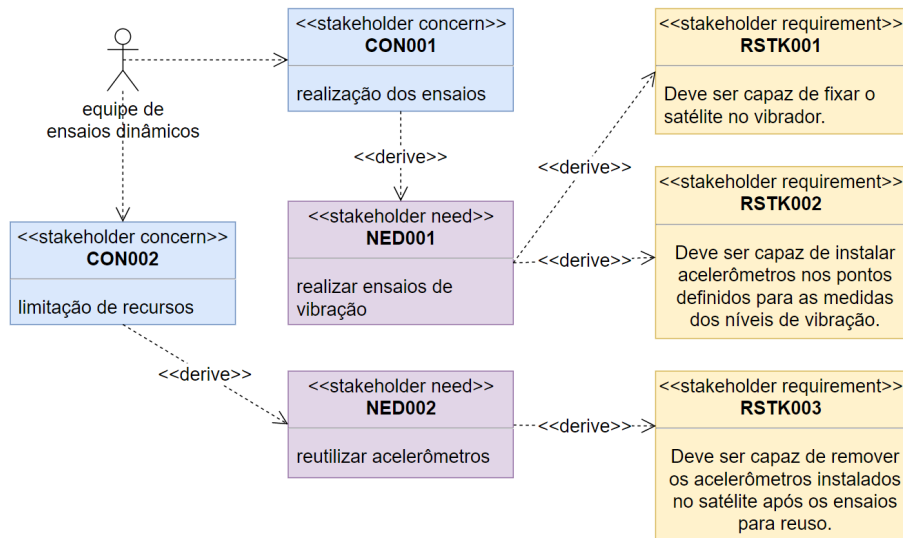
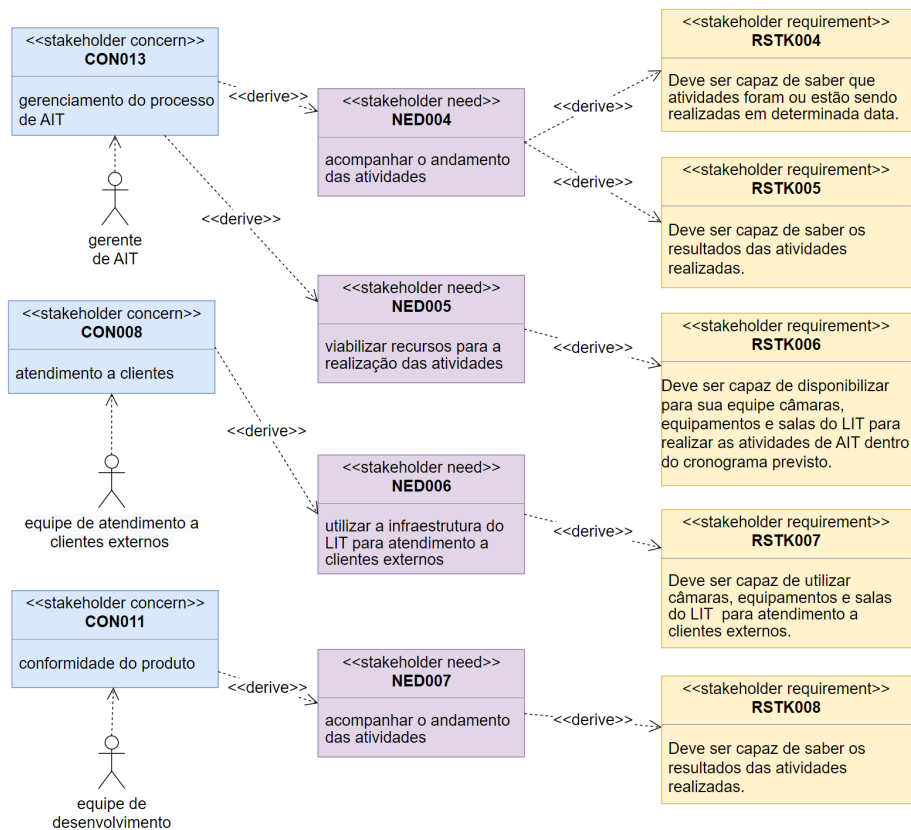


Figura 4.5 - Desdobramento de *concerns* em requisitos de *stakeholder* (*stakeholders* da organização).



Após identificar os requisitos de *stakeholder*, o próximo passo para o engenheiro de sistemas é derivar a partir deles os requisitos de sistema. Os requisitos de sistema descrevem o que o sistema deve fazer ou que características deve ter para que os requisitos de *stakeholder* sejam satisfeitos.

Por exemplo, para o requisito RSTK001, para que a equipe de Ensaio Dinâmico consiga fixar o satélite no vibrador, o satélite deverá possuir uma interface mecânica que, além de ser compatível com o sistema de fixação do satélite no veículo lançador, tenha as características físicas (layout de furação e dimensões) da mesa vibratória. Tal interface deverá também ter características mecânico-dinâmicas adequadas para a realização dos ensaios previstos.

Enquanto o engenheiro de sistemas trabalha para transformar os requisitos do *stakeholder* em requisitos de sistema, com base nas mesmas informações, o engenheiro de software pode trabalhar para identificar Casos de Uso para o sistema de informação (SI) que dará suporte ao processo do ciclo de vida em questão. Se outros processos do ciclo de vida também serão contemplados pelo SI, requisitos dos *stakeholders* do sistema espacial em um determinado cenário podem também influenciar o SI nesses outros processos.

Por exemplo, para o requisito RSTK001, as informações das características físicas dos sistemas de vibração do LIT, assim como do sistema de fixação do satélite no veículo lançador (identificadas em outro cenário, cenário de Lançamento), deverão estar disponíveis no SI para o projetista da interface mecânica. Após o projeto e fabricação da interface mecânica, o correspondente procedimento para a fixação do satélite sobre a mesa vibratória também deverá estar disponível no SI.

Para os requisitos de *stakeholder* RSTK002 e RSTK003, instalar e remover acelerômetros, o engenheiro de sistemas deverá pensar em requisitos de sistema que garantam que os pontos do satélite definidos para as medidas dos níveis de vibração estejam acessíveis para que a equipe de Ensaio Dinâmico possa instalar e remover os acelerômetros, ou seja, não poderá haver barreira física, como cabamentos ou equipamentos bloqueando o acesso a estes pontos.

Para esses mesmos requisitos, RSTK002 e RSTK003, o engenheiro de software entende que funcionalidades para controle da instalação e remoção de acelerômetros devem estar disponíveis no SI.

Agora, vamos analisar os requisitos dos *stakeholders* da organização, mostrados na

Figura 4.5. Para os requisitos RSTK004 e RSTK005, o engenheiro de sistema deverá se preocupar com o que o LIT deverá prover para que esses requisitos sejam satisfeitos, como entregar relatórios de resultados das atividades (ensaios, testes) após a conclusão de cada uma delas. Lembrando que a solução sistema para a SCE engloba produto e organização, nesse caso o requisito de sistema é um requisito a ser satisfeito pela organização LIT.

Para esses mesmos requisitos, o engenheiro de software consegue identificar que o SI deverá prover funcionalidades para o registro da execução das atividades e seus resultados, assim como para a visualização desses registros pelo gerente de AIT. Voltando ao diagrama da Figura 4.3, pode-se identificar que visualizar os resultados das atividades também é de interesse da Equipe de Desenvolvimento, já que a conformidade do produto é uma preocupação para ela.

Já para os requisitos RSTK006 e RSTK007, o engenheiro de sistemas deverá se preocupar com requisitos de sistema que possibilitem o compartilhamento da infraestrutura entre as atividades de AIT do satélite AMAZONIA 1 e de outros programas que possam estar acontecendo em paralelo, além das atividades para clientes externos ao INPE, como por exemplo, o LIT não poderá agendar serviços de ensaios dinâmicos para clientes externos ao INPE nas datas previstas para os ensaios do satélite.

Baseado nesses requisitos, o engenheiro de software entende que o SI deverá prover funcionalidades para agendamento da utilização da infraestrutura e que essa agenda deve ser compartilhada entre a equipe de AIT e as equipes que atendem clientes externos ao INPE.

4.3 Análise funcional e sua influência nos sistemas de informação

A SCE propõe que seja realizada uma análise funcional tanto para o produto, em cada processo do seu ciclo de vida, quanto para a organização que implementa os processos.

A análise funcional consiste em um conjunto de atividades que resulta em uma descrição da estrutura funcional e do comportamento funcional requeridos para o sistema.

A estrutura revela uma visão estática das funções do sistema, enquanto o comportamento está preocupado com a execução temporal, lógica, causal e condicional das funções.

4.3.1 Análise funcional da organização

4.3.1.1 Estrutura funcional da organização que implementa o processo de AIT

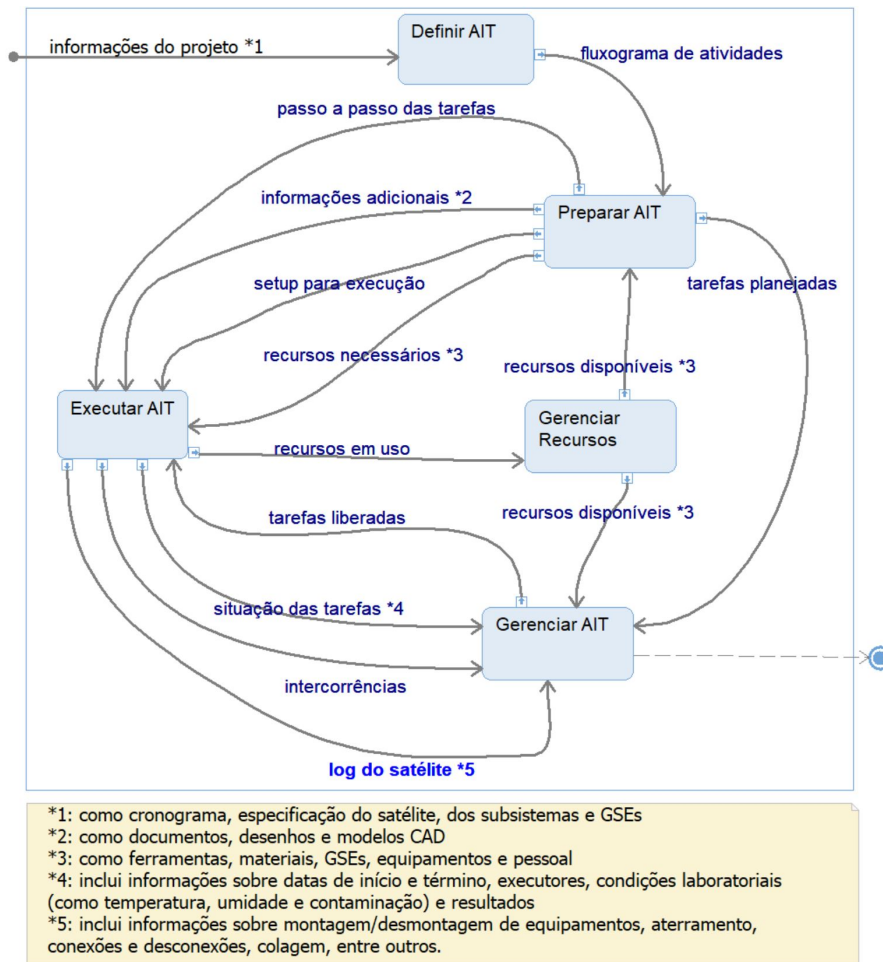
Realizar AIT de satélites é a principal função da organização LIT e pode ser dividida em 5 macro funções (INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS (INPE), 2017):

- Definir AIT: são definidas as atividades que serão executadas (desde as macro atividades até o nível de procedimento) e a sequência em que essas atividades devem ser executadas.
- Preparar AIT: detalham-se as atividades, passo a passo, identificando e apontando os recursos necessários aos operadores para executá-las, como *setup*, equipamentos do satélite e GSEs (*Ground Support Equipment*).
- Gerenciar AIT: o gerente de AIT organiza a execução das atividades, elabora o cronograma, libera tarefas para execução e trata das questões e problemas que ocorrem durante o processo.
- Executar AIT: as atividades definidas são efetivamente realizadas e sua realização é documentada.
- Gerenciar recursos: os recursos necessários para o processo de AIT, como materiais, ferramentas, pessoal, GSEs, são gerenciados e mantidos em condições de uso.

A Figura 4.6 ilustra esta estrutura funcional num diagrama de atividades da SysML, focando no fluxo de informações de alto nível entre as macro funções.

Durante a realização das atividades na macro função “Executar AIT”, por exemplo, uma série de informações é gerada e precisa estar disponível para o gerente de AIT na macro função “Gerenciar AIT”. Entre essas informações podemos citar: que equipamentos foram montados ou desmontados, que cabos foram conectados ou desconectados, se houve alguma intercorrência, quando e por quem cada atividade foi executada, quais foram os resultados dos testes, etc.

Figura 4.6 - Estrutura funcional da organização que implementa o processo de AIT do AMAZONIA 1.



A descrição da estrutura funcional da organização também é de interesse do engenheiro de software, pois as informações que fluem entre os processos devem ser contempladas pelo SI. O fluxo de informações é certamente uma fonte de requisitos para o SI. O SI deverá atuar para armazenar e prover essas informações.

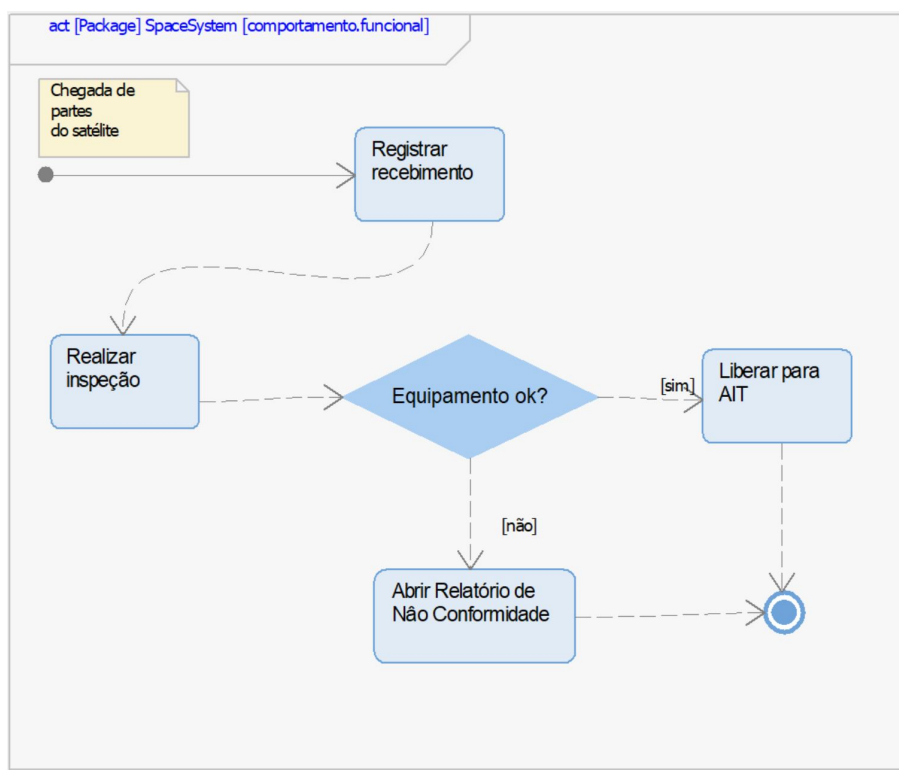
4.3.1.2 Comportamento funcional da organização que implementa o processo de AIT

O comportamento funcional da organização é claramente uma fonte de requisitos para o SI, pois define a dinâmica do processo que será apoiado por ele. Mapear o comportamento funcional é uma atividade comum para o desenvolvedor de sistemas de informação e também é uma atividade proposta pela SCE para o engenheiro de

sistemas.

A Figura 4.7 mostra um Diagrama de Atividades SysML representando uma pequena parte do comportamento funcional da organização que implementa o processo de AIT. Equipamentos que chegam para AIT passam por inspeção antes de serem liberados para uso e em caso de problemas na inspeção, um relatório de não conformidade é aberto.

Figura 4.7 - Parte do modelo de comportamento funcional da organização que implementa o processo de AIT do AMAZONIA 1.



O comportamento funcional da organização, além de apontar atividades a serem apoiadas pelo SI, aponta também regras de negócio a serem implementadas no SI, de forma que atenda o comportamento normal do fluxo do processo e também as exceções.

Uma das funções do sistema de informação é ajudar a garantir que as regras de negócio sejam cumpridas. Embora nem todas as ações incorretas possam ser bloqueadas pelo sistema de informação, ele pode cercá-las o quanto possível e minimizar suas

ocorrências.

Um exemplo de regra de negócio a ser implementada pelo sistema de informação que emerge do modelo de comportamento funcional da organização ilustrado na Figura 4.7:

- O equipamento não pode ser montado no satélite se não tiver sido aprovado na inspeção.

4.3.2 Análise funcional do produto

A análise funcional do produto é uma atividade da SCE com pouco potencial de gerar artefatos para o SI, já que o funcionamento do produto não está na área de atuação do SI. Exemplos de funções a serem desempenhadas pelo satélite AMAZÔNIA 1 são:

- gerar energia;
- armazenar energia ;
- distribuir energia;
- controlar atitude;
- estabilizar atitude;
- controlar e determinar sua órbita;
- realizar manobras orbitais.

4.4 Arquitetura de implementação e seu relacionamento com os sistemas de informação

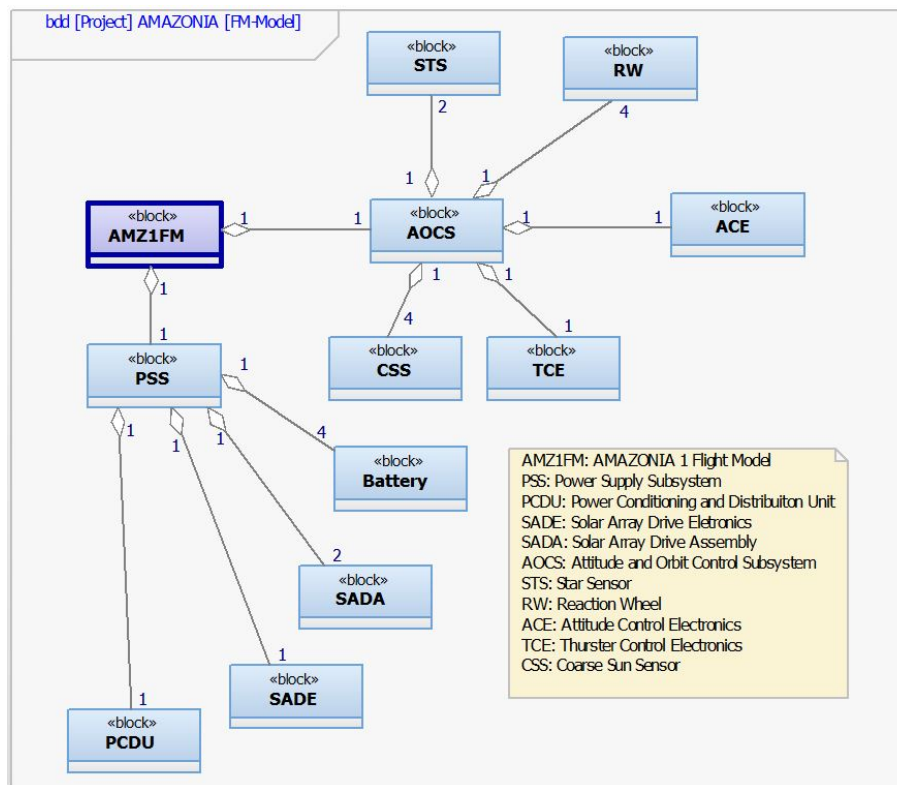
Seguindo a abordagem SCE, a partir da arquitetura funcional essencial e dos requisitos não funcionais capturados, uma arquitetura funcional alocável é derivada e os requisitos do sistema são, então, alocados para elementos físicos (como, por exemplo, equipamentos e elementos de infraestrutura), lógicos (como, por exemplo, softwares) ou organizacionais (como equipes de trabalho), resultando em uma arquitetura de implementação. Isso é feito simultaneamente para os elementos do produto e da organização da solução sistema.

4.4.1 Arquitetura de implementação do produto

Uma pequena parte do modelo de arquitetura de implementação do produto está representada neste capítulo pelos diagramas de Bloco (BDD) e de Bloco Interno (IBD) da SysML, Figuras 4.8 e 4.9, respectivamente.

No diagrama da Figura 4.8 o modelo de voo do satélite AMAZONIA 1 (AMAZONIA 1 FM) é representado como uma composição dos seus subsistemas; e seus subsistemas são representados como uma composição de seus equipamentos.

Figura 4.8 - Diagrama de Blocos SysML simplificado de parte do modelo de voo do satélite AMAZONIA 1.

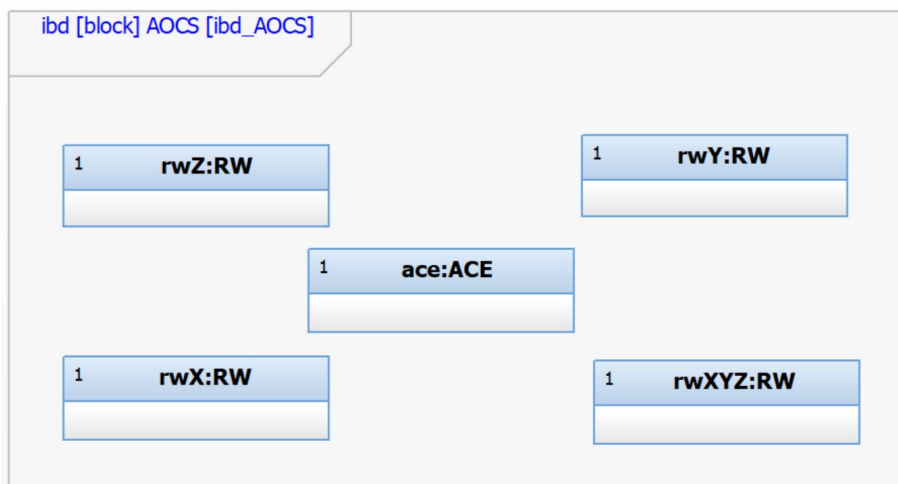


O diagrama apresenta dois subsistemas do AMAZONIA 1 FM: PSS (*Power Supply Subsystem*) e AOCS (*Attitude and Orbit Control Subsystem*); e alguns dos equipamentos que compõe esses subsistemas. O subsistema PSS implementa as funções de gerar, armazenar e distribuir energia. O subsistema AOCS implementa as funções de controlar e estabilizar atitude, de controlar e determinar a órbita do satélite e de realizar manobras orbitais.

De acordo com a notação SysML, no BDD as caixas representam os Blocos e seu primeiro compartimento é utilizado para nomeá-los. As linhas com um losango em uma extremidade conectando os Blocos representam uma Associação de Partes, que são comumente usadas para decompor um Bloco em suas Partes. Os números na outra extremidade das linhas representam a quantidade de instâncias ou usos do Bloco relacionado. Por exemplo, o subsistema PSS é decomposto em 1 PCDU, 1 SADE, 2 SADA e 4 baterias.

Já a Figura 4.9 mostra instâncias do Blocos RW e ACE em um IBD do subsistema AOCS. Segundo a notação SysML, em um IBD as caixas representam as Partes e em seu primeiro compartimento aparecem o nome da Parte e o Bloco do qual a Parte é um tipo, no formato: Nome da Parte: Nome do tipo.

Figura 4.9 - Diagrama de Blocos Internos SysML simplificado representando o subsistema AOCS do modelo de voo do satélite AMAZONIA 1.



Esses modelos expressam a implementação do sistema. Ou seja, no caso dos satélites, os subsistemas, componentes, softwares ou quaisquer elementos destinados a desempenhar as funções definidas nos modelos funcionais do produto.

Essa estrutura de implementação do produto é de interesse do SI, pois será referenciada nele para registrar diversas ações, como montagem de equipamentos e conexão e desconexão de cabos. Por exemplo: no SI poderá ser registrado que o equipamento x foi montado na posição y do subsistema z. Blocos e Partes do modelo SysML correspondem, portanto, a prováveis objetos do modelo de dados do SI.

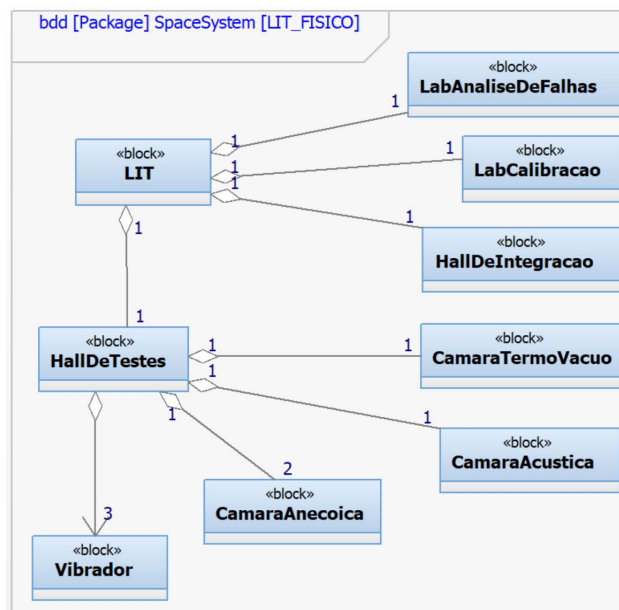
Então, embora esses modelos da Engenharia de Sistemas não tenham uma relação estrutural direta com os modelos do SI (um Bloco no modelo Sysml não corresponderá a uma Classe no modelo de dados do SI), podemos ver uma relação semântica entre eles que pode ser explorada para um trabalho conjunto.

4.4.2 Arquitetura de implementação da organização

Uma parte da arquitetura de implementação da organização (LIT) está representada neste capítulo em diagramas de Bloco da SysML.

O BDD da Figura 4.10 apresenta alguns dos elementos físicos que compõem o LIT: laboratório de análise de falhas, laboratório de calibração, hall de integração e hall de testes. Esse diagrama representa também o desdobramento do hall de testes em partes que o compõe: 1 câmara termo vácuo, 1 câmara acústica, 2 câmaras anecoicas e 3 vibradores.

Figura 4.10 - Diagrama de Blocos SysML simplificado representando parte da arquitetura de implementação do LIT (elementos físicos).

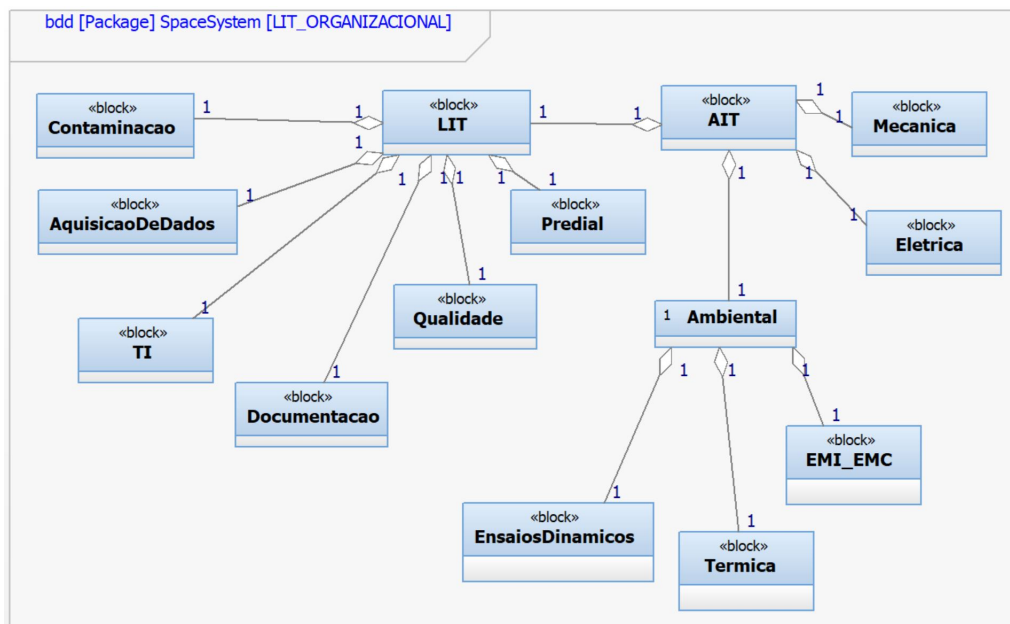


A arquitetura física da organização é de interesse do SI no sentido de que, no caso de AIT de satélite, por exemplo, os espaços físicos e as câmaras de teste são itens que poderão ter seu uso gerenciado via SI.

O diagrama da Figura 4.11 apresenta elementos organizacionais que compõem o LIT, as equipes de: controle de contaminação, aquisição de dados; tecnologia da informação, documentação, qualidade, manutenção predial e AIT.

Esse diagrama apresenta ainda o desdobramento da equipe de AIT, ou seja, suas subdivisões: equipe de mecânica, equipe de elétrica e equipe de testes ambientais. A equipe de testes ambientais se divide ainda em: equipe de ensaios dinâmicos, equipe da térmica e equipe de EMI-EMC (interferência e compatibilidade eletromagnética).

Figura 4.11 - Diagrama de Blocos SysML simplificado representando parte da arquitetura de implementação do LIT (elementos organizacionais - equipes).



Visto que funções da organização poderão contar com o SI para serem executadas, na arquitetura organizacional as equipes designadas para desempenhar as funções apontam atores de Casos de Uso e informações relevantes para definir o controle de acesso ao SI.

Esses modelos são, portanto, uma fonte de dados para o SI, assim como os modelos de implementação do produto.

5 PROPOSTA DE UM MÉTODO PARA INTEGRAR O DESENVOLVIMENTO DE SISTEMAS DE INFORMAÇÃO COM O DESENVOLVIMENTO DE SISTEMAS ESPACIAIS

Constatada a viabilidade de abordar o problema do desenvolvimento de SI de forma integrada e simultânea com o desenvolvimento do sistema espacial, conforme apresentado no Capítulo 4, este capítulo propõe um método que oferece um roteiro, um plano geral de como isso pode ser realizado, de forma que o esforço empregado no domínio da Engenharia de Sistemas seja também aproveitado no domínio da Engenharia de Software.

O termo “método” foi utilizado neste trabalho no sentido de indicar um caminho a ser percorrido para desenvolver um SI de forma integrada e simultânea com o desenvolvimento do satélite. Um caminho que pode ser repetido em diferentes casos de desenvolvimento de SI para apoiar diferentes processos do ciclo de vida de diferentes satélites. A definição que mais se aproxima do que se deseja expressar com o termo “método” neste trabalho é a de Bunge (1980): “método é um procedimento regular, explícito e passível de ser repetido para conseguir-se alguma coisa, seja material ou conceitual.”

O método aqui proposto assume o uso da SCE e da MBSE como as abordagens de engenharia de sistemas para o desenvolvimento do satélite e baseia-se em três pilares: o uso da MBSE na prática da SCE, a especialização da SCE para contemplar o desenvolvimento dos sistemas de informação e a definição de uma convenção de modelagem a ser adotada para a construção dos modelos no domínio da Engenharia de Sistemas que possibilite a integração entre os modelos dos dois domínios. Esses pilares são explicados nas Seções 5.1, 5.2 e 5.3, respectivamente, e o método é apresentado na Seção 5.4.

5.1 Uso da MBSE na prática da SCE - MBSCE

Desde sua criação em 1999 (LOUREIRO, 1999), a SCE sugere o uso de modelos para as análises por ela propostas. O *Total View Framework* é um framework de modelagem (LOUREIRO, 1999), o que significa que os elementos do *framework* são modelos (modelo de produto, modelo de organização, modelo de requisitos, modelo funcional, modelo de implementação).

Porém, apenas o uso de modelos no processo de engenharia não caracteriza a abordagem MBSE. Adotar a MBSE significa ter os modelos como a principal fonte de

informação e principal forma de troca de informações entre os envolvidos no processo de engenharia (FRIEDENTHAL et al., 2014).

Para maior proveito do esforço empreendido no processo de engenharia de sistemas, este trabalho propõe o uso da MBSE na prática da SCE, o que vamos chamar de MBSCE.

A MBSCE mostra-se adequada, pois além de somar os benefícios da MBSE e da SCE, no contexto desta pesquisa traz a vantagem da familiaridade que o engenheiro de software tem com o uso de modelos.

Na Engenharia de Software é comum o uso de modelos para descrever aspectos do sistema em desenvolvimento, prática que facilita a visualização e entendimento do sistema. Modelos são criados para especificar requisitos, estrutura e/ou comportamento de um sistema, representando-o em seus diversos aspectos (GUEDES, 2018).

A MBSCE torna o processo de engenharia de satélites mais propício para os objetivos desta pesquisa.

5.2 Especialização da SCE para contemplar o desenvolvimento dos sistemas de informação

Na ótica da SCE, a solução sistema é composta por elementos do produto e das organizações que implementam os processos do ciclo de vida desse produto. Ao desenvolver a organização pelo processo da SCE, os sistemas de informação surgem na análise de implementação como elementos que vão implementar determinadas funções identificadas pela análise funcional da SCE. Portanto, dentro desse contexto, os sistemas de informação aparecem na solução sistema como elementos da organização.

O desenvolvimento dos sistemas de informação tradicionalmente ocorre de forma independente do desenvolvimento do sistema espacial. Não consta na literatura trabalhos que abordem o desenvolvimento desses sistemas de forma integrada com o desenvolvimento dos sistemas espaciais. Esse desenvolvimento independente não aproveita o esforço já empregado para o desenvolvimento do sistema espacial.

Nesta pesquisa foi observado que tanto as atividades do desenvolvimento do produto, quanto às do desenvolvimento da organização propostas pela SCE produzem artefatos para o desenvolvimento dos sistemas de informação (Capítulo 4) sendo, portanto, viável e vantajoso conduzir o desenvolvimento desses sistemas juntamente

com o desenvolvimento do sistema espacial.

Embora os sistemas de informação sejam elementos da decomposição da organização, os resultados das análises da dimensão de análise aplicadas aos elementos das camadas superiores da dimensão de estrutura, assim como os resultados das análises aplicadas aos elementos da dimensão de estrutura do produto, são também altamente aplicáveis ao desenvolvimento dos sistemas de informação.

Visto que não só o resultado das análises aplicadas ao elemento “organização”, mas também o resultado das análises aplicadas ao elemento “produto” influenciam os sistemas de informação e que os sistemas de informação são elementos da organização há, portanto, uma interação entre o desenvolvimento do produto e da organização que não é explorada pela SCE em sua proposta original.

A proposta deste trabalho é trazer os sistemas de informação para um desenvolvimento conjunto e simultâneo com o produto e a organização (em nível macro) e tirar proveito da interação identificada entre o desenvolvimento desses dois elementos.

Embora os sistemas de informação originalmente se enquadrem em níveis mais baixos da dimensão de estrutura da SCE, por serem elementos da decomposição da organização, a proposta é desenvolvê-los simultaneamente com o sistema espacial desde os níveis mais altos da dimensão de estrutura tanto da organização, quanto do produto, fazendo uso dos artefatos gerados pelas atividades da engenharia de sistemas.

Os sistemas de informação passam a ser desenvolvidos de forma simultânea e colaborativa com o sistema espacial e não de forma independente e isolada.

Nesta proposta, as análises da dimensão de análise continuam sendo aplicadas ao produto e à organização, enquanto o desenvolvimento dos sistemas de informação faz uso do resultado dessas análises aplicadas a esses elementos. O engenheiro de software atua junto com o engenheiro de sistemas na aplicação da SCE, porém com o olhar voltado para o desenvolvimento do SI, enquanto o olhar do engenheiro de sistemas está voltado para o desenvolvimento do sistema espacial.

Os objetos das análises da dimensão de análise continuam sendo o produto e as organizações que implementam os processos do ciclo de vida desse produto, porém, dos resultados dessas análises também emergem os sistemas de informação.

Essa proposta de especialização é ilustrada na Figura 5.1 que expressa que os siste-

mas de informação passam a ser tratados como um elemento especial da organização e que os resultados das análises da dimensão de análise aplicadas ao produto e à organização se aplicam aos sistemas de informação.

A Tabela 5.1 resume a relação dos modelos da MBSCE com o desenvolvimento dos sistemas de informação e orienta a atuação do engenheiro de software.

Figura 5.1 - Proposta de especialização da abordagem SCE.

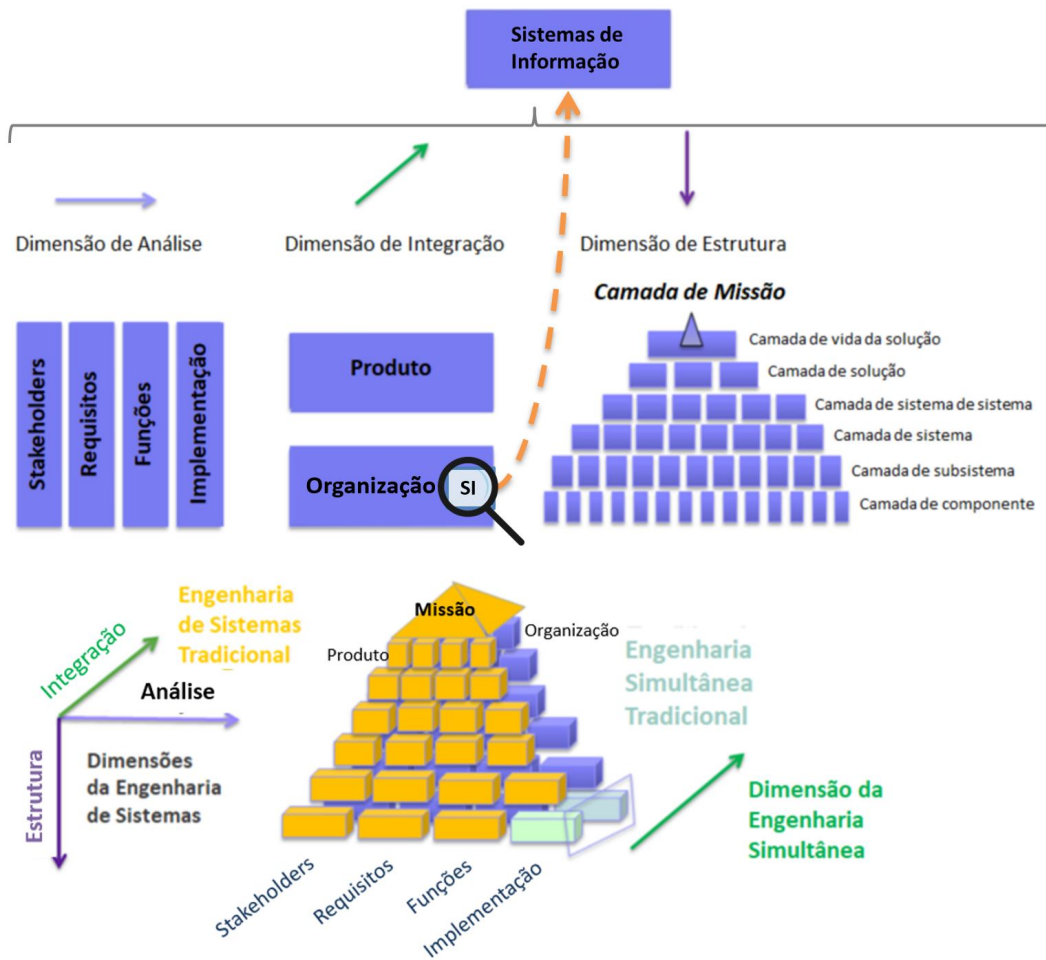


Tabela 5.1 - MBSCE X desenvolvimento de sistemas de informação.

		DIMENSÃO DE INTEGRAÇÃO	
		PRODUTO	ORGANIZAÇÃO
DIMENSÃO DE ANÁLISE	STAKEHOLDERS	<p>Os <i>concerns</i> e requisitos relacionados ao sistema espacial expressos nesses modelos formam uma base para que o engenheiro de software identifique os requisitos para o sistema de informação, já que o sistema de informação será uma ferramenta de apoio para que as necessidades dos <i>stakeholders</i> sejam atendidas.</p>	
	REQUISITOS	<p>Como o sistema de informação pode atuar para contribuir no atendimento dos requisitos dos <i>stakeholders</i> do sistema espacial? Esta é a principal questão para o engenheiro de software durante a Análise de <i>Stakeholders</i> e Análise de Requisitos.</p>	
	FUNÇÕES	<p>A arquitetura funcional do produto não influencia o sistema de informação e o sistema de informação não interfere nela.</p> <p>Os modelos funcionais do produto têm pouca ou nenhuma influência no desenvolvimento do sistema de informação.</p>	<p>Estrutura funcional (visão estática): esses modelos contêm as informações que fluem entre os processos da organização, expressam a necessidade de informação dos atores do processo. O sistema de informação deve ser moldado para que essas informações estejam disponíveis nele.</p> <p>Comportamento funcional (visão dinâmica): esses modelos expressam a forma como os processos são executados na organização no que diz respeito à sua execução temporal, lógica, causal e condicional. Apontam, portanto, regras de negócio para o sistema de informação, além de atividades a serem apoiadas.</p>
	IMPLEMENTAÇÃO	<p>Esses modelos expressam a implementação do produto. Ou seja, descrevem os itens necessários para desempenhar as funções definidas nos modelos funcionais do produto.</p> <p>No caso dos satélites, os itens físicos ou lógicos (subsistemas, componentes, softwares) destinados a desempenhar as funções definidas nos modelos funcionais do produto.</p> <p>A estrutura do produto é de interesse do sistema de informação, pois pode ser referenciada nele. No caso de AIT de satélites, por exemplo, será referenciada para registrar diversas ações, como por exemplo, o que chegou, o que foi montado, o que foi conectado etc.</p> <p>Esses modelos contribuem, portanto, para a definição da estrutura de dados do sistema de informação e também são fonte de entrada para popular a base de dados.</p>	<p>Esses modelos expressam a implementação das organizações que realizam os processos do ciclo de vida do produto. Ou seja, descrevem os itens necessários para desempenhar as funções definidas nos modelos funcionais da organização.</p> <p>Tanto a estrutura física da organização (prédios, salas, câmaras, laboratórios etc.), quanto a estrutura organizacional (departamentos, equipes etc.) são de interesse do sistema de informação.</p> <p>No caso de AIT de satélites, por exemplo, os espaços físicos e as câmaras de teste são itens que terão seu uso gerenciado no sistema de informação. Esses modelos contribuem, portanto, para a definição da estrutura de dados do sistema de informação e também são fonte de entrada para popular a base de dados.</p> <p>Quanto à estrutura organizacional (equipes designadas para desempenhar as funções, por exemplo), além de também contribuir com a estrutura e base de dados, apontam atores, papéis e informações relevantes para definir o controle de acesso ao sistema de informação.</p>

5.3 Convenção de modelagem e integração dos modelos com o sistema de informação

Assumindo a prática da MBSCE, que permite maior proveito do esforço empreendido no processo de engenharia de sistemas, o método propõe que uma convenção de modelagem seja definida para possibilitar a integração dos modelos de engenharia do sistema espacial com os modelos do SI.

Definir uma convenção de modelagem, no contexto deste trabalho, significa definir como os elementos nos modelos da Engenharia de Sistemas serão identificados de modo a sinalizar que são elementos de interesse do SI, visto que nem tudo será utilizado no domínio da Engenharia de Software.

Elementos do modelo, construídos de acordo com a convenção adotada, devem ser mapeados para os modelos do SI.

O padrão da convenção e do mapeamento vai depender da linguagem de modelagem escolhida para a engenharia do sistema espacial e da linguagem de modelagem e paradigma adotados para os modelos do SI. De forma geral, deve-se definir que o elemento x do modelo da engenharia de sistemas equivale ao elemento y do modelo do SI, como exemplificado na Tabela 5.2.

Definir uma convenção e fazer o mapeamento entre os dois domínios viabiliza a criação de software para ler os modelos da engenharia de sistemas, interpretá-los de acordo com o mapeamento e popular o banco de dados do SI.

Tabela 5.2 - Exemplo de mapeamento entre o domínio da Engenharia de Sistemas e o domínio da Engenharia de Software.

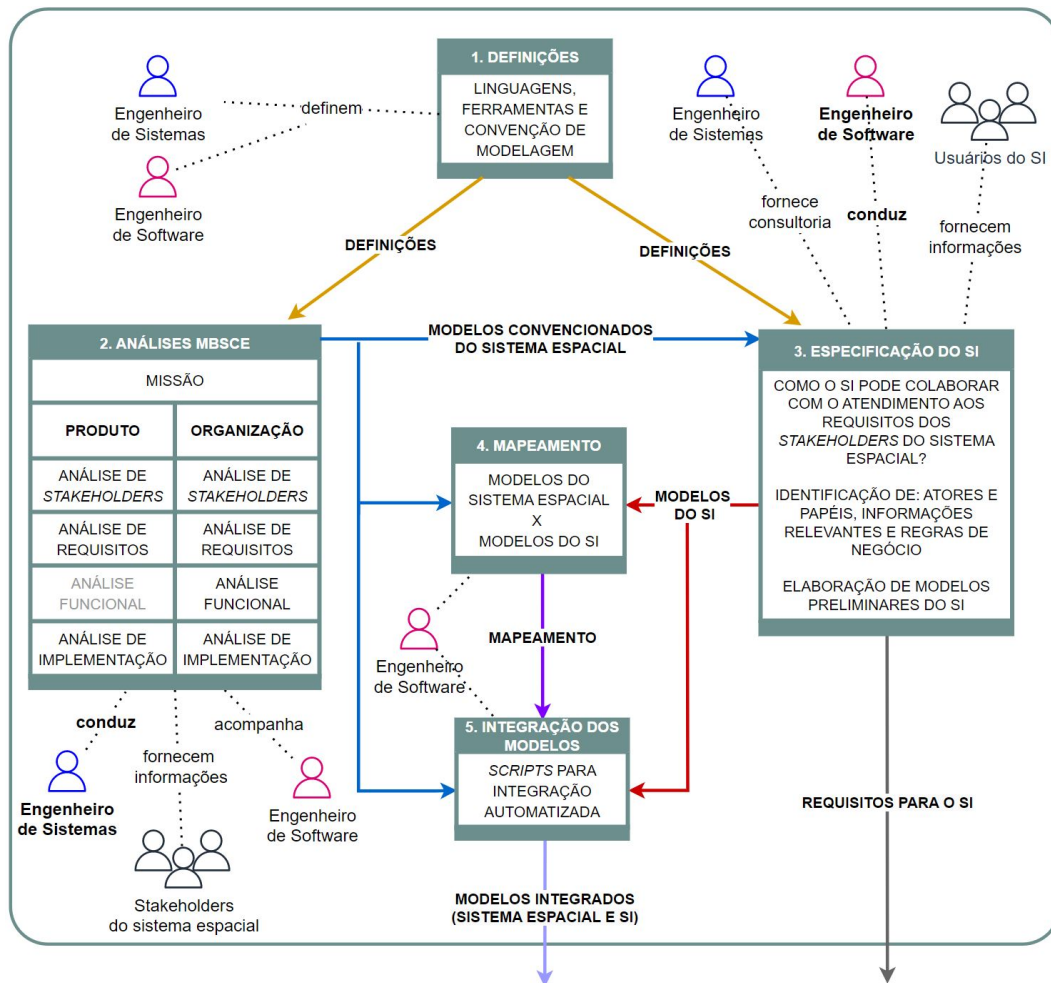
Elemento (no modelo da Engenharia de Sistemas)	Elemento (no modelo de dados do SI)	Atributo do elemento (no modelo da Engenharia de Sistemas)	Atributo do elemento (no modelo de dados do SI)
elemento 1	elemento A	atributo 1	atributo a
		atributo 2	atributo b
elemento 2	elemento B	atributo 3	atributo c
	elemento C	atributo 4	atributo d
		atributo 5	atributo e
elemento 3	elemento D	atributo 6	atributo f

5.4 Método proposto

5.4.1 Descrição do método

O método proposto é ilustrado na Figura 5.2 e explicado a seguir.

Figura 5.2 - Método proposto.



Focando nos processos do ciclo de vida do produto para o qual haja interesse de desenvolvimento de um SI, o engenheiro de software deve trabalhar juntamente com o engenheiro de sistemas como descrito a seguir:

- 1. Definições:** Engenheiro de software e engenheiro de sistemas definem lin-

guagens de modelagem, ferramentas de modelagem e uma convenção de modelagem.

2. **Análises MBSCE:** Engenheiro de sistemas e engenheiro de software percorrem a MBSCE (o primeiro visando o desenvolvimento do sistema espacial e o segundo visando o desenvolvimento do SI).

- Engenheiro de sistemas conduz as atividades da MBSCE e gera artefatos em modelos convencionados.
- Engenheiro de software acompanha as atividades podendo interagir com os *stakeholders* do sistema espacial visando a elicitação de requisitos para o SI
- Os *stakeholders* do sistema espacial comunicam suas necessidades e requisitos quanto ao sistema espacial e fornecem requisitos para o SI indiretamente através dos modelos do sistema espacial ou diretamente em interações com o engenheiro de software.

3. **Especificação do SI:** Engenheiro de software define requisitos e gera modelos para o SI baseado nos artefatos gerados pelo engenheiro de sistemas e nas interações com os *stakeholders* do sistema espacial.

- Nessa atividade o engenheiro de software, baseado nos artefatos gerados pelo engenheiro de sistemas, trabalha para identificar como o SI pode colaborar com o atendimento aos requisitos dos *stakeholders* do sistema espacial.
- Atores e papéis para o SI, informações a serem gerenciadas pelo SI e regras de negócio podem ser identificadas nos modelos resultantes das análises da MBSCE.
- A interação do engenheiro de software com os *stakeholders* do sistema espacial (potenciais usuários do SI) contribui para o esclarecimento das necessidades e expectativas quanto o SI.
- De modo geral, o papel do engenheiro de sistemas nesse passo é de um consultor, que estará disponível para sanar dúvidas do engenheiro de software sobre o processo do ciclo de vida do satélite e sobre os requisitos do sistema espacial. Porém, dependendo do processo do ciclo de vida a ser apoiado pelo SI, ele pode também se enquadrar com um usuário do SI.

- Como resultado dessa atividade, tem-se um conjunto de requisitos preliminares para o SI que formam uma base consistente para seguir com o desenvolvimento do software. Também resultam modelos de dados do SI em nível suficiente de detalhe para integração com os modelos do sistema espacial.
4. **Mapeamento:** É definido um mapeamento entre os modelos da engenharia de sistemas e os modelos da engenharia de software (o que viabiliza a integração dos modelos dos dois domínios).
 5. **Integração dos modelos:** Modelos da engenharia de software e da engenharia de sistemas são integrados via software, segundo mapeamento definido.

Na saída do método temos uma base de requisitos consistente para dar prosseguimento ao desenvolvimento do SI, além de modelos preliminares do SI integrados com os modelos do sistema espacial que poderão ser evoluídos e refinados ao decorrer do desenvolvimento do SI.

5.4.2 Requisitos para utilização do método

Abordagem de engenharia de sistemas

O método proposto tem como requisito a utilização da MBSCE (MBSE guiada pela SCE) como abordagem de engenharia do satélite.

O processo proposto pela SCE é requerido por antecipar os requisitos dos processos do ciclo de vida do produto para o início do projeto, o que viabiliza o desenvolvimento simultâneo do SI, e porque os artefatos gerados por essa abordagem mais abrangente de engenharia de sistemas, que inclui também a organização na solução sistema, fornecem entradas mais adequadas para as atividades de desenvolvimento do SI.

O uso da engenharia de sistemas baseada em modelos (MBSE) é requerido porque o uso de modelos viabiliza a integração automatizada entre os domínios da Engenharia de Software e da Engenharia de Sistemas.

Processo de desenvolvimento de software

Conforme apontado por [Sommerville \(2018\)](#), uma fase de engenharia de requisitos claramente identificável antes de começar a implementação é aplicável na maioria dos sistemas grandes de software.

Sendo assim, o método proposto é aplicável independentemente do processo de engenharia de software a ser adotado (cascata, incremental ou integração e configuração) para o desenvolvimento do SI.

As saídas do método proposto constituem uma base inicial consistente, tanto quanto aos requisitos, quanto ao projeto do SI, sobre a qual o engenheiro de software poderá trabalhar para seguir com o desenvolvimento do SI até sua implementação, independentemente do processo de engenharia de software a ser adotado. Ao longo do desenvolvimento do software, os requisitos serão detalhados e refinados de acordo com o processo de desenvolvimento de software escolhido.

Um processo evolutivo, no entanto, mostra-se interessante, pois a medida que o desenvolvimento do sistema espacial avança, novos requisitos e detalhamento de requisitos surgem para o SI. Pequenas entregas favorecem validação e ajustes precoces no SI.

Linguagens e ferramentas de modelagem

As linguagens de modelagem utilizadas não precisam ser necessariamente a SysML e a UML, utilizadas neste trabalho para demonstração da viabilidade e exemplo de aplicação do método. O fato é que os modelos do sistema espacial influenciam o projeto do SI pelo que significam e não pela linguagem em que foram criados. Independentemente da linguagem utilizada para modelar o sistema espacial, os modelos são relevantes para o engenheiro de software que pode extrair deles requisitos para o SI.

Já para a integração automatizada dos modelos dos dois domínios, dois requisitos precisam ser atendidos.

Um requisito é que a linguagem escolhida para modelar o sistema espacial seja passível de ser utilizada de forma que os modelos sigam uma convenção pré-definida pelo engenheiro de software e pelo engenheiro de sistemas.

O outro requisito é que a ferramenta de modelagem permita a leitura dos modelos, para que eles possam ser interpretados conforme a convenção e o mapeamento definido pelo engenheiro de software e sejam, então, refletidos nos modelos do SI.

5.4.3 Aplicabilidade do método

O método proposto, para ser executado de forma integral, tem como premissa que o desenvolvimento do SI ocorrerá simultaneamente com o desenvolvimento do sistema espacial, possibilitando a interação do engenheiro de software durante todo o processo da MBSCE, além de dar conhecimento ao engenheiro de sistemas do SI que atenderá determinado processo do ciclo de vida do produto, o que pode colaborar com o atendimento aos requisitos dos *stakeholders* do sistema espacial. Essa forma de trabalho possibilita, portanto, a colaboração mútua entre os dois domínios.

Dado que esta proposta assume o uso da SCE e que as atividades da SCE acontecem nas fases iniciais do desenvolvimento dos sistemas espaciais, fases 0 a C de acordo com a nomenclatura da ESA, o método é aplicável para desenvolvimento simultâneo de SI para ser utilizado nos processos do ciclo de vida do satélite que acontecem dentro das fases posteriores, pois haverá tempo hábil para colocar o software em operação quando esses processos estiverem de fato acontecendo.

Para os processos de ciclo de vida que ocorrem dentro das fases 0 a C, embora o desenvolvimento de um novo SI aplicável ao próprio satélite em desenvolvimento fique inviável pelo fato dos processos já estarem em curso, o método pode ser utilizado para desenvolver SI para futuras missões similares e para evoluir SI já existente para que atenda necessidades específicas de cada missão.

A seguir são comentados possíveis cenários nos quais o método proposto poderá ser usado, total ou parcialmente.

A organização não tem SI para determinado processo do ciclo de vida do satélite em desenvolvimento e opta por desenvolver um SI sob medida

Esse é o cenário ideal para a utilização do método. Nesse cenário o método pode ser utilizado em sua totalidade.

A organização já tem um SI para determinado processo do ciclo de vida do satélite em desenvolvimento e deseja evoluí-lo para atender nova missão

Um novo satélite ao passar pelos mesmos processos de ciclo de vida pode ter requisitos diferentes, ainda que, de forma geral, o processo do ciclo de vida seja o mesmo. Outra missão, outros *stakeholders* e outras necessidades com relação ao sistema espacial também geram outras necessidades a serem atendidas pelo SI.

Sendo assim, ainda que a organização já tenha um SI que serviu missões anteriores, a participação do engenheiro de software no desenvolvimento de cada nova missão possibilita que ele identifique as necessidades específicas dessa missão para evoluir o SI já existente de acordo com os novos requisitos. Nesse caso, o método pode ser utilizado para entender os requisitos da nova missão e evoluir o SI existente para atender a esses novos requisitos.

Nesse cenário, na atividade 1 (Definições), já haverá definições quanto ao SI que não poderão ser negociadas. Já os modelos do sistema espacial deverão ser construídos de acordo com convenção acordada entre o engenheiro de sistemas e o engenheiro de software, como propõe o método, e as informações desses modelos deverão ser mapeadas para o SI normalmente.

A organização não tem SI para determinado processo do ciclo de vida do satélite em desenvolvimento e opta por comprar um SI

Nesse cenário, o método pode ser utilizado e os requisitos resultantes da sua aplicação serão a base para especificar um SI para compra. Porém, nesse caso, os modelos do SI não estão sob controle do engenheiro de software que atua no método e, por isso, as atividades 4 (mapeamento) e 5 (integração dos modelos) ficam prejudicadas.

Dependendo do SI que for comprado, pode ser que seja possível alguma integração entre os modelos, embora os modelos do SI não tenham sido construídos de forma customizada para a necessidade da missão. Um mapeamento posterior à escolha do software a ser comprado pode ser possível, dependendo do produto escolhido. Para isso, no entanto, pode ser necessário que alterações sejam feitas nos modelos do satélite para possibilitar o mapeamento e integração.

O sistema espacial já passou pela fase de desenvolvimento

A proposta do método é que o SI seja desenvolvido simultaneamente com o sistema espacial, mas a autora considera que se o desenvolvimento de um sistema espacial já foi realizado de acordo com a MBSCE, é possível também que o engenheiro de software trabalhe com os artefatos gerados pelo engenheiro de sistemas para desenvolver um SI ou especificar um SI para compra.

Nesse caso, porém, a integração dos modelos fica prejudicada, já que os modelos do sistema espacial não estariam de acordo com uma convenção pré-definida. Alterações nos modelos do sistema espacial podem ser consideradas para viabilizar a integração com os modelos do SI em desenvolvimento.

Deve-se considerar também que, nesse cenário, ainda perdem-se os benefícios da interação entre os dois domínios durante o processo de engenharia do satélite.

6 EXEMPLO DE APLICAÇÃO E AVALIAÇÃO DO MÉTODO

Na primeira seção deste capítulo vamos partir do exemplo de aplicação da SCE, assumindo a prática da MBSCE, para a engenharia do satélite AMAZONIA 1, apresentado no Capítulo 4, e demonstrar a aplicação do método proposto no desenvolvimento de um sistema de informação (SI) para apoio ao processo de AIT desse satélite.

Na segunda seção será apresentada a avaliação do método, realizada por profissionais da área de desenvolvimento de sistemas de informação.

6.1 Exemplo de aplicação do método

Como explicado no Capítulo 5, a aplicação do método consiste em que o engenheiro de software percorra a MBSCE juntamente com o engenheiro de sistemas, observando os artefatos gerados pela engenharia de sistemas com o objetivo de especificar o SI que dará suporte ao processo de ciclo de vida de interesse. Além disso, para possibilitar a integração entre os modelos do domínio da engenharia de sistemas e os modelos do domínio da engenharia de software, uma convenção de modelagem deve ser acordada entre o engenheiro de sistemas e o engenheiro de software e um mapeamento entre os elementos dos modelos deve ser definido.

6.1.1 Definições

A linguagem de modelagem utilizada para a engenharia do satélite de exemplo é a SysML, como visto no Capítulo 4. Para o desenvolvimento do SI foi adotado o paradigma da orientação a objetos e a linguagem UML para modelagem. A ferramenta de modelagem utilizada é a Rhapsody (IBM, 2023b). A convenção de modelagem definida será explicada na Seção 6.1.4.

6.1.2 Análises MBSCE

Conforme apresentado no Capítulo 4.

6.1.3 Especificação do sistema de informação a partir dos artefatos da MBSCE

Nas Tabelas 6.1, 6.2, 6.3 e 6.4, temos na linha 1 requisitos dos *stakeholders* do sistema espacial (apresentados nas Figuras 4.4 e 4.5), na linha 2 as observações do engenheiro de software sobre esses requisitos e na linha 3 suas influências na especificação do SI. Observando os requisitos dos *stakeholders* do sistema espacial, o engenheiro de

software identifica Casos de Uso e Classes para o SI.

Tabela 6.1 - Influências dos requisitos dos *stakeholders* no SI - 1.

1	RSTK001 - Equipe de ensaios dinâmicos deve ser capaz de fixar o satélite no vibrador.
2	Procedimento para a fixação do satélite sobre a mesa vibratória deverá estar disponível no SI. Vídeos demonstrativos devem estar disponíveis no SI.
3	Casos de Uso: Cadastrar procedimentos, Visualizar procedimentos. Classes: Task (para cadastro de tarefas, como a de fixar o satélite na mesa vibratória), Video (para cadastro dos vídeos referentes aos procedimentos)

Tabela 6.2 - Influências dos requisitos dos *stakeholders* no SI - 2.

1	RSTK002 - Equipe de ensaios dinâmicos deve ser capaz de instalar acelerômetros nos pontos definidos para as medidas dos níveis de vibração. RSTK003 - Equipe de ensaios dinâmicos deve ser capaz de remover os acelerômetros instalados no satélite após os ensaios para reuso.
2	Funcionalidades para controle da instalação e remoção de acelerômetros devem estar disponíveis no SI. Visualização 3D da estrutura do satélite destacando os pontos de instalação dos acelerômetros e quais pontos estão com acelerômetros instalados deverá ser provida pelo SI.
3	Casos de Uso: Cadastrar estrutura do satélite, Cadastrar acelerômetros, Registrar instalação e remoção de acelerômetros, Visualizar satélite em 3D. Classes para cadastro da estrutura do satélite: Satellite, SatelliteModel, Subsystem, SatelliteEquipment, EquipmentType. Classes para controle de instalação de acelerômetros: Accelerometer, AccelerometerInstallation.

Tabela 6.3 - Influências dos requisitos dos *stakeholders* no SI - 3.

1	RSTK004 - Gerente de AIT deve ser capaz de saber que atividades foram ou estão sendo realizadas em determinada data.
	RSTK005 - Gerente de AIT deve ser capaz de saber os resultados das atividades realizadas.
	RSTK008 - Equipe de Desenvolvimento deve ser capaz de saber os resultados das atividades realizadas.
2	Funcionalidades para o registro da execução das atividades e seus resultados devem estar disponíveis no SI.
	Funcionalidade para visualização dos registros das atividades pelo gerente de AIT deve estar disponível no SI.
3	Casos de Uso: Planejar atividades de AIT, Registrar execução de atividades, Consultar atividades, Acompanhar plano de AIT (pode oferecer visualização gráfica do planejamento e andamento das atividades).
	Classes: Task (para cadastro das tarefas), PlannedTask (para cadastro do planejamento das tarefas), ExecutedTask (para cadastro dos dados de execução das tarefas)

Tabela 6.4 - Influências dos requisitos dos *stakeholders* no SI - 4.

1	RSTK006 - Gerente de AIT deve ser capaz de disponibilizar para sua equipe câmaras, equipamentos e salas do LIT para realizar as atividades de AIT dentro do cronograma previsto.
	RSTK007 - Equipe de atendimento a clientes do LIT deve ser capaz de utilizar câmaras, equipamentos e salas do LIT para atendimento a clientes.
2	Funcionalidades para agendamento da utilização da infraestrutura e compartilhamento da agenda entre a equipe de AIT e as equipes que atendem clientes do LIT devem estar disponíveis no SI.
3	Casos de Uso: Cadastrar itens da infraestrutura, Agendar uso da infraestrutura.
	Classes: InfrastructureItem (para cadastro dos itens da infraestrutura), InfrastructureSchedule (para cadastro dos dados de agendamento do uso dos itens da infraestrutura).

Nas Tabelas 6.5, 6.6 e 6.7, temos na linha 1 elementos do modelo de estrutura funcional da organização (ver Figura 4.6), na linha 2 as observações do engenheiro de software sobre esses elementos do modelo e na linha 3 suas influências na especificação do SI. Observando os modelos de estrutura funcional da organização, o engenheiro de software identifica Casos de Uso e Classes para o SI.

Tabela 6.5 - Influências dos modelos de estrutura funcional da organização no SI - 1.

1	Fluxo “log do satélite”
2	Informações a serem disponibilizadas pelo SI: equipamentos montados no satélite (<i>assembly</i>) numa determinada data, quantidade de conexões (<i>mates</i>) de um dado conector, histórico de <i>assembly/disassembly</i> , histórico de <i>mate/demate</i> .
3	Casos de Uso: Cadastrar equipamento para AIT, Registrar <i>assembly</i> e <i>disassembly</i> , Registrar <i>mate</i> e <i>demate</i> , Cadastrar conectores. Classes para cadastro de equipamentos: SatelliteEquipment, ReceivedEquipment Classes para registro de montagem e desmontagem de equipamentos: Assembly, Disassembly Classes para registro de conexão e desconexão de conectores: Connector, Mate, Demate

Tabela 6.6 - Influências dos modelos de estrutura funcional da organização no SI - 2.

1	Fluxo “intercorrências”
2	Ocorrências de não conformidades na macro função “Executar AIT” devem estar disponíveis no SI para a macro função “Gerenciar AIT”.
3	Casos de Uso: Cadastrar NCR (relatório de Não Conformidade), Consultar NCR Classes: Ncr (para cadastro dos relatórios de não conformidade)

Tabela 6.7 - Influências dos modelos de estrutura funcional da organização no SI - 3.

1	Fluxo “recursos disponíveis”
2	Os recursos para realização de AIT deverão estar cadastrados no SI e deve ser possível fazer a alocação e liberação desses recursos. Esses recursos incluem ferramentas, materiais, GSEs, equipamentos e pessoal.
3	Casos de Uso: Cadastrar ferramentas, Cadastrar materiais, Cadastrar GSEs, Cadastrar equipamentos para AIT, Cadastrar equipe. Classes para cadastro dos recursos: Tool, Material, Gse, ReceivedEquipment, Professional

Nas Tabelas 6.8, 6.9, 6.10 e 6.11, temos na linha 1 elementos do modelo de comportamento funcional da organização (ver Figura 4.7), na linha 2 as observações do

engenheiro de software sobre esses elementos do modelo e na linha 3 suas influências na especificação do SI. Observando os modelos de comportamento funcional da organização, o engenheiro de software identifica Casos de Uso, Classes e regras de negócio para o SI.

Tabela 6.8 - Influências dos modelos de comportamento funcional da organização no SI - 1.

1	Atividade “Receber Equipamento”
2	Funcionalidade para registrar o recebimento de equipamentos que chegam para AIT deve estar disponível no SI.
3	Casos de Uso: Registrar recebimento de equipamento para AIT Classes: ReceivedEquipment

Tabela 6.9 - Influências dos modelos de comportamento funcional da organização no SI - 2.

1	Atividade “Realizar Inspeção”
2	Funcionalidade para registrar a inspeção realizada nos equipamentos deve estar disponível no SI.
3	Casos de Uso: Registrar Inspeção de equipamento Classes para registro da inspeção: InspectionSheet, InspectionSheetItem

Tabela 6.10 - Influências dos modelos de comportamento funcional da organização no SI
- 3.

1	Bifurcação “Equipamento ok?”
2	Regras de negócio: a: O equipamento não pode ser montado no satélite se não tiver sido aprovado na inspeção. b: Deverá ser aberto relatório de não conformidade quando equipamento não passar na inspeção.
3	Implementação da regra de negócio “a” no SI: O sistema deve alertar se o equipamento informado durante o registro de atividade não foi aprovado em inspeção. Implementação da regra de negócio “b” no SI: Quando o usuário registrar que o equipamento não passou na inspeção, o sistema deve perguntar se o usuário deseja abrir um relatório de não conformidade e gerar um, caso a resposta seja sim.

Tabela 6.11 - Influências dos modelos de comportamento funcional da organização no SI
- 4.

1	Atividade “Abrir Relatório de Não Conformidade”
2	Funcionalidade para registrar a ocorrência de não conformidades deve estar disponível no SI.
3	Casos de Uso: Cadastrar NCR Classes: Ncr

A Figura 6.1 mostra um diagrama com os Casos de Uso e as Figuras 6.2, 6.3 e 6.4 mostram diagramas de classes emergentes da análise do engenheiro de software durante a aplicação do método.

Figura 6.1 - Diagrama de Casos de Uso emergente da aplicação do método proposto.

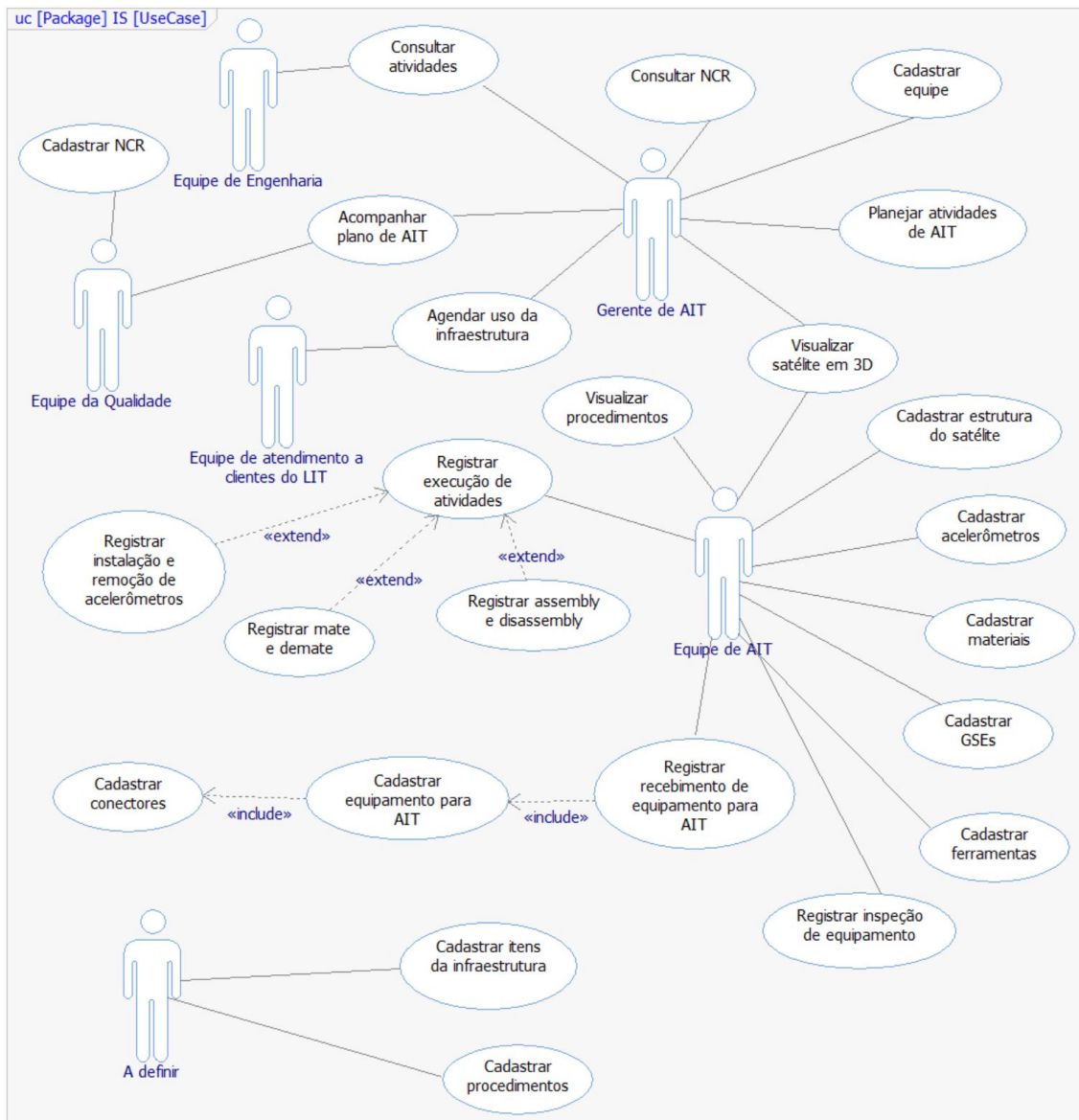


Figura 6.2 - Diagrama de Classes emergente da aplicação do método proposto - 1.

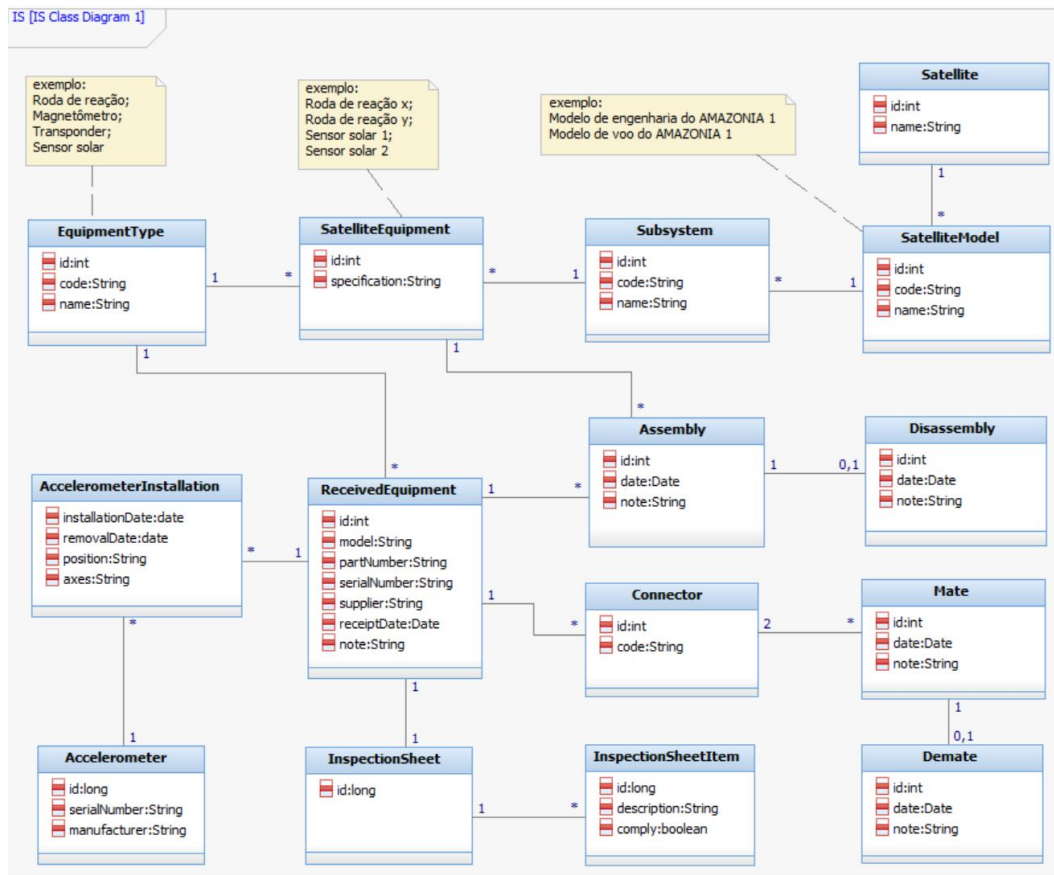


Figura 6.3 - Diagrama de Classes emergente da aplicação do método proposto - 2.

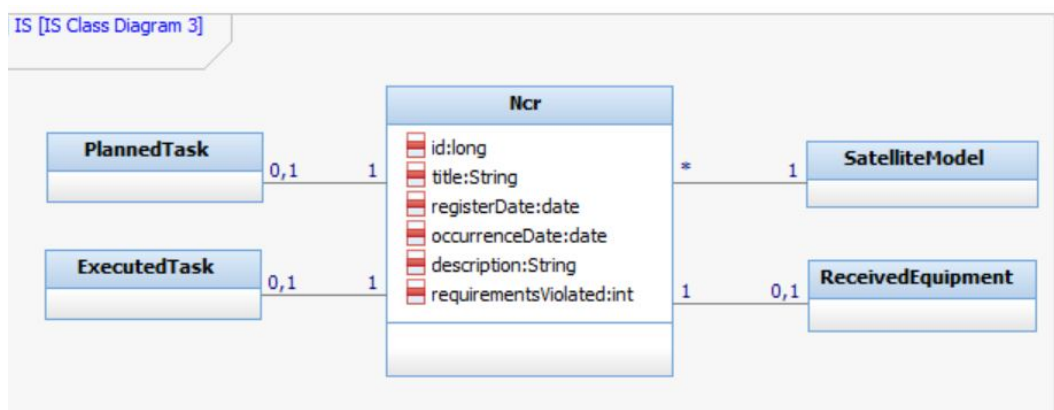
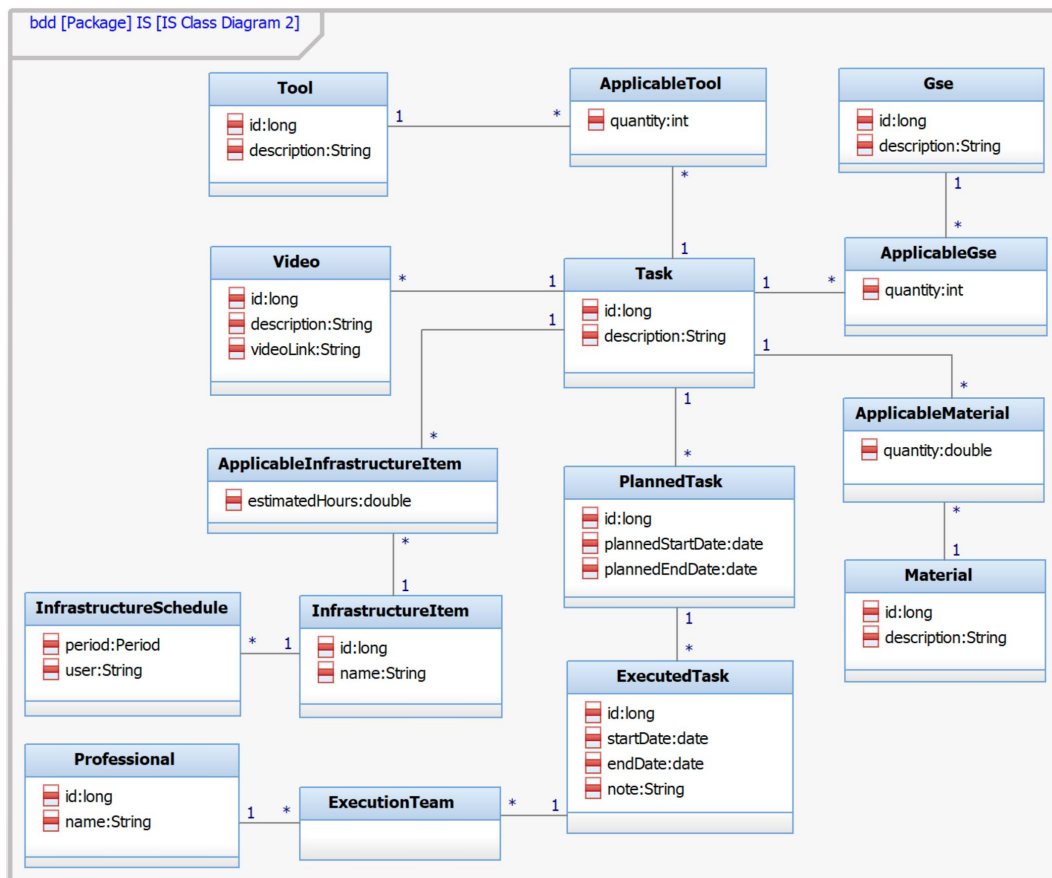


Figura 6.4 - Diagrama de Classes emergente da aplicação do método proposto - 3.



6.1.4 Mapeamento e integração dos modelos dos dois domínios

Olhando para os diagramas das Figuras 4.8, 4.9 e 6.2, vemos que a estrutura do satélite é modelada nos diagramas estruturais (IBD e BDD) no domínio da Engenharia de Sistemas e também no Diagrama de Classes no domínio da Engenharia de Software, mas de maneiras diferentes.

Podemos observar que os elementos do BDD e IBD não correspondem diretamente aos elementos estruturais do Diagrama de Classes, mas que existe um relacionamento entre os modelos. Elementos do BDD e IBD correspondem a instâncias das Classes do SI.

O Bloco AMZ1FM no BDD da Figura 4.8 corresponde a uma instância da classe SatelliteModel. Os Blocos dos subsistemas (AOCS e PSS) correspondem a instâncias

da Classe Subsystem.

Os Blocos que aparecem na decomposição dos subsistemas no BDD da Figura 4.8 correspondem a instâncias da classe EquipmentType. Essa Classe representa os tipos de equipamento que compõe o satélite, como uma roda de reação ou um sensor solar.

Já a Classe SatelliteEquipment representa um tipo de equipamento numa posição específica do satélite, como uma roda de reação na posição x ou y. Instâncias da Classe SatelliteEquipment não aparecem no BDD, mas aparecem no IBD dos Blocos de subsistema, como na Figura 4.9.

Embora esses modelos da Engenharia de Sistemas e o modelo de Classes da Engenharia de Software não tenham relação direta, podemos definir uma estratégia de integração. Como apresentado no Capítulo 5, essa estratégia inclui a definição de uma convenção de modelagem para a construção dos modelos do sistema espacial que possibilite a identificação dos elementos de interesse do SI.

No início desta pesquisa, a autora publicou um trabalho (SILVA; LOUREIRO, 2018) utilizando supertipos para identificar as Classes do SI nas quais os elementos dos modelos da Engenharia de Sistemas deveriam ser alocados como instâncias, porém essa estratégia mostrou-se limitada.

O uso de estereótipos da SysML para implementar esse *link* entre os modelos provou ser mais apropriado. Os estereótipos permitem classificar elementos com “algo em comum”. Com este recurso, podemos então classificar os elementos dos modelos SysML, identificando a qual Classe do SI eles estão relacionados.

O engenheiro de software aponta para o engenheiro de sistemas, que é quem faz os modelos SysML, quais elementos são de interesse do SI e com quais estereótipos eles devem ser identificados.

A convenção adotada para este exemplo de aplicação foi classificar os elementos da SysML com estereótipos acordados entre o engenheiro de sistemas e o engenheiro de software.

Definiu-se que os estereótipos seriam iniciados com o prefixo “is_” (uma referência a *Information System*) para indicar que os elementos com eles classificados são de interesse do SI .

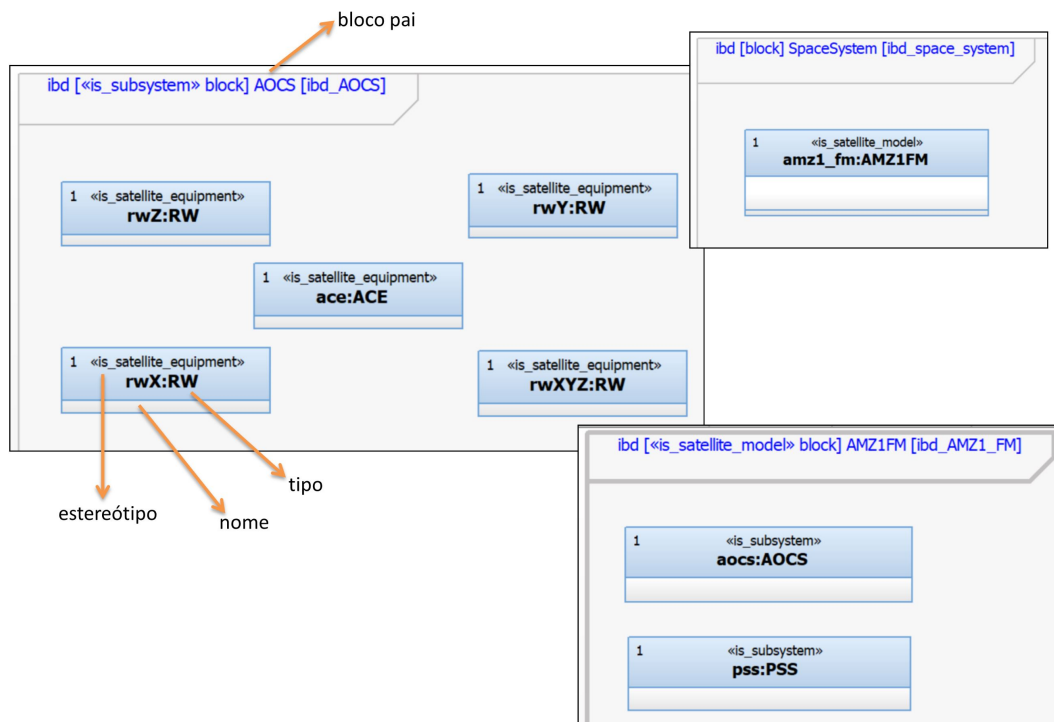
A Figura 6.5 mostra IBDs com Partes classificadas com estereótipos, para exempli-

ficar. Os seguintes estereótipos foram utilizados:

- «is_satellite_equipment»
- «is_satellite_model»
- «is_subsystem»

As propriedades dos elementos também são apontadas nessa figura porque são referenciadas na Tabela 6.12.

Figura 6.5 - IBD com Partes classificadas com estereótipos.



Definida a convenção, é necessário agora mapear o destino de cada informação do modelo SysML que interessa ao SI, conforme exemplificado na Tabela 6.12. Nessa tabela, é feito um mapeamento para leitura de IBDs dos modelos SysML para popular Classes do SI. Na coluna “Atributo da Parte”, os termos entre parênteses correspondem ao nome dos atributos na ferramenta Rhapsody (IBM, 2023b), que foi utilizada para criar os modelos SysML.

A ferramenta Rhapsody possui uma biblioteca, chamada JAVA API (IBM, 2009), que permite ler e escrever nos modelos. Utilizando essa biblioteca, foi criado um script na linguagem groovy (APACHE, 2023) para ler os modelos SysML, interpretá-los de acordo com o mapeamento definido e gerar entradas para o SI a partir dos modelos de engenharia.

Tabela 6.12 - Mapeamento entre os dois domínios.

Estereótipo (no modelo SysML)	Classe (no modelo do SI)	Atributo da Parte (no modelo SysML)	Atributo da Classe (no modelo do SI)
is_subsystem	Subsystem	nome (name)	code
		bloco pai (ofClass.name)	satelliteModel
is_satellite_equipment	EquipmentType	tipo (other- Class.name)	code
	SatelliteEquipment	nome (name)	specification
		tipo (other- Class.name)	equipmentType
		bloco pai (ofClass.name)	subsystem
is_satellite_model	SatelliteModel	nome (name)	code

A Figura 6.6 apresenta o código escrito em groovy que faz uso da biblioteca JAVA API para a leitura dos modelos do Rhapsody. A Figura 6.7 apresenta o código groovy que cria objetos, a partir dos elementos do Rhapsody, para popular o banco de dados do SI.

Figura 6.6 - Código groovy para ler os modelos do Rhapsody.

```
/*Classe que lê o projeto do Rhapsody e armazena os elementos de interesse do SI */
class ReadRhapsody {

    private IRPProject prj //projeto do Rhapsody

    /*Listas para armazenar os elementos de interesse do SI */
    List satelliteEquipments = []
    List subsystems = []
    List satelliteModels = []

    ReadRhapsody(){
        /*Pega o projeto ativo do Rhapsody.
        * Precisa usar o java 32-bit para rodar, porque a biblioteca rhapsody.dll é 32-bits*/
        prj = RhapsodyAppServer.activeRhapsodyApplication.activeProject()
        initialize()
    }

    /* Lê os modelos ativos no Rhapsody e popula as listas com os elementos de interesse do SI */
    private void initialize(){
        //Pega todos os elementos do projeto ativo
        List allElements = prj.nestedElementsRecursive.toList()

        allElements.each { element ->
            //armazena todos os esteriótipos do elemento
            List stereotypes = element.stereotypes.toList()*name
            /*verifica se o elemento está classificado com esteriótipo existente no mapeamento
            acordado entre o engenheiro de software e o engenheiro de sistemas (Tabela 6.12 da tese)
            e o inclui na respectiva lista
            */
            if ("is_subsystem" in stereotypes)
                subsystems << element
            else if ("is_satellite_equipment" in stereotypes)
                satelliteEquipments << element
            else if ("is_satellite_model")
                satelliteModels << element
        }
    }
}
```


Figura 6.7 - Código groovy para criar objetos para o SI a partir dos elementos dos modelos do Rhapsody.

```
/* Classe que cria objetos para popular o banco de dados do SI
 * a partir dos elementos lidos do modelo Rhapsody */
class InformationSystem {
    List<SatelliteModel>    satelliteModels    = []
    List<Subsystem>        subsystems        = []
    List<EquipmentType>    equipmentTypes    = []
    List<SatelliteEquipment> satelliteEquipments = []

    public InformationSystem(){
        println "Reading Rhapsody Model ..."
        ReadRhapsody rhapsody = new ReadRhapsody()

        //popula subsystems
        rhapsody.subsystems.each{subsystem ->
            //cria o satelliteModel que contém o subsystem
            SatelliteModel satelliteModel = new SatelliteModel(code:subsystem.ofClassName)
            //verifica se o satelliteModel já está na lista de satelliteModels, se não adiciona
            if (! satelliteModels.find{ SatelliteModel model-> model.code == subsystem.ofClassName })
                satelliteModels << satelliteModel
            //cria o subsystem e o adiciona na lista de subsystems
            subsystems << new Subsystem(code: subsystem.name, satelliteModel: satelliteModel)
        }

        //popula satelliteEquipment
        rhapsody.satelliteEquipments.each { satelliteEquipment ->
            //cria o equipmentType do satelliteEquipment
            EquipmentType equipmentType = new EquipmentType(code: satelliteEquipment.otherClassName)
            //verifica se o equipmentType já está na lista equipmentTypes, se não, adiciona
            if (! equipmentTypes.find{ EquipmentType it-> it.code == satelliteEquipment.otherClassName})
                equipmentTypes << equipmentType
            //obtem o subsystem do satelliteEquipment
            Subsystem subsystem = subsystems.find{ Subsystem it-> it.code == satelliteEquipment.ofClassName }
            //cria o satelliteEquipment e o adiciona na lista de satelliteEquipments
            satelliteEquipments << new SatelliteEquipment(specification: satelliteEquipment.name,
                type:equipmentType , subsystem: subsystem)
        }
    }
}
```

6.2 Avaliação do método

O método foi submetido à avaliação de profissionais da área de desenvolvimento de sistemas de informação, com o intuito de conhecer a percepção deles sobre a proposta deste trabalho de doutorado.

A avaliação foi feita por meio de um questionário. Os possíveis respondentes foram consultados sobre a disponibilidade para contribuir com a pesquisa. Aos que se voluntariaram a responder, foi enviado um link (SILVA, 2023) com um vídeo, no

qual a autora explica o método proposto, e o questionário elaborado pela autora.

6.2.1 Elaboração do questionário

O objetivo geral do questionário é colher a opinião de profissionais com alguma experiência no desenvolvimento de sistemas de informação sobre o método proposto neste trabalho. Visto que o método afeta principalmente a atividade de especificação do SI, o questionário foi direcionado para profissionais com alguma experiência nessa atividade. O público-alvo do questionário foi, portanto, profissionais com alguma experiência em especificação de sistemas de informação. Foram convidados a responder o questionário profissionais com ou sem experiência na área espacial.

De acordo com (GIL, 2022), construir um questionário consiste em traduzir os objetivos da pesquisa em questões específicas.

A seguir são apresentados os objetivos específicos do questionário aplicado e as questões deles originadas. Para as respostas das questões de 1 a 4 a autora optou por utilizar uma escala variando de 0 (zero) a 4, onde 0 (zero) significa “nada” e 4 significa “muito”. Assim, os valores de 1 a 3 oferecem aos respondentes opções intermediárias.

- Objetivo específico 1: Entender a percepção dos respondentes sobre a relevância do engenheiro de software estar inserido no processo de engenharia de sistemas (relevância do trabalho simultâneo e colaborativo proposto pelo método).
 - Questão 1: Você acha que a participação do engenheiro de software no processo de engenharia do sistema espacial afeta positivamente o sistema de informação em desenvolvimento em qual escala?
- Objetivo específico 2: Entender a percepção dos respondentes sobre a contribuição dos modelos da MBSCE, principalmente no caso do desenvolvimento do SI não ocorrer simultaneamente ao desenvolvimento do sistema espacial. A autora considera que se o desenvolvimento de um sistema espacial já foi realizado de acordo com a MBSCE, é possível também que o engenheiro de software trabalhe com os artefatos gerados pelo engenheiro de sistemas para desenvolver um SI para novas missões. Os artefatos gerados pelo engenheiro de sistemas para o desenvolvimento do satélite podem servir de alicerce para a especificação do SI.

- Questão 2: Você acha que os modelos do sistema espacial gerados pelo engenheiro de sistemas contribuem com a especificação do sistema de informação em qual escala?
- Objetivo específico 3: Capturar a percepção dos respondentes sobre a contribuição do método para o processo de desenvolvimento do SI. A autora entende que o método proposto contribui com o processo de desenvolvimento dos sistemas de informação, pois aproveita o esforço realizado no domínio da engenharia de sistemas e dá melhores condições para o engenheiro de software entender as expectativas dos futuros usuários e especificar corretamente o SI.
 - Questão 3: Você acha que, de modo geral, o método proposto afeta positivamente o processo de desenvolvimento do sistema de informação em qual escala?
- Objetivo específico 4: Capturar a percepção dos respondentes sobre a contribuição do método para o resultado final do sistema de informação. A autora entende que a contribuição do método para o processo de especificação do SI afeta diretamente o resultado final, pois direciona para uma solução mais alinhada com as expectativas dos futuros usuários, além do que, a integração dos modelos da engenharia de sistemas com o SI traz as vantagens de reduzir retrabalho para os usuários e reduzir inconsistências nos dados.
 - Questão 4: Você acha que, de modo geral, o método proposto afeta positivamente o resultado final do sistema de informação em qual escala?

Além dessas questões, foi disponibilizado um espaço para escrita livre e opcional sobre as percepções sobre o método.

- Questão 5: Se você tem algum comentário adicional, crítica ou sugestão sobre o método proposto, sinta-se à vontade para compartilhar nesse espaço.

E duas questões finais com o propósito de identificar os perfis dos respondentes.

- Questão 6: Qual seu tempo de experiência com desenvolvimento de sistemas de informação?
 - Para responder a essa questão foi disponibilizada uma lista com 4 opções: menos de 2 anos; de 2 a 5 anos; de 5 a 10 anos; mais de 10 anos.
- Questão 7: Tem experiência na área espacial?
 - Para responder a essa questão foram disponibilizadas somente as opções “sim” e “não”.

Seguindo a orientação de (GIL, 2022) e (COHEN et al., 2017), antes da aplicação do questionário no real público-alvo, essas questões passaram por um pré-teste, foram validadas pelo orientador desta pesquisa e por mais dois profissionais experientes na área de desenvolvimento de sistemas de informação. Eles avaliaram:

- se o vídeo explicativo que deve preceder a resposta do questionário contém todas as informações necessárias para que os respondentes estejam aptos a dar uma resposta válida; e
- se a linguagem e os conceitos estão ao alcance dos respondentes (COHEN et al., 2017).

6.2.2 Resultados

Responderam o questionário 13 profissionais, sendo 3 com de 5 a 10 anos de experiência com desenvolvimento de SI e 10 com mais de 10 anos de experiência. Quanto a experiência na área espacial, 5 respondentes possuem experiência e 8 não possuem. O questionário respondido pelos profissionais está disponível no apêndice A.

O método foi avaliado positivamente pelos profissionais consultados, como pode ser observado nas Figuras 6.8, 6.9, 6.10 e 6.11, que apresentam uma compilação das respostas às perguntas 1, 2, 3, 4, respectivamente.

Figura 6.8 - Compilação das repostas da questão 1.

1. Você acha que a participação do engenheiro de software no processo de engenharia do sistema espacial afeta positivamente o desenvolvimento do sistema de informação em qual escala?

13 respostas

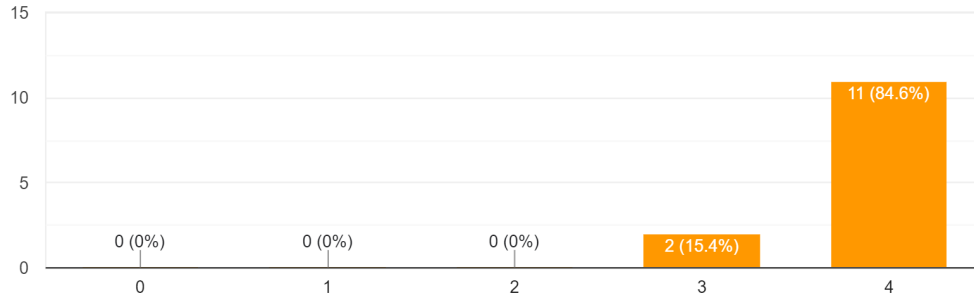


Figura 6.9 - Compilação das repostas da questão 2.

2. Você acha que os modelos do sistema espacial gerados pelo engenheiro de sistemas contribuem com a especificação do sistema de informação em qual escala?

13 respostas

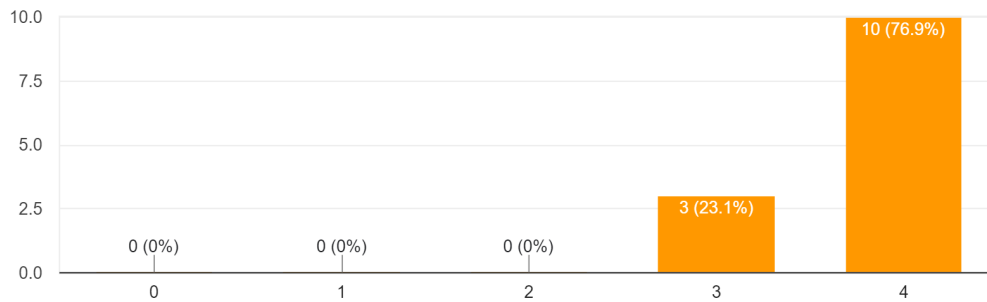


Figura 6.10 - Compilação das repostas da questão 3.

3. Você acha que, de modo geral, o método proposto afeta positivamente o processo de desenvolvimento do sistema de informação em qual escala?

13 responses

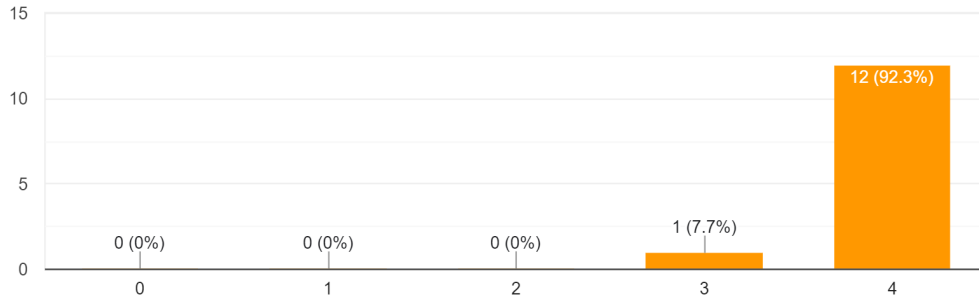
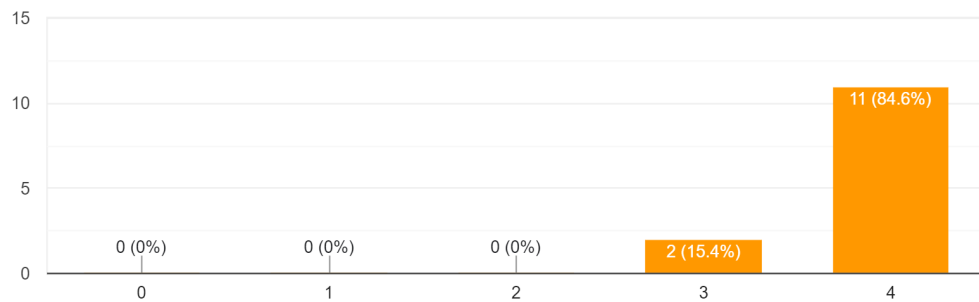


Figura 6.11 - Compilação das repostas da questão 4.

4. Você acha que, de modo geral, o método proposto afeta positivamente o resultado final do sistema de informação em qual escala?

13 responses



Os apontamentos feitos na questão 5, na qual os respondentes puderam escrever livremente suas considerações sobre o método, estão compilados na lista a seguir e comentados, no caso de questionamentos.

- O método proposto acelera o processo de desenvolvimento do SI porque diminui o ruído entre o engenheiro de software e os *stakeholders* e dá melhor entendimento ao engenheiro de software sobre as reais necessidades do

usuário, o que reflete positivamente no resultado final do SI.

- A participação do engenheiro de software em cada missão é importante, pois novas missões trazem novos requisitos e o SI deve estar preparado para atendê-los.
- Foi questionada a ausência do engenheiro de sistemas acompanhando o passo 3 do método (no qual é feita a especificação do SI pelo engenheiro de software).
 - Comentário da autora: De fato, na versão que foi apresentada aos profissionais, o engenheiro de sistemas não consta como um ator no passo 3. No passo 3 do método, o engenheiro de software interage com os *stakeholders* do sistema espacial que serão futuros usuários do SI. Em princípio, considerou-se a participação do engenheiro de sistemas nesse passo apenas no papel de usuário do SI, se esse for o caso. No processo de AIT, por exemplo, o engenheiro de sistemas também é um *stakeholder* do sistema espacial e possivelmente do SI. Porém, o questionamento do respondente levou a reflexão sobre o papel do engenheiro de sistemas também como um consultor sobre os requisitos do sistema espacial durante o passo 3 do método. A autora considerou pertinente incluir o engenheiro de sistemas também como ator do passo 3 independentemente de ser *stakeholder* do sistema espacial no processo do ciclo de vida a ser atendido pelo SI.
- Foi questionado sobre o paralelismo dos passos 2 e 3, já que o passo 2 gera entradas para o 3.
 - Comentário da autora: Os passos 2 e 3 ocorrem simultaneamente, uma vez que o engenheiro de software acompanha o passo 2 já trabalhando no 3, ou seja, tentando identificar como o SI pode colaborar com o atendimento aos requisitos do sistema espacial. A medida que os resultados do passo 2 amadurecem, a especificação do SI também amadurece. O processo de elicitação de requisitos de software é um processo de descoberta que envolve várias iterações, independentemente do uso do método. Passos 2 e 3, são portanto, paralelos e iterativos.
- Foi questionado sobre a possibilidade de gerar classes do SI a partir dos diagramas da engenharia de sistemas, visto que apresentam similaridades.

- Comentário da autora: De fato, como apontado na tese, existem muitas informações de interesse comum entre os modelos do domínio da engenharia de sistemas e do domínio da engenharia de software. Neste trabalho, para aproveitamento dessas informações foi proposta uma estratégia de integração que envolve a identificação dos elementos de interesse para o SI nos modelos da engenharia de sistemas e um mapeamento entre os modelos. Isso porque os modelos, embora tenham informações de interesse comum, não têm uma relação direta (como, por exemplo, um Bloco não corresponde a uma Classe), é preciso uma interpretação para ajustar os modelos do sistema espacial aos modelos do SI. A geração automática de classes para o SI a partir dos diagramas da engenharia de sistemas não é trivial, porém é sim interessante e merece ser objeto de estudos futuros.
- Foi questionado se, do ponto de vista de gestão de projetos, o método não tornaria as áreas de engenharia de sistemas e de engenharia de software muito dependentes. Por exemplo, atrasos em uma etapa da fase de projeto do satélite poderia impactar nas entregas do time de projeto de software.
 - Comentário da autora: De fato o método torna o desenvolvimento do SI dependente dos resultados das análises da Engenharia de Sistemas e isso é esperado, já que a proposta é um trabalho conjunto. O SI está sendo desenvolvido para atender processos do ciclo de vida do satélite. As funções da organização a serem apoiadas pelo SI nesses processos estão sendo definidas nas análises da engenharia de sistemas. Atrasos nas entregas da engenharia de sistemas significam que os requisitos do sistema espacial ainda não estão maduros, dessa forma, caso o SI avance de forma independente, aumenta o risco de que ele não atenda às necessidades.
- Foi levantada a hipótese da figura de um facilitador que transite entre os dois domínios para atuar removendo possíveis impedimentos entre as áreas, especialmente quando as equipes de engenharia de sistemas e de desenvolvimento do SI forem de organizações diferentes.
 - Comentário da autora: A ideia do método é o trabalho conjunto do engenheiro de sistemas e do engenheiro de software, o que descarta a necessidade de um facilitador. O engenheiro de software deve estar inserido no processo de engenharia de sistemas e deve interagir com o

engenheiro de sistemas. Ainda que a equipe de software seja externa à organização de engenharia de sistemas, o engenheiro de software deve participar das atividades do passo 2 do método acompanhando o trabalho da equipe de engenharia de sistemas.

7 DISCUSSÃO

7.1 Comparação do método proposto com as abordagens da fundamentação teórica

Três tópicos da fundamentação teórica desta tese merecem ser discutidos para melhor entendimento do trabalho realizado. No Capítulo 2 foram apresentadas duas abordagens de engenharia de sistemas MBSE (Seção 2.6) e SCE (Seção 2.5). E na Seção 2.1 foi apresentada a disciplina da Engenharia de Software.

A MBSE e a SCE são abordagens que não se excluem. São escolhas independentes e que podem se complementar.

A MBSE é uma abordagem que se preocupa em substituir documentos por modelos, que passam a ser a fonte principal de informação no processo de engenharia. A escolha se dá entre a MBSE e a engenharia de sistemas baseada em documentos.

Independentemente disso, a SCE é uma proposta mais abrangente para a engenharia de sistemas complexos em comparação com a Engenharia de Sistemas tradicional, já que integra a Engenharia Simultânea e a Engenharia de Sistemas. A escolha se dá entre a SCE e a Engenharia de Sistemas tradicional.

Tanto a SCE, como a Engenharia de Sistemas tradicional podem fazer uso da MBSE. E ambas também podem ser praticadas com a abordagem baseada em documentos.

Diante disso, o método proposto apresenta-se como uma especialização da SCE, assumindo o uso da MBSE na sua prática, o que chamamos de MBSCE.

Essa especialização aprofunda a MBSCE para desenvolver o elemento “sistema de informação (SI)” simultaneamente com o produto e a organização durante o processo de engenharia do sistema espacial e não de forma independente, desconectada desse processo.

Os processos da Engenharia de Software seguem normalmente para o desenvolvimento do SI, com a diferença de que se darão em grande parte, especialmente na atividade de especificação, dentro do processo da MBSCE.

Além disso, o método propõe a definição de uma convenção de modelagem e de um mapeamento da relação entre os elementos dos dois domínios que possibilitam o desenvolvimento de software para automatizar a integração entre os modelos. Isso contribui para a consistência das informações e minimiza retrabalho, uma vez que

mudanças nos modelos do sistema espacial podem ser refletidas automaticamente nos modelos do SI.

Em resumo, a MBSE e a SCE são direcionadas para o desenvolvimento do sistema espacial e os processos da Engenharia de Software são direcionados para o desenvolvimento dos sistemas de informação, enquanto o método proposto une os domínios da Engenharia de Sistemas e da Engenharia de Software para o desenvolvimento simultâneo e colaborativo do sistema espacial e dos sistemas de informação.

7.2 Contribuições frente ao estado da arte dos temas pesquisados

Considerando o estado arte dos temas estudados, apresentado no Capítulo 3, podemos observar que sistemas de informação para atendimento a produtos complexos, como satélites, são pouco discutidos na literatura. Este trabalho contribui com a área de Engenharia de Sistemas Espaciais trazendo um tema relevante, já que sistemas de informação têm o potencial de trazer melhorias para os processos de ciclo de vida de sistemas espaciais.

Além disso, de acordo com os trabalhos encontrados nas bases pesquisadas, com exceção dos trabalhos da própria autora, que já abordam o tema desta pesquisa, esta tese traz um trabalho inédito na área de Engenharia de Sistemas Espaciais. Diferente dos demais trabalhos da revisão de literatura que abordam ou o tema “sistemas de informação” ou o tema “MBSE”, este trabalho une esses dois temas com a proposta de integrar o desenvolvimento de SI com a MBSE.

Embora os trabalhos de [Halvorson e Thomas \(2022\)](#) e [Guo et al. \(2014\)](#) também abordem a integração da MBSE com o desenvolvimento de software, eles tratam de softwares embarcados que diferem de sistemas de informação por serem softwares acoplados no produto em desenvolvimento na MBSE, estão diretamente ligados a implementação do produto, enquanto sistemas de informação apoiam processos do ciclo de vida do produto e requerem abordagens diferentes para integração com a MBSE.

Este trabalho contribui com a Engenharia de Sistemas Espaciais trazendo uma extensão da aplicação da MBSE para atingir também o desenvolvimento de sistemas de informação de forma a afetar positivamente o desenvolvimento do sistema espacial.

7.3 Análise crítica do método proposto

Este trabalho apresenta um método que integra o desenvolvimento de sistemas de informação ao processo de engenharia de sistemas espaciais. Para a aplicação do método proposto é necessário que o engenheiro de software, responsável pelo desenvolvimento do SI, trabalhe junto com a equipe de Engenharia de Sistemas e caminhe com ela no processo de desenvolvimento do satélite. Isso implica numa mudança de paradigma que pode não ser de fácil implementação, principalmente se os sistemas de informação para apoio aos processos de ciclo de vida do satélite não forem entendidos como peças importantes para o sucesso da missão.

Um outro ponto que pode impactar no uso do método é o fato de ele ter como pilares a SCE e a MBSE, que por si só já exigem mudanças de paradigma numa área bastante conservadora, que é a área de engenharia espacial.

A SCE requer um olhar mais abrangente e menos tradicional para o desenvolvimento do sistema. Já a MBSE, embora esteja ganhando espaço nas grandes agências e indústrias da área espacial, as experiências da NASA, ESA, Airbus e Thales Alenia Space, relatadas no Capítulo 3, mostram também que essa não é uma transição fácil e rápida, é um projeto de anos que envolve esforços de convencimento e adaptação.

Além disso, as ferramentas para a prática da MBSE ainda são limitadas e não oferecem muitas facilidades. Para a integração automatizada entre os dois domínios, por exemplo, a ferramenta precisa oferecer recursos que possibilitem a leitura dos modelos.

A despeito de todas as dificuldades e desafios dessa transição para MBSCE e da incorporação do desenvolvimento dos sistemas de informação no processo de engenharia espacial, o método proposto aponta um caminho possível e com potencial de trazer ganhos tanto para o domínio da Engenharia de Sistemas, como para o domínio da Engenharia de Software.

7.4 Limitações desta pesquisa

- Embora espere-se que o método proposto seja aplicável também a outros produtos complexos, esta pesquisa limitou-se ao estudo do desenvolvimento de um satélite;
- Embora espere-se que o método proposto seja utilizável com outras linguagens de modelagem, esta pesquisa limitou-se ao uso da SysML para o

sistema espacial e da UML para o SI.

- Embora espere-se que, com que o método proposto, sistemas de informação possam ser desenvolvidos para qualquer processo do ciclo de vida de um satélite, esta pesquisa limitou-se a estudar o processo de AIT do satélite.

7.5 Considerações finais

Sistemas de informação têm o potencial de ajudar a realizar tarefas de maneira mais produtiva e eficiente (STAIR; REYNOLDS, 2017), para os processos do ciclo de vida dos produtos espaciais, essa realidade não é diferente.

Um sistema de informação desenvolvido sob medida adequa-se aos processos e às características da organização, enquanto ao optar por comprar um sistema de prateleira, a organização acaba por ter que adequar seus processos, sua forma de trabalho, ao sistema. Quanto mais específicas as atividades de uma organização, mais vantajosa a opção de desenvolver um sistema customizado.

A especificidade dos processos do ciclo de vida dos satélites e das organizações que implementam esses processos tornam o desenvolvimento de sistemas de informação customizados a opção mais adequada.

Diante disso, o tema desta pesquisa mostra-se relevante, pois traz para discussão o desenvolvimento de sistemas de informação customizados para processos do ciclo de vida de satélites, buscando de forma inédita uma integração entre o desenvolvimento do produto espacial e o desenvolvimento dos sistemas de informação.

Tendo em vista a necessidade de melhorar a forma como as informações são armazenadas, manipuladas e trocadas durante os processos do ciclo de vida dos satélites, entende-se que a mudança para o paradigma orientado a modelos em substituição ao orientado a documentos também tem grande relevância. O método proposto tem como pilares a SCE e MBSE (MBSCE). A MBSCE além de unir os benefícios das duas abordagens, favorece a integração do trabalho do engenheiro de software, que visa construir os sistemas de informação, com o trabalho do engenheiro de sistemas, que visa construir o sistema espacial, pelos seguintes motivos:

- a) os processos do ciclo de vida do produto são analisados desde o início do projeto, fornecendo informações em tempo hábil para o desenvolvimento de sistemas de informação que irão dar apoio a esses processos;

- b) aborda também o desenvolvimento das organizações que implementam os processos do ciclo de vida do produto e essas organizações são as usuárias dos sistemas de informação, e conseqüentemente afetadas por ele, portanto favorece a contribuição mútua entre os dois domínios;
- c) engenheiros de software tem familiaridade com o uso de modelos, que se tornam uma linguagem comum entre os dois domínios;
- d) o uso de modelos facilita criar mecanismos para troca automatizada de informações entre os dois domínios.

De forma geral, o método proposto favorece não só a integração, mas também a colaboração entre os domínios da Engenharia de Sistemas e da Engenharia de Software.

O desenvolvimento simultâneo do SI não só aproveita o esforço empreendido no domínio da Engenharia de Sistemas, como também colabora com o desenvolvimento do sistema espacial. As organizações são as usuárias dos sistemas de informação e também são parte da solução sistema, na ótica da SCE. Com o método proposto, o desenvolvimento do sistema espacial (produto e organização) contribui com o desenvolvimento dos sistemas de informação ao mesmo tempo que o desenvolvimento simultâneo dos sistemas de informação contribui com o desenvolvimento da organização.

O conhecimento prévio do engenheiro de sistemas sobre os sistemas de informação que darão apoio à execução dos processos da organização pode influenciar na arquitetura funcional e de implementação dessa organização, pois as funções da organização podem ser executadas de forma automatizada pelo SI, que torna-se parte da solução para atender a determinados requisitos dos *stakeholders*.

Tem-se, então, um método de desenvolvimento não só simultâneo, mas também colaborativo entre os dois domínios.

O método proposto traz ganhos de produtividade, pois o esforço do engenheiro de sistemas é aproveitado e não repetido pelo engenheiro de software, e ganhos de qualidade, pois a participação do engenheiro de software no processo da MBSCE dá mais clareza quanto às necessidades a serem atendidas, o que implica em um SI mais alinhado com a expectativa dos usuários.

Ainda é importante ressaltar que, embora o trabalho tenha sido exemplificado com

um satélite específico, o método proposto é aplicável a qualquer outro satélite, de pequeno, médio ou grande porte, além de ter potencial para ser aplicado também a outros produtos complexos, já que consiste exatamente em entender, junto com o engenheiro de sistemas, as necessidades de cada missão e dessas necessidades derivar também requisitos para os sistemas de informação que irão apoiar os processos do ciclo de vida do produto em desenvolvimento.

8 CONCLUSÃO

8.1 Atendimento aos objetivos

O método apresentado no Capítulo 5 e exemplificado no Capítulo 6 atende ao objetivo geral desta pesquisa que é “propor um método para integrar o desenvolvimento dos sistemas de informação que darão suporte aos processos do ciclo de vida de satélites ao processo de Engenharia de Sistemas Espaciais, de forma que o esforço realizado no domínio da Engenharia de Sistemas seja aproveitado no domínio da Engenharia de Software”.

8.1.1 Atendimento aos objetivos específicos

- a) Identificação de oportunidades: identificar, nas abordagens de Engenharia de Sistemas e nos trabalhos já publicados nessa área, oportunidades para integrar o desenvolvimento de sistemas de informação com o desenvolvimento de satélites.

Este objetivo específico foi atendido, pois pesquisando as abordagens de engenharia que constam na literatura e os trabalhos já publicados nessa área, Capítulos 2 e 3, foi identificado que a SCE e a MBSE tem características que favorecem a integração do desenvolvimento de sistemas de informação com o desenvolvimento de satélites, o que embasou os resultados deste trabalho.

- b) Avaliação de viabilidade: avaliar a viabilidade de integrar o desenvolvimento de sistemas de informação com o desenvolvimento de satélites.

No Capítulo 4 foi avaliada a viabilidade de integração do desenvolvimento de sistemas de informação no processo da SCE aplicada para desenvolvimento de satélites. Foi demonstrado que adotando a SCE para a engenharia de um satélite, é viável integrar o trabalho do engenheiro de software, que visa desenvolver os sistemas de informação, no processo de desenvolvimento desse satélite. O uso da SCE em conjunto com a MBSE também foi avaliado e foi demonstrado que o uso conjunto dessas duas abordagens viabiliza uma integração automatizada entre os modelos do sistema espacial e os modelos dos sistemas de informação.

- c) Proposta de um método: propor um método que integre o desenvolvimento de sistemas de informação com o desenvolvimento de satélites.

Este objetivo específico foi atendido no Capítulo 5, o qual apresenta um

método que integra o desenvolvimento de sistemas de informação no processo de desenvolvimento de satélites.

- d) Aplicação do método: exemplificar a aplicação do método proposto com um caso real de desenvolvimento de sistema de informação para apoiar um processo do ciclo de vida de um satélite.

Este objetivo específico foi atendido no Capítulo 6, o qual exemplifica a aplicação do método proposto para o desenvolvimento de um SI para apoiar o processo de AIT do satélite AMAZONIA1.

- e) Avaliação do método por profissionais da área de desenvolvimento de sistemas de informação: submeter o método proposto a uma avaliação por desenvolvedores de sistemas de informação.

O método proposto foi submetido a avaliação por profissionais com experiência no desenvolvimento de SI por meio de um questionário, conforme apresentado na Seção 6.2

- f) Análise crítica do método: fazer uma análise crítica do método proposto. Este objetivo específico foi atendido na Seção 7.3.

8.2 Contribuições

Os itens a seguir destacam as contribuições deste trabalho de doutorado para a área de Engenharia de Sistemas Espaciais:

- traz para discussão um assunto ainda não abordado na literatura pesquisada e que tem o potencial de trazer benefícios para a área de Engenharia de Sistemas Espaciais: o relacionamento entre o desenvolvimento de sistemas espaciais e o desenvolvimento dos sistemas de informação que apoiam os processos do ciclo de vida do produto espacial.
- propõe a integração do trabalho do engenheiro de software, que visa construir os sistemas de informação, no processo de engenharia do sistema espacial, o que tem o potencial de reduzir retrabalho e possibilitar contribuições mútuas entre os domínios da Engenharia de Sistemas e da Engenharia de Software.
- propõe a MBSCE, que é a prática da MBSE guiada pelos princípios da SCE, o que traz como benefícios a solução mais abrangente da SCE com as vantagens da MBSE;

- propõe uma convenção de modelagem e um mapeamento que possibilitem a integração dos modelos de engenharia do sistema espacial com os modelos dos sistemas de informação, vínculo que favorece a consistência das informações nos dois domínios. Conforme a revisão de literatura realizada, o possível relacionamento e a possível colaboração entre os modelos construídos no domínio da Engenharia de Sistemas e os modelos construídos no domínio da Engenharia de Software, para SI, são assuntos ainda não explorados;
- propõe a automatização da integração dos modelos por meio de software, o que minimiza retrabalho e inconsistências, já que mudanças nos modelos do sistema espacial podem ser refletidas automaticamente no SI.
- estende a aplicação da MBSCCE para atingir também o domínio da engenharia de software afetando positivamente o desenvolvimento do sistema espacial.

A contribuição principal deste trabalho de doutorado para a área de Engenharia de Sistemas Espaciais é a proposta de um método que integra o desenvolvimento de sistemas de informação com o desenvolvimento de satélites, de forma que o esforço investido nas atividades de Engenharia de Sistemas seja aproveitado no domínio da Engenharia de Software.

Embora a pesquisa tenha sido desenvolvida utilizando o caso de um satélite, o método proposto tem potencial de ser aplicável para o desenvolvimento de qualquer outro produto complexo, da área espacial ou não, cujos processos do ciclo de vida possam se beneficiar do apoio de sistemas de informação.

8.3 Trabalhos futuros

Esta seção apresenta sugestões de trabalhos futuros que são desejáveis para a continuidade deste trabalho.

8.3.1 Desenvolvimento de uma ferramenta direcionada para a aplicação do método proposto.

Neste trabalho a autora utilizou a ferramenta IBM Rhapsody (IBM, 2023b) para modelagem, porém o uso do método será facilitado com uma ferramenta que ofereça aos engenheiros recursos para edição intuitiva de acordo com o que é proposto pelo método. Por exemplo, a ferramenta IBM Rhapsody possui um *add-on* que direciona

o engenheiro no método Harmony (IBM, 2023a) para engenharia de sistemas e a ferramenta Capella direciona o engenheiro no método Arcadia (ROQUES, 2017) para engenharia de sistemas.

8.3.2 Desenvolvimento de um *framework* para direcionar a integração automatizada de modelos de engenharia de sistemas com modelos de engenharia de software.

Neste trabalho foi demonstrada a viabilidade de integração automatizada, porém uma contribuição relevante seria a definição de um padrão para desenvolvimento dos modelos visando integração.

8.3.3 Estudar a possibilidade de gerar modelos para o SI automaticamente a partir dos modelos do sistema espacial.

Neste trabalho não foi explorada a geração automática de modelos do SI a partir dos modelos da engenharia de sistemas. Como visto nesta tese, é preciso uma interpretação para ajustar os modelos do sistema espacial aos modelos do SI. A complexidade de gerar modelos do SI automaticamente não está propriamente no mecanismo de ler modelos do sistema espacial e gerar modelos do SI e sim em automatizar a interpretação dos modelos dos dois domínios. A possibilidade dessa interpretação automatizada deve ser objeto de um novo estudo.

8.3.4 Estudar a possibilidade de evoluir o método proposto para a utilização em conjunto com PBSE (*Pattern-Based Systems Engineering* - Engenharia de Sistemas baseada em padrões).

A PBSE é uma abordagem de engenharia de sistemas que oferece maneiras específicas de estender a MBSE para explorar Padrões, focando no reuso de conhecimento no desenvolvimento de sistemas complexos (SCHINDEL; PETERSON, 2013). De acordo com Schindel e Peterson (2013) Padrões trazem benefícios para Engenharia de Sistemas por meio de melhores modelos disponíveis a custos mais baixos.

Visto que sistemas de informação provavelmente não serão exclusivos para uma única missão e que missões espaciais normalmente apresentam elementos comuns, a PBSE tem potencial de trazer melhorias para a engenharia de sistemas espaciais. Dessa forma, como um trabalho futuro, um estudo de uma possível evolução do método proposto para integrar o desenvolvimento do SI com a PBSE mostra-se interessante.

REFERÊNCIAS BIBLIOGRÁFICAS

- APACHE. **Groovy language documentation**. 2023. Disponível em: <<<http://www.groovy-lang.org/single-page-documentation.html>>>. Acesso em: 21 abr. 2023. 80
- AQUINO, E. R.; LOUREIRO, G.; BURGER, E. E. Mbse para engenharia de sistema de cubesats. In: SEMINÁRIO DE INICIAÇÃO CIENTÍFICA E INICIAÇÃO EM DESENVOLVIMENTO TECNOLÓGICO E INOVAÇÃO (SICINPE), 2018, São José dos Campos. **Anais...** São José dos Campos: INPE, 2018. 32
- BERNABEL, G.; CORALLO, A.; LOMBARDO, R.; MACI, S.; GALLI, V.; CANNOLETTA, D.; NOTARO, A. Development of a framework to support virtual review within complex-product lifecycle management. In: PAOLIS L. T.; MONGELLI, A. (Ed.). **Augmented and virtual reality**. Berlin: Springer, 2015. p. 449 – 457. 34
- BONNET, S.; VOIRIN, J.-L.; EXERTIER, D.; NORMAND, V. Not (strictly) relying on sysml for mbse: language, tooling and development perspectives: the arcadia/capella rationale. In: ANNUAL IEEE SYSTEMS CONFERENCE, 2016. **Proceedings...** Orlando: IEEE, 2016. 30
- BUNGE, M. **Epistemologia: curso de atualização**. 2. ed. [S.l.]: Editora da Universidade de São Paulo, 1980. 55
- BURGER, E. E. **A conceptual MBSE framework for satellite AIT planning**. Tese (Doutorado em Engenharia e Gerenciamento de Sistemas Espaciais) — Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos, 2018. 3, 32
- CALIO, E.; GIORGIO, F. D.; PASQUINELLI, M. Deploying model-based systems engineering in thales alenia space italia. In: ITALIA CONFERENCE ON SYSTEMS ENGINEERING, 2., 2016, Turin, Itália. **Proceedings...** [S.l.], 2016. p. 112–118. 30
- CASTRO, J.; KOLP, M.; MYLOPOULOS, J. Towards requirements-driven information systems engineering: the tropos project. **Information Systems**, v. 27, n. 6, p. 365–389, 2002. ISSN 0306-4379. Disponível em: <<<https://www.sciencedirect.com/science/article/pii/S0306437902000121>>>. 14
- CHIANG, T.; TRAPPEY, A.; KU, C. Using a knowledge-based intelligent system to support dynamic design reasoning for a collaborative design community. **International Journal of Advanced Manufacturing Technology**, v. 31, n. 5-6, p. 421 – 433, 2006. Cited by: 22. 34
- COHEN, L.; MANION, L.; MORRISON, K. **Research methods in education**. 8. ed. London e New York: Routledge, 2017. ISBN 1138209880. 85

COICEV, M.; LOUREIRO, G. Mbse i&t sysml applied to the development of egse for satellites assembly, integration and testing (ait) - a practical case.

International Journal of Advanced Engineering Research and Science, v. 6, p. 391–397, 01 2019. 32

CZIEP, T.; STINDL, T.; STÄBLER, T. The digital concurrent engineering platform dcep. In: INTERNATIONAL ASTRONAUTICAL CONGRESS, 72., 2021, Dubai. **Proceedings...** [S.l.], 2021. 34

DELLIGATTI, L. **SysML distilled: a brief guide to the systems modeling language**. [S.l.]: Addison-Wesley Professional, 2013. ISBN 0321927869. 19, 20, 21

ECLIPSE. **Equivalences and differences between SYSML and Arcadia/Capella**. 2017. Disponível em: <<https://www.eclipse.org/capella/arcadia_capella_sysml_tool.html>>. Acesso em: 29 maio 2020. 30

_____. **Eclipse Papyrus**. c2019. Disponível em: <<<https://www.eclipse.org/papyrus/>>>. Acesso em: 29 maio 2020. 31

EUROPEAN SPACE AGENCY (ESA). **ECSS-M-ST-10C Rev.1: Space project management - Project planning and implementation**. Noordwijk: [s.n.], 2010. 16, 17

_____. **What is the CDF?** 2020. Disponível em: <<http://www.esa.int/Enabling_Support/Space_Engineering_Technology/CDF/What_is_the_CDF>>. Acesso em: 29 maio 2020. 29

FISCHER, P.; LÜDTKE, D.; LANGE, C.; ROSHANI, F.-C.; DANNEMANN, F.; GERNDT, A. Implementing model-based system engineering for the whole lifecycle of a spacecraft. **CEAS Space Journal**, v. 9, n. 3, p. 351–365, 2017. 35

FOX, K.; CURTIS, H.; RUFFOLO, M. Epic: transitioning dart i&t from a human-to-human process to a digital environment. In: IEEE AEROSPACE CONFERENCE, 2023. **Proceedings...** [S.l.], 2023. p. 1–11. 35

FRANCO, R. **Um modelo de referência MBSE da interface entre satélite e um veículo lançador escolhido usando AHP**. Tese (Mestrado em Engenharia e Gerenciamento de Sistemas Espaciais) — Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos, 2018. 32

FRIEDENTHAL, S.; MOORE, A.; STEINER, R. **OMG Systems Modeling Language (OMG SysML™) tutorial**. [S.l.]: INCOSE, 2009. 21, 22

_____. **A practical guide to SysML: the systems modeling language**. 3. ed. [S.l.]: Morgan Kaufmann, 2014. 20, 21, 56

FUJIKAWA, H.; OGASAWARA, T. A case study of 3d electromechanics and reliability integrated design on knowledge based design process. In: CHU, H.; AGUILAR, J.; ROLLAND, J.; LACHOWICZ, S. (Ed.). **ISAS/CITSA 2004:**

International Conference on Cybernetics and Information Technologies, Systems and Applications and 10th International Conference on Information Systems Analysis and Synthesis, Vol 4, Proceedings. [S.l.], 2004. p. 53–57. 35

GIL, A. C. **Como elaborar projetos de pesquisa.** 4. ed. São Paulo: Atlas, 2002. ISBN 85-224-3169-8. 8

_____. _____. 7. ed. Barueri: Atlas, 2022. ISBN 6559771636. 8, 83, 85

GREGORY, J.; BERTHOUD, L.; TRYFONAS, T.; ROSSIGNOL, A.; FAURE, L. The long and winding road: MBSE adoption for functional avionics of spacecraft. **Journal of Systems and Software**, v. 160, 2020. 30

GUEDES, G. T. A. **UML 2 - uma abordagem prática.** 3. ed. São Paulo: Novatec, 2018. ISBN 978-8575226469. 14, 15, 56

GUO, J.; GILL, E.; FIGARI, S. Model-based systems engineering to support the development of nano-satellites. In: INTERNATIONAL ASTRONAUTICAL CONGRESS, 65., 2014, Toronto. **Proceedings...** Toronto: IAF, 2014. p. 6991–7003. 35, 36, 92

HALVORSON, M.; THOMAS, L. D. Architecture framework standardization for satellite software generation using mbse and f prime. In: IEEE AEROSPACE CONFERENCE, 2022, Montana. **Proceedings...** Montana: AIAA, 2022. p. 1–20. 35, 36, 92

HOFFMANN, L. T.; BRANCO, M. S. A.; BRANCO, R. H. F.; GOBBI, D.; SILVA, C. M. Z.; PERONDI, L. F. A engenharia de sistemas espaciais da missão equars. In: SIMPÓSIO BRASILEIRO DE GEOFÍSICA ESPACIAL E AERONOMIA, 2018, Santa Maria. **Anais...** Santa Maria, 2018. 31, 32

HOLLADAY, J. B.; KNIZHNIK, J.; WEILAND, K. J.; STEIN, A.; SANDERS, T.; SCHWINDT, P. Mbse infusion and modernization initiative (miami): “hot” benefits for real nasa applications. In: IEEE AEROSPACE CONFERENCE, 2019. **Proceedings...** Montana: IEEE, 2019. p. 1–14. 26, 27, 28

IBM. **Rational rhapsody API reference manual.** [S.l.]: IBM Corporation, 2009. 80

_____. **Essentials of IBM rational rhapsody for systems engineers v.7.6.1 - student manual vol 1.** [S.l.]: IBM Corporation, 2012. 21, 22

_____. **The harmony process.** 2023. Disponível em: <<<https://www.ibm.com/docs/en/engineering-lifecycle-management-suite/design-rhapsody/9.0.2?topic=secsyscontroller-harmony-process>>>. Acesso em: 15 nov. 2023. 100

_____. **IBM engineering systems design rhapsody.** 2023. Disponível em: <<<https://www.ibm.com/products/systems-design-rhapsody>>>. Acesso em: 21 abr. de 2023. 10, 69, 79, 99

INCOSE. **Incase systems engineering handbook: a guide for system life cycle processes and activities**. 4. ed. [S.l.]: Wiley, 2015. 6, 21

INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS (INPE). **Minuta da reunião realizada em 21 de setembro de 2017 no LIT/INPE, LIT25-LIT05-MR-001**. 2017. São José dos Campos: INPE/LIT, 2017. 46

_____. **Amazon Mission**. 2022. Disponível em: <<<http://www.inpe.br/amazonia1/en/>>>. Acesso em: 13 jul. 2022. 39

_____. **Ensaio dinâmicos**. 2022. Disponível em: <<<http://www.lit.inpe.br/pt-br/vibracao>>>. Acesso em: 13 jul. 2022. 40

JACOBSON, I. **Object oriented software engineering: a use case driven approach**. New York: Addison Wesley, 1992. 15

JESSICA. **Applying MBSE to a space mission**. Sep 2017. Disponível em: <<<https://blogs.esa.int/cleanspace/2017/08/28/applying-mbse-to-a-space-mission>>>. Acesso em: 29 maio 2020. 29

_____. **From active debris removal to in-orbit servicing: the legacy of e.deorbit**. Nov 2018. Disponível em: <<<https://blogs.esa.int/cleanspace/2018/11/26/from-active-debris-removal-to-in-orbit-servicing-the-legacy-of-e-deorbit>>>. Acesso em: 29 mai. 2020. 29

JIANG, P.; MAIR, Q. An actor-oriented approach to distributed product management systems. In: IEEE COMPUTER SOCIETY'S INTERNATIONAL COMPUTER SOFTWARE AND APPLICATIONS CONFERENCE. **Proceedings...** Oxford: IEEE, 2002. 34

LOUREIRO, G. **A systems engineering and concurrent engineering framework for the integrated development of complex products**. Tese (Doutorado) — Loughborough University, Loughborough UK, 1999. 17, 55

LOUREIRO, G.; PANADES, W.; SILVA, A. Lessons learned in 20 years of application of systems concurrent engineering to space products. **Acta Astronautica**, v. 151, p. 44–52, 2018. ISSN 0094-5765. Disponível em: <<<https://www.sciencedirect.com/science/article/pii/S0094576518304272>>>. 17, 18, 19

LOUREIRO, M.; LEANEY, D. A systems engineering environment for integrated satellite development. **Acta Astronautica**, v. 44, n. 7, p. 425–435, 1999. ISSN 0094-5765. Disponível em: <<<https://www.sciencedirect.com/science/article/pii/S0094576599000892>>>. 35

MAGNET, A. **Designing freedom of flight**. jun 2016. Disponível em: <<<https://www.eclipse.org/papyrus/resources/airbus-usecasesstory.pdf>>>. Acesso em: 29 maio 2020. 31

MALETZ, M.; ZOIER, M.; CRISTEA, L.; BRISSON, D. The semantic interaction of knowledge, data and processes within the product development process. In: NORDDDESIGN CONFERENCE, 2008, Tallin, Estonia. **Proceedings...** [S.l.], 2008. 34

MELO, A. C. **Desenvolvendo aplicações com UML 2.2**. 3. ed. Rio de Janeiro: Brasport, 2011. ISBN 978-8574524443. 14, 15

OBJECT MANAGEMENT GROUP (OMG). **Unified modeling language - OMG document number formal/2017-12-05**. [S.l.]: OMG, 2017. 14, 15

PARROTT, E. **The value of successful MBSE adoption**. Allen, TX; United States, 2016. Disponível em:
<https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20170001634.pdf>. Acesso em: 29 maio 2020. 26

PARSIFAL. **About Parsifal**. 2023. Disponível em:
<<<https://parsif.al/about/>>>. Acesso em: 17 out. 2023. 10

PRATOMO, B.; FITRIANINGSIH, E.; NASSER, E. N. Implementation of the space project documentation standard using web-based application software. In: . [S.l.: s.n.], 2021. D5. Cited by: 0. 34

PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de software: uma abordagem profissional**. 8. ed. Porto Alegre: AMGH, 2016. ISBN 978-85-8055-534-9. 13, 14

REID, S.; MATTHYSSEN, A.; HEINEN, W. Model based functional verification; strengthening the link between testing and operations. In: SPACEOPS CONFERENCE, 2016, Daejeon, Korea. **Proceedings...** [S.l.]: AIAA, 2016. 35

ROQUES, P. **Systems architecture modeling with the arcadia method: a practical guide to capella**. London: Iste Press - Elsevier, 2017. ISBN 1785481681. 100

ROSING, M. v.; SCHEER, A.-W.; SCHEEL, H. v. **The complete business process handbook, volume 1 - body of knowledge from process modeling to BPM**. [S.l.: s.n.], 2015. ISBN 978-0127999593. 1, 15

SCHINDEL, B.; PETERSON, T. **Introduction to Pattern-Based Systems Engineering (PBSE): leveraging MBSE techniques**. 2013. Disponível em:
<<https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:pbse_tutorial_glrc_2013_v1.6.3_reduced_pdf.pdf>>. Acesso em: 11 dez. 2023. 100

SHARPLES, R. Continuous model based system engineering (mbse) improvement via human system integration and customer change. In: KARWOWSKI, W.; AHARAM, T. (Ed.). **Intelligent human system integration**. Berlin: Springer, 2018. 30

SILVA, A. C.; LOUREIRO, G. Integrated development of space systems - design for assembly, integration and testing of satellites - d4ait. In: IEEE INTERNATIONAL CONFERENCE ON INDUSTRIAL ENGINEERING AND ENGINEERING MANAGEMENT, 2011. **Proceedings...** [S.l.], 2011. p. 590–594. 2

SILVA, A. C. d. P. **Avaliação do método para desenvolvimento de sistemas de informação para apoiar processos do ciclo de vida de satélites**. 2023. Disponível em: <<<https://forms.gle/LED39QujxpVapj1U7>>>. 82

SILVA, A. C. d. P.; LOUREIRO, G. The relationship between the model based systems engineering models and information systems to support space products lifecycle processes. In: INTERNATIONAL ASTRONAUTICAL CONGRESS, 2018, Bremen, Germany. **Proceedings...** [S.l.], 2018. 34, 78

_____. Iniciativas de adesão à mbse na área espacial. In: RODRIGUES, A. C.; BARBOSA, A. L.; TEIXEIRA, A. F.; BATISTA, C. L. G.; OLIVEIRA, C. M. d.; CUELLAR, D. A. (Ed.). **Anais do 11º Workshop em Engenharia e Tecnologia Espaciais**. São José dos Campos: Instituto Nacional de Pesquisas Espaciais (INPE), 2020. ISSN 2177-3114. Disponível em: <<<http://urlib.net/rep/8JMKD3MGPDW34R/433LTT2>>>. Acesso em: 06 jan. 2021. 9, 24

SILVA, A. C. de P.; LOUREIRO, G. Simultaneous and collaborative development of satellites and information systems to support their lifecycle processes. **Information Systems**, v. 111, p. 102125, 2023. ISSN 0306-4379. Disponível em: <<<https://www.sciencedirect.com/science/article/pii/S030643792200103X>>>. 34

SILVA, E.; MENEZES, E. **Metodologia da pesquisa e elaboração de dissertação**. 4. ed. [S.l.]: Universidade Federal de Santa Catarina - UFSC, 2005. 8

SILVA, L. A. **Teprom - framework de engenharia simultânea de sistemas baseado em modelos para testes funcionais elétricos em satélites**. Tese (Doutorado em Engenharia e Gerenciamento de Sistemas Espaciais) — Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos, 2023. 2

SKOBELEV, P.; LAKHIN, O.; POLNIKOV, A.; SIMONOVA, E. Approach to the solution of aerospace product lifecycle management problem based on network-centric principles. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, v. 9266, p. 169 – 178, 2015. Cited by: 4. 34

SOMMERVILLE, I. **Engenharia de software**. 10. ed. São Paulo: Pearson Universidades, 2018. ISBN 978-8543024974. 13, 14, 15, 16, 63

STAIR, R. M.; REYNOLDS, G. W. **Principles of information systems**. 13. ed. [S.l.]: Cengage Learning, 2017. ISBN 978-1305971776. 15, 16, 94

STARK, J. **Product Lifecycle Management**. 3st. ed. [S.l.]: Springer, 2016. 33

- THALESGROUP. **Space for life**. 2020. Disponível em: <<<https://www.thalesgroup.com/en/global/activities/space>>>. Acesso em: 29 maio 2020. 29
- VIOLANTE, M. G.; VEZZETTI, E.; ALEMANNI, M. An integrated approach to support the requirement management (rm) tool customization for a collaborative scenario. **International Journal on Interactive Design and Manufacturing**, v. 11, n. 2, p. 191 – 204, 2017. 34
- WEILAND, K. J.; HOLLADAY, J. Model-based systems engineering pathfinder: informing the next steps. **INCOSE International Symposium**, v. 27, n. 1, p. 1594–1608, 2017. Disponível em: <<<https://onlinelibrary.wiley.com/doi/abs/10.1002/j.2334-5837.2017.00449.x>>>. 25, 26
- WOLAHAN, A.; BIESBROEK, R. Model based systems engineering applied to esa’s e.deorbit mission. In: 68 INTERNATIONAL ASTRONAUTICAL CONGRESS 2017, 68., 2017, Adelaide, Australia. **Proceedings...** [S.l.], 2017. 29
- XU, W.; LI, N.; TANG, H.; CHA, J. Research on software resource sharing management in collaborative design environment based on remote virtual desktop. In: ISPE CONFERENCE, 2014. **Proceedings...** [S.l.], 2014. p. 278 – 286. 34
- XUEMEI, L.; XIAOLANG, Y. A visualization framework for product manufacturing data. **Procedia CIRP**, v. 104, p. 1046 – 1051, 2021. 34
- ZHAO, C.; MING, X.; WANG, X. Collaborative project management with supplier involvement in complex product development. **Journal of Computational Information Systems**, v. 6, n. 11, p. 3491 – 3500, 2010. Cited by: 2. Disponível em: <<<https://www.scopus.com/inward/record.uri?eid=2-s2.0-78649972483&partnerID=40&md5=6bc187ad9137b8876d63c910e9116d99>>>. 34
- ZHAO, S.; YIN, M. Research on an information integration framework on the large complex product development pmis with erp. **Materials Science Forum**, v. 532-533, p. 917 – 920, 2006. Cited by: 2. 34
- ZOWGHI, D.; COULIN, C. Requirements elicitation: a survey of techniques, approaches, and tools. In: AURUM, A.; WOHLIN, C. (Ed.). **Engineering and Managing Software Requirements**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005. p. 19–46. 14

APÊNDICE A - QUESTIONÁRIO PARA AVALIAÇÃO DO MÉTODO PROPOSTO

Este apêndice apresenta o questionário respondido por especialistas em desenvolvimento de sistemas de informação para avaliação do método proposto nesta tese.

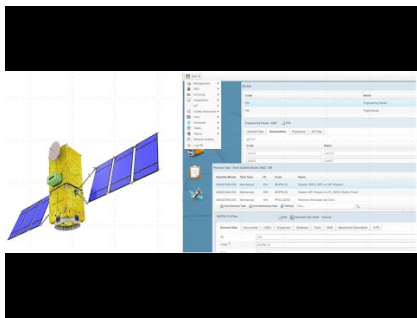


Avaliação do método para desenvolvimento de Sistemas de Informação para apoiar processos do ciclo de vida de satélites

Para responder a esse questionário, após assistir o vídeo de apresentação do método, considere que você é responsável pelo desenvolvimento de um sistema de informação para apoiar um processo do ciclo de vida de um satélite, como o processo de AIT utilizado como exemplo no vídeo.

This form is automatically collecting emails from all respondents. [Change settings](#)

Vídeo de apresentação do método:



1. Você acha que a participação do engenheiro de software no processo de engenharia do sistema espacial afeta positivamente o desenvolvimento do sistema de informação em qual escala? *

	0	1	2	3	4	
nada	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	muito

2. Você acha que os modelos do sistema espacial gerados pelo engenheiro de sistemas contribuem com a especificação do sistema de informação em qual escala? *

	0	1	2	3	4	
nada	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	muito

3. Você acha que, de modo geral, o método proposto afeta positivamente o processo de desenvolvimento do sistema de informação em qual escala?

	0	1	2	3	4	
nada	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	muito



4. Você acha que, de modo geral, o método proposto afeta positivamente o resultado final do sistema de informação em qual escala? *

	0	1	2	3	4	
nada	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	muito

5. Se você tem algum comentário adicional, crítica ou sugestão sobre o método proposto, sinta-se à vontade para compartilhar nesse espaço.

Long answer text

6. Qual seu tempo de experiência com desenvolvimento de sistemas de informação? *

- Menos de 2 anos
- de 2 a 5 anos
- de 5 a 10 anos
- Mais de 10 anos

7. Tem experiência na área espacial? *

- Sim
- Não



Avaliação do método para desenvolvimento de Sistemas de Informação para apoiar processos do ciclo de vida de satélites

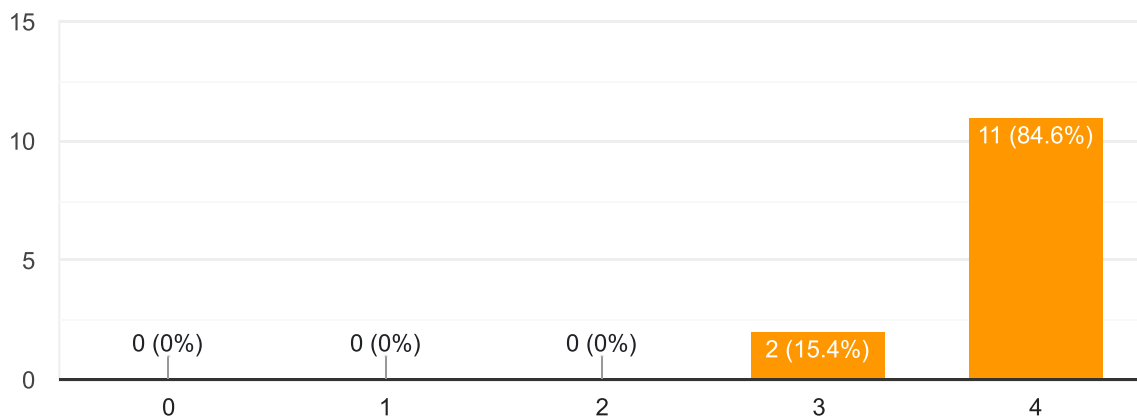
13 responses

[Publish analytics](#)

1. Você acha que a participação do engenheiro de software no processo de engenharia do sistema espacial afeta positivamente o desenvolvimento do sistema de informação em qual escala?

 Copy

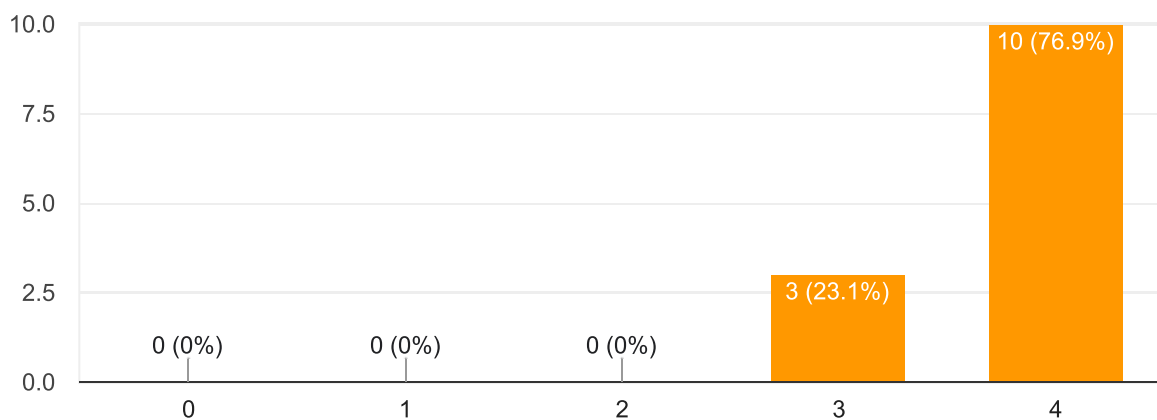
13 responses



2. Você acha que os modelos do sistema espacial gerados pelo engenheiro de sistemas contribuem com a especificação do sistema de informação em qual escala?

 Copy

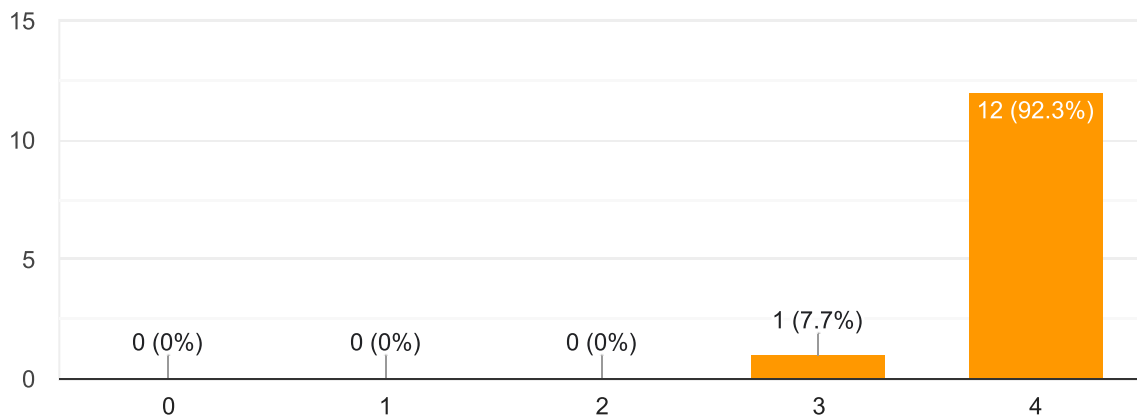
13 responses



3. Você acha que, de modo geral, o método proposto afeta positivamente o processo de desenvolvimento do sistema de informação em qual escala?



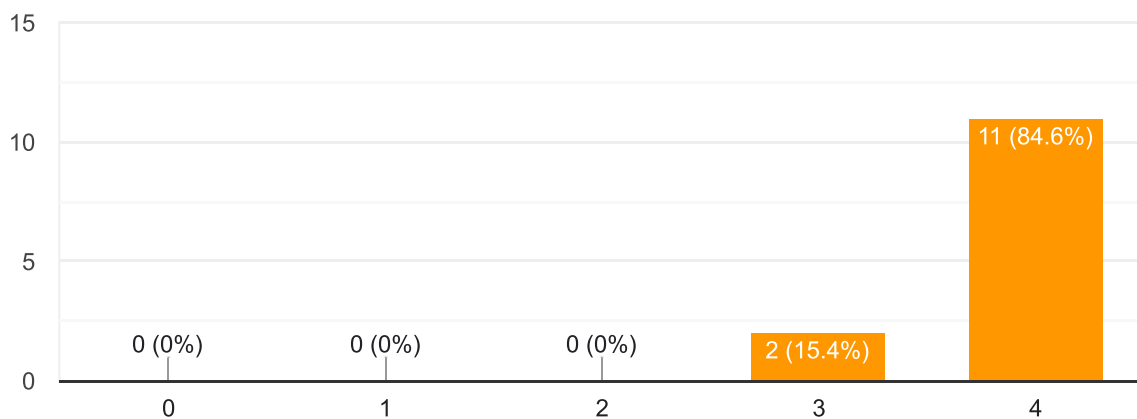
13 responses



4. Você acha que, de modo geral, o método proposto afeta positivamente o resultado final do sistema de informação em qual escala?



13 responses



5. Se você tem algum comentário adicional, crítica ou sugestão sobre o método proposto, sinta-se à vontade para compartilhar nesse espaço.

8 responses

Método contribui para o avanço da MBSE para uso em conjunto com o desenvolvimento de SI na área espacial

A apresentação nos leva a entender que provavelmente dados de insumos e equipamentos utilizados no processo de AIT poderão ser considerados como entrada de dados no sistema de informação, como certificados de calibração de equipamentos e sensores utilizados na medição de vibração do satélite; a depender do escopo.

O objetivo do levantamento de requisitos (visão do engenheiro de software) é especificar um sistema de informação que ajude a atender os requisitos do sistema espacial, dos resultados das análises do engenheiro de sistemas. Tudo que o engenheiro de software entender que pode contribuir com isso vai servir de informação para a especificação do sistema de informação.

Juntar as duas áreas de Engenharia de Software e Desenvolvimento de Sistemas, para trocarem informações principalmente na fase de levantamento de dados junto ao Stakeholder, faz com que se evite informações duplicadas e desconectas.

Alguns dos modelos gerados pelas arquiteturas do sistema são para refinamento e validação de requisitos de sistema/subsistemas e podem ser descartados a depender da solução adotada.

Conhecendo um pouco do contexto, é perceptível a importância da participação do Engenheiro de Software no processo, principalmente pelo motivo de que cada satélite tem sua missão e suas especificações, dentro de um processo complexo de desenvolvimento, e a criação de um sistema de informação tem que ser genérica o suficiente para permitir os registros relacionados a vários satélites. Pela minha experiência com desenvolvimento de software, eu acredito que, com a participação do Engenheiro de Software em todo o processo e com o uso dos artefatos produzidos pelo Engenheiro Espacial, o ruído entre cliente e engenheiro de software será reduzido, permitindo que o desenvolvimento do sistema de informação seja mais rápido.

No passo 3 do método proposto, talvez seja interessante o Engenheiro de Sistemas acompanhar, assim como o Engenheiro de Software acompanha no passo 2. Isso garante que não acha nenhum ruído na informação e possíveis dúvidas podem ser sanadas mais rapidamente. Talvez não seja tão paralelizável esses passos, uma vez que o 2 irá gerar inputs para o 3. Me parece que se for algo feito em conjunto, embora possa levar um maior tempo, pelo menos a princípio, haverá ganhos a coesão das informações.

Um ponto que me chamou atenção foi a questão de gerar as classes a partir dos diagramas, a natureza do projeto, me parece ter muitos pontos que podem ser aproveitados de um projeto para outro. Talvez seja legal uma base e ter essa geração apenas em pontos específicos, ou seja, teria meio que um template customizável, mas isso já seria entrando na implementação em si.

Concluindo: Acho que terá alguns pontos a se discutir, que serão encontrados quando se



colocar em prática, mas como proposta de modelo, está muito bom mesmo, parabéns, ficou bem clara a apresentação e o objetivo da proposta.

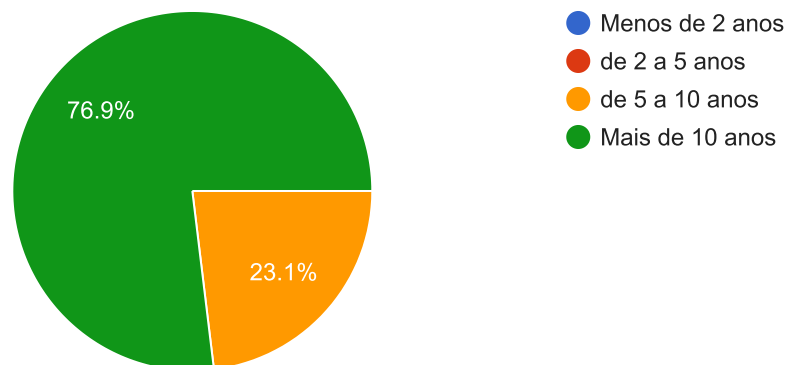
Acredito que quando a equipe de desenvolvimento do sistema entende o como aquilo será usado e para o que ele será usado, entendendo o dia a dia, as necessidades do usuário, isso contribui de forma muito positiva em como o software será desenhado e no seu resultado final, então se as duas equipes fazem um trabalho em conjunto eu só consigo ver ganhos.

Do ponto de vista de gestão de projetos, o método não tornaria as áreas de engenharia de sistemas e de engenharia de software muito dependentes? Não poderia haver situações em que um atraso em uma etapa da fase de projeto do satélite impactaria diretamente a qualidade das entregas do time de projeto de software, por exemplo? Imaginando uma situação em que a e equipe de software não está alocada no mesmo espaço da equipe de engenharia de sistemas (terceirização de serviços, por exemplo) a utilização do método deveria ser flexível o suficiente para que uma área não impacte, ou impacte minimamente as entregas da outra área. Talvez para isso seja interessante propor a atuação de um profissional que consiga transitar entre as duas áreas e tenha capacidade de atuar como um facilitador, removendo possíveis impedimentos entre as áreas.

6. Qual seu tempo de experiência com desenvolvimento de sistemas de informação?



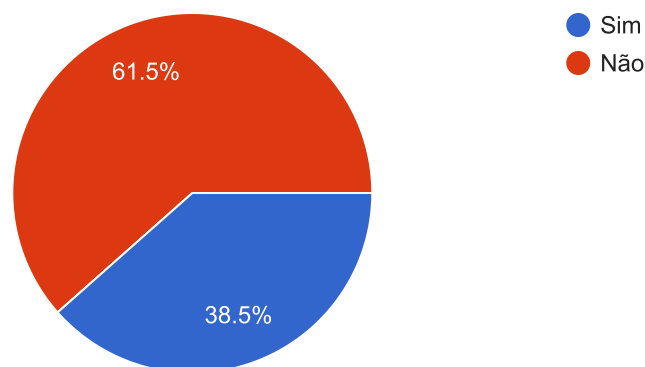
13 responses



7. Tem experiência na área espacial?



13 responses



ANEXO A - ARTIGOS PUBLICADOS

Este anexo apresenta os artigos publicados referentes a este trabalho de doutorado.

A.1 Influências da Engenharia Simultânea de Sistemas no desenvolvimento de um sistema de informação para o processo de AIT de satélites

Artigo apresentado no Congresso Brasileiro de Sistemas em outubro de 2023.

INFLUÊNCIAS DA ENGENHARIA SIMULTÂNEA DE SISTEMAS NO DESENVOLVIMENTO DE UM SISTEMA DE INFORMAÇÃO PARA O PROCESSO DE AIT DE SATÉLITES

Engenharia de Sistemas, organização, complexidade

Ana Claudia de Paula Silva

Instituto Nacional de Pesquisas Espaciais - INPE

ana.claudia@inpe.br

Geilson Loureiro

Instituto Nacional de Pesquisas Espaciais - INPE

geilson.loureiro@inpe.br

RESUMO

Embora tradicionalmente o desenvolvimento de satélites e dos sistemas de informação que apoiam os processos do seu ciclo de vida sejam realizados de forma independente, muitas das informações resultantes do esforço do engenheiro de sistemas para desenvolver o satélite também são de interesse para o desenvolvimento dos sistemas de informação. Utilizando como estudo de caso o satélite Amazonia1, neste trabalho percorremos o processo proposto pela Engenharia Simultânea de Sistemas a fim de identificar as suas influências no desenvolvimento de um sistema de informação para apoiar o processo de montagem, integração e testes de satélites. Concluímos que os artefatos da Engenharia Simultânea de Sistemas resultantes do esforço do engenheiro de sistemas para desenvolver um satélite também tem influência nos sistemas de informação, o que sugere que um trabalho colaborativo entre os dois domínios é viável e possibilita que o esforço realizado no domínio da engenharia de sistemas seja aproveitado e não repetido no domínio da engenharia de software.

1. INTRODUÇÃO

A Engenharia Simultânea de Sistemas (SCE - *Systems Concurrent Engineering*) é uma abordagem para desenvolvimento de sistemas complexos, como é o caso de sistemas espaciais.

Esse artigo analisa as influências dos artefatos gerados pela SCE na especificação de um sistema de informação para o processo de montagem, integração e testes (AIT - *Assembly, Integration and Testing*) de satélites.

Usamos como estudo de caso o desenvolvimento de um sistema de informação para apoiar o processo de AIT do satélite Amazonia 1. Amazonia 1 é um satélite de observação da Terra inteiramente projetado, montado, integrado, testado e operado pelo Brasil (INPE, 2023). O processo de AIT desse satélite aconteceu no LIT (Laboratório de Integração e Testes) do INPE (Instituto Nacional de Pesquisas Espaciais).

Quando falamos de sistemas de informação neste trabalho, falamos dos sistemas que tem como principal componente um software desenvolvido para apoiar processos de negócio das organizações.

Processo de negócio, por definição, são conjuntos de atividades ou tarefas relacionadas executadas para entregar um resultado esperado, como um produto ou um serviço (ROSING et al., 2015).

De acordo com a ESA (*European Space Agency*), projetos espaciais são normalmente divididos em 7 fases. Fases zero, A, B, e C que vão da análise da missão até o projeto detalhado e fases D, E e F, que vão da produção e qualificação, passando pela operação até o descarte do produto espacial (ESA, 2010). O processo de AIT de satélites acontece na fase D. As atividades da SCE acontecem dentro das fases zero à C, nas quais o satélite é desenvolvido.

Nas diversas etapas pelas quais um satélite passa, conjuntos de atividades são realizados para entregar um resultado esperado, ou seja, existem processos de negócio que podem se beneficiar de um sistema de informação.

Este trabalho integra duas áreas de conhecimento, a Engenharia de Sistemas Espaciais, na qual ocorre o desenvolvimento de satélites, e a Engenharia de Software, na qual ocorre o desenvolvimento de sistemas de informação.

Dentre as diversas abordagens para a engenharia de sistemas espaciais, a SCE tem uma proposta mais abrangente que aborda também o desenvolvimento das organizações que implementam os processos do ciclo de vida do produto.

Quando olhamos para o processo da SCE, vemos que muitas das informações levantadas pelo engenheiro de sistemas para desenvolver um produto espacial também são necessárias e levantadas pelo engenheiro de software, no domínio da engenharia de software, para desenvolver sistemas de informação que darão suporte aos processos do ciclo de vida desse produto. Como tradicionalmente essas atividades são realizadas de forma independente, o esforço realizado no domínio da Engenharia de Sistemas não é aproveitado e sim repetido no domínio da Engenharia de Software, o que motiva a investigação sobre a possibilidade de um trabalho conjunto entre os dois domínios.

Este trabalho é motivado pelo potencial de que o esforço realizado no domínio da Engenharia de Sistemas para desenvolver um produto espacial possa ser aproveitado no domínio da Engenharia de Software para desenvolver sistemas de informação para dar suporte aos processos do ciclo de vida desse produto.

2. ABORDAGEM

Para a realização deste trabalho a SCE foi aplicada, em nível suficiente para exemplificação, no caso do satélite Amazonia 1. Assumimos o uso da Engenharia de Sistemas Baseada em Modelos (MBSE – *Model-Based Systems Engineering*) na prática da SCE, ou seja, as atividades propostas pela SCE foram realizadas utilizando modelos.

Seguindo as atividades propostas pela SCE, foram construídos modelos do satélite Amazonia 1 e do LIT, organização que realizou o processo de AIT desse satélite.

A partir dos exemplos da aplicação da SCE, foram analisadas as influências que os artefatos gerados por essa abordagem podem exercer na especificação de um sistema de informação para apoiar o processo de AIT de satélites.

3. RESULTADOS

3.1. PROCESSO DE AIT DE SATÉLITES E RELEVÂNCIA DE UM SISTEMA DE INFORMAÇÃO

As atividades de AIT são uma sequência lógica e inter-relacionada de eventos, cujo objetivo principal é alcançar um alto grau de confiança de que o satélite está em conformidade com os parâmetros de desempenho especificados (SILVA; LOUREIRO, 2011). AIT inclui a montagem e integração dos vários subsistemas e a realização de testes funcionais e ambientais no satélite como um sistema (SILVA; LOUREIRO, 2011).

No processo de AIT, é realizado um conjunto de atividades para que um satélite montado, integrado e testado seja entregue. Embora o processo de AIT de satélites tenha uma sequência lógica e quase padronizada, cada organização o executa com suas particularidades, de acordo com sua cultura e estratégias, o que justifica o desenvolvimento de um sistema de informação customizado para dar suporte a esse processo.

Um processo típico de AIT de satélites está representado na Figura 1. As partes do satélite são recebidas, são feitas atividades de montagem mecânica, integração e testes elétricos, até que o satélite esteja pronto para os testes ambientais. Após os testes ambientais, inicia-se a campanha de lançamento.

Cada fase desse processo inclui uma série de atividades que envolvem diversos equipamentos, pessoas, tarefas e informações, sendo, portanto, um processo complexo. Durante esse processo, várias informações são geradas e requeridas.

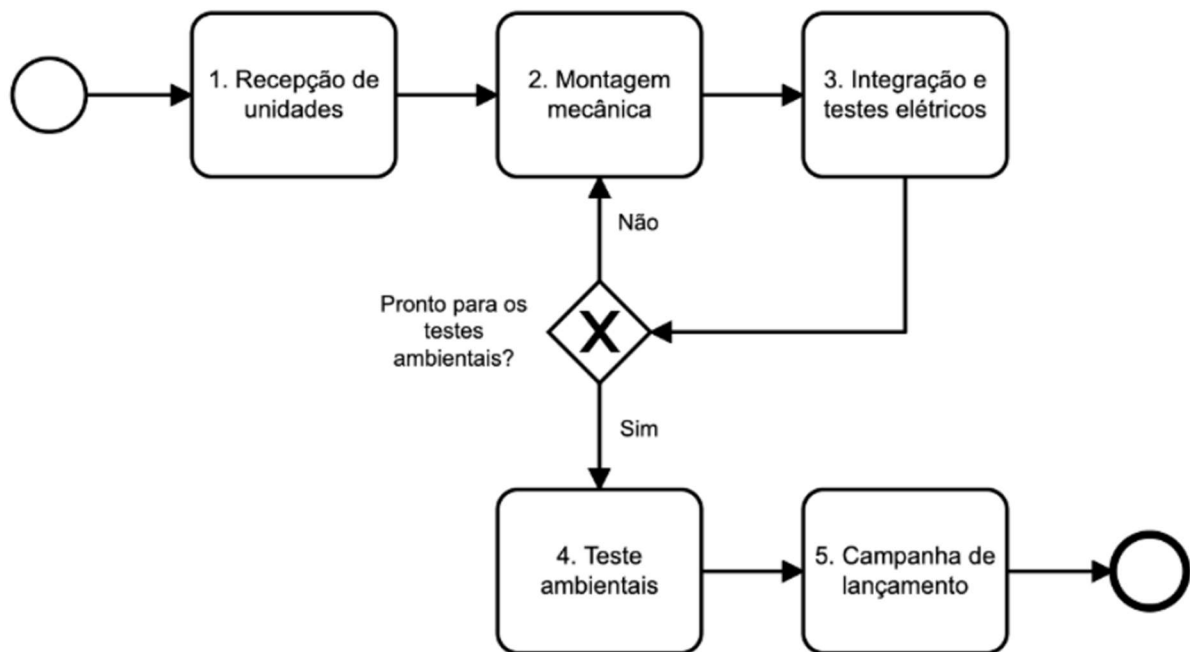


Figura 1- Típico processo de AIT de satélites

Fonte: Silva (2023)

No LIT/INPE, por exemplo, muitas dessas informações estão contidas em documentos. A equipe de AIT utiliza uma série de ferramentas para ajudá-la a gerenciar esses documentos, as diversas tarefas que devem realizar e todas as informações que fluem por todo o processo de AIT. Essas ferramentas incluem sistemas de gerenciamento de documentos, planilhas e banco de dados locais. São ferramentas desconectadas, dedicadas a pontos específicos do processo, o que dificulta o acesso à informação e exige grande esforço da equipe para manter a consistência das informações.

O papel do sistema de informação é apoiar o processo de AIT armazenando, processando e provendo essas informações, com as vantagens de eliminar retrabalho, reduzir inconsistências, facilitar o compartilhamento de informações e criar uma base de conhecimento que estará disponível para a organização independentemente da rotatividade da equipe.

3.2. ENGENHARIA SIMULTÂNEA DE SISTEMAS (SCE)

A SCE, representada na Figura 2, é uma abordagem para engenharia de sistemas complexos que antecipa os requisitos dos processos do ciclo de vida para os estágios iniciais do desenvolvimento do produto, e preocupa-se não somente com desenvolvimento do produto, mas também com o desenvolvimento das organizações que executam os processos do ciclo de vida desse produto (LOUREIRO et al., 2018).



Figura 2 - Framework da SCE

Fonte: Traduzida e adaptada de Loureiro et al. (2018)

A dimensão de análise inclui os subprocessos do processo de Engenharia de Sistemas que são aplicados aos elementos de cada camada da estrutura de decomposição do produto e da organização.

A dimensão de integração contém os elementos que devem ser integrados na solução sistema (elementos do produto e da organização).

Com base em uma declaração de missão, a SCE propõe que sejam identificados os processos do ciclo de vida do sistema de interesse. Processos podem ser desdobrados em cenários. As análises da dimensão de análise são realizadas para cada processo ou cenário.

3.3. INFLUÊNCIAS DA SCE NO DESENVOLVIMENTO DE UM SISTEMA DE INFORMAÇÃO PARA O PROCESSO DE AIT DO SATÉLITE AMAZONIA1

A seguir são apontadas influências das análises propostas pela SCE no desenvolvimento de um sistema de informação para apoiar o processo de AIT do satélite Amazonia 1.

ANÁLISES DE STAKEHOLDERS E DE REQUISITOS

Na Análise de *Stakeholders*, são identificados os *stakeholders* do sistema espacial e seus *concerns* (desejos, vontades, objetivos, interesses, preocupações). *Concerns* são desdobrados em necessidades e as necessidades em requisitos de *stakeholder*. A Figura 3 ilustra esse procedimento para o cenário de Ensaios Ambientais do Amazonia 1.

O cenário de Ensaios Ambientais é um desdobramento do processo de AIT. Nesse cenário o satélite é exposto a condições controladas em laboratório com o objetivo de verificar e garantir a sua capacidade de suportar as condições ambientais em todas as fases de sua vida útil, desde o momento do lançamento até o término previsto de sua operação em órbita (INPE/LIT, 2022). Entre os ensaios realizados nesse cenário estão os ensaios de vibração, que são referenciados na Figura 3.

Dos requisitos de *stakeholder* são derivados os requisitos de sistema.

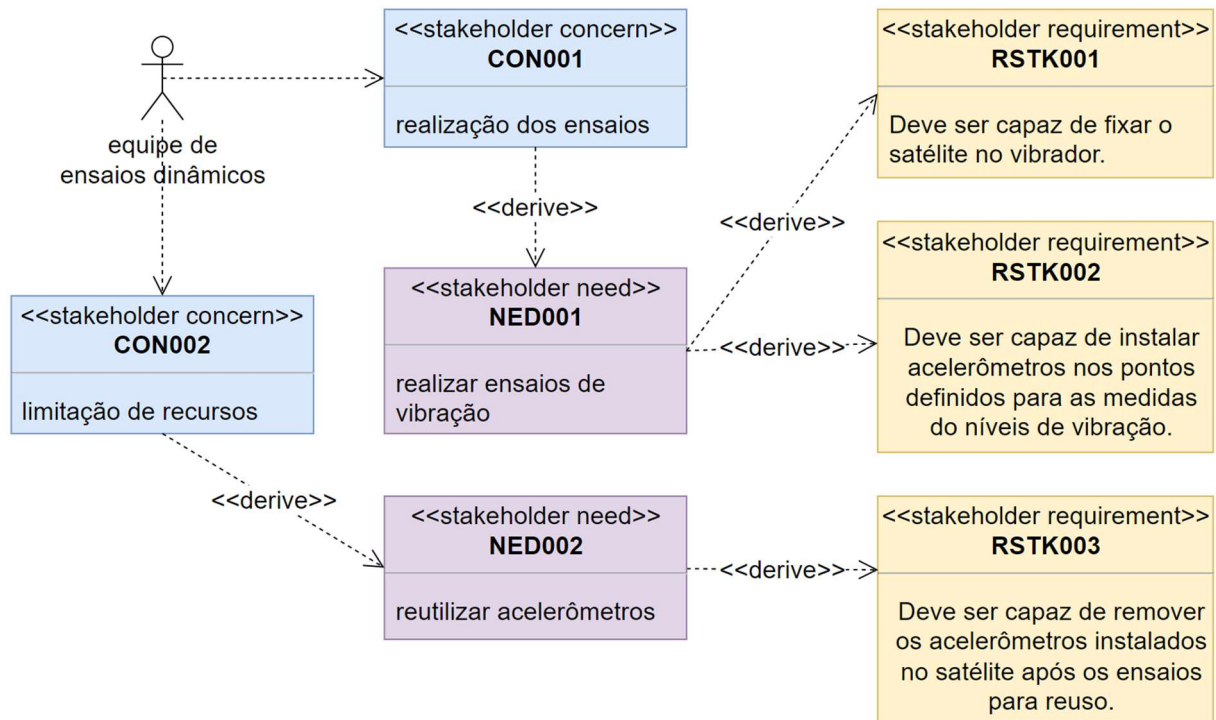


Figura 3- Desdobramento de *concerns* em requisitos de stakeholder

Os *concerns*, necessidades e requisitos relacionados ao sistema espacial, expressos nesses modelos, também influenciam o sistema de informação.

Por exemplo, para o requisito RSTK001, o engenheiro de software entende que procedimentos para fixação do satélite sobre a mesa vibratória deverão estar disponíveis no sistema de informação. Já os requisitos RSTK002 e RSTK003 sinalizam para o engenheiro de software que funcionalidades para controle da instalação e remoção de acelerômetros devem estar disponíveis no sistema de informação.

ANÁLISE FUNCIONAL

A análise funcional resulta em uma descrição da estrutura funcional e do comportamento funcional requeridos para o sistema. A estrutura revela uma visão estática das funções do sistema, enquanto o comportamento está preocupado com a execução temporal, lógica, causal e condicional das funções.

A Figura 4 ilustra a estrutura funcional da organização, focando no fluxo de informações de alto nível entre as macro funções. Esses modelos expressam para o engenheiro de software a necessidade de informação dos atores do processo. O sistema de informação deve ser moldado para que essas informações estejam disponíveis nele.

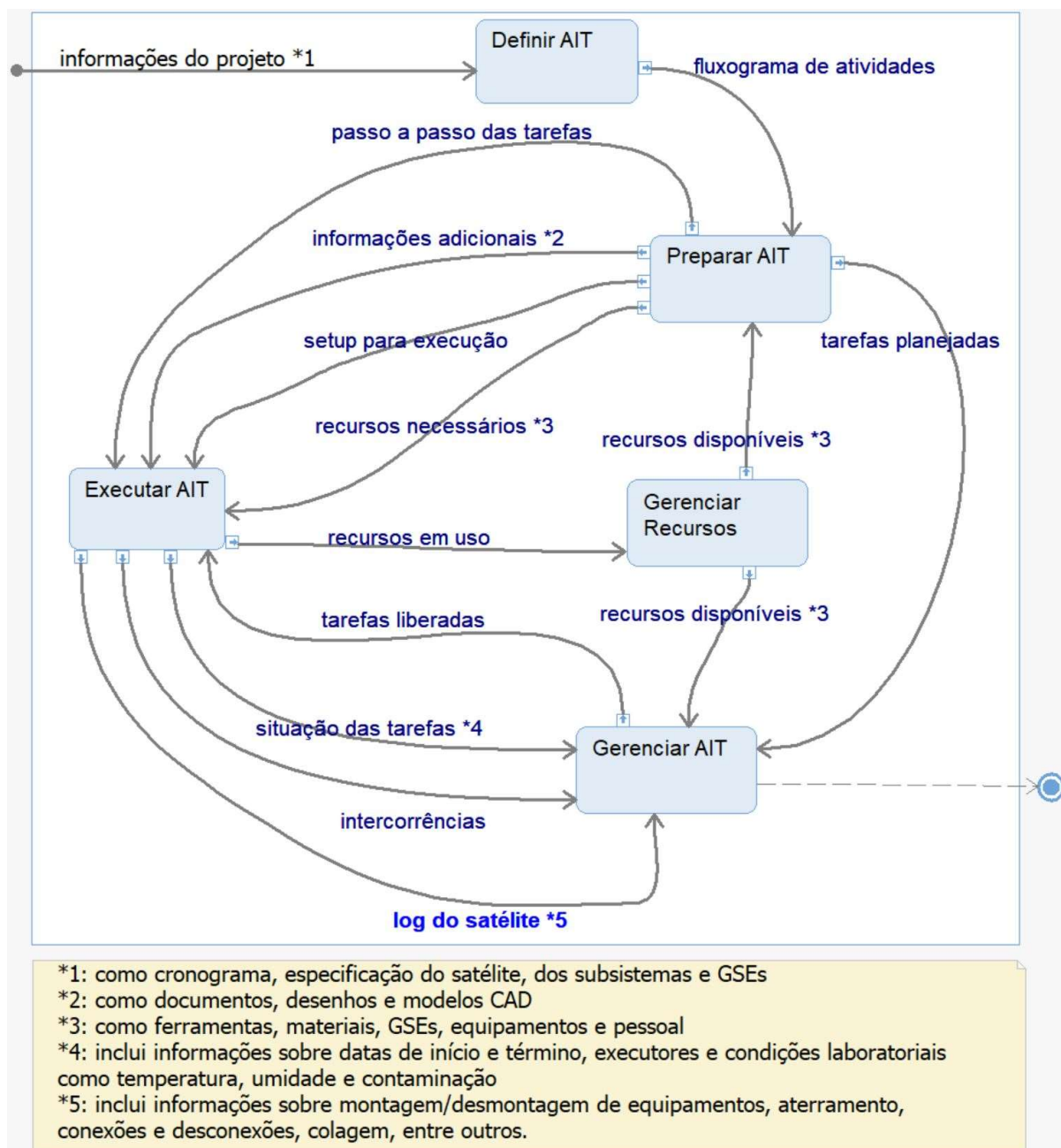


Figura 4 – Modelo da estrutura funcional da organização

Por exemplo, olhando para o fluxo “log do satélite”, o engenheiro de software consegue identificar informações que devem ser disponibilizadas pelo sistema de informação, como equipamentos montados no satélite numa determinada data, quantidade de conexões de um dado conector, histórico de montagens/desmontagens, histórico de conexões/desconexões.

A Figura 5 ilustra uma pequena parte do comportamento funcional da organização. Modelos do comportamento funcional expressam a forma como os processos são executados na organização no que diz respeito à sua execução temporal, lógica, causal e condicional. O comportamento funcional da organização, além de apontar atividades a serem apoiadas pelo sistema de informação, aponta também regras de negócio a serem implementadas.

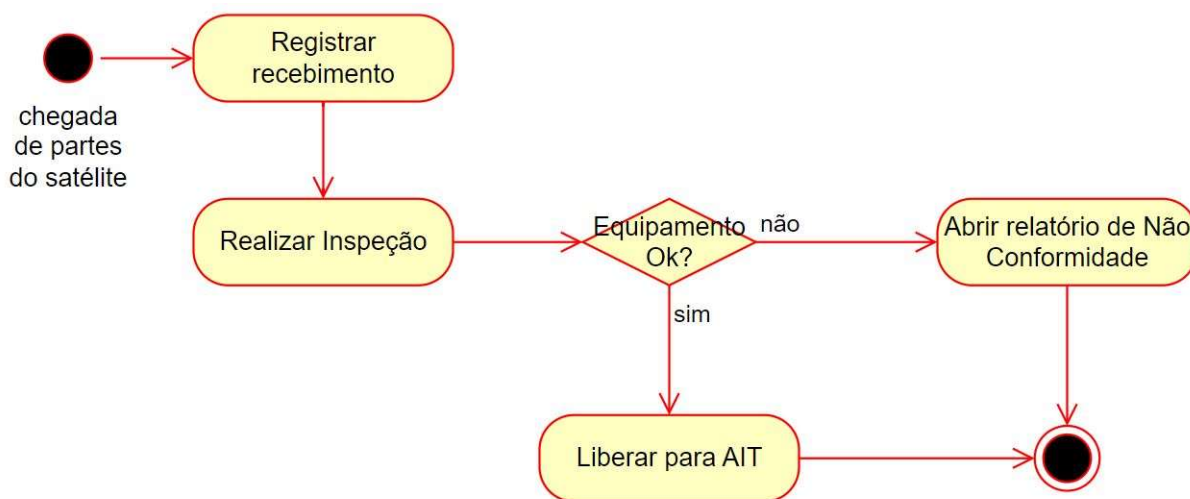


Figura 5 – Modelo do comportamento funcional da organização

Um exemplo de regra de negócio identificada no modelo de comportamento da organização ilustrado na Figura 5 e de sua implementação no sistema de informação:

- Regra de negócio: equipamento não pode ser montado no satélite se não tiver sido aprovado na inspeção.

- Implementação: sistema alerta se o equipamento informado durante registro de atividade não foi aprovado em inspeção.

Quanto a análise funcional do produto, ela tem pouco potencial de influenciar o sistema de informação, já que o funcionamento do produto não está na área de atuação do sistema de informação.

ANÁLISE DE IMPLEMENTAÇÃO

Na análise de implementação, os requisitos de sistema são alocados para elementos físicos (como equipamentos e elementos de infraestrutura), lógicos (como softwares) ou organizacionais (como equipes de trabalho), resultando em uma arquitetura de implementação.

Os modelos resultantes dessa análise descrevem os itens necessários para desempenhar as funções do sistema.

No caso do satélite, subsistemas, componentes e softwares destinados a desempenhar as funções definidas nos modelos funcionais do produto.

No caso da organização, tanto a estrutura física (prédios, salas, câmaras etc.), quanto a estrutura organizacional (departamentos, equipes etc.) podem ser modeladas.

A estrutura do produto influencia o sistema de informação, pois pode ser referenciada nele. No caso de AIT de satélites será referenciada para registrar ações como recebimento de partes, montagens, conexões etc. A estrutura física da organização também influencia, pois os espaços físicos e as câmaras de teste são itens que terão seu uso gerenciado no sistema de informação.

Esses modelos contribuem para a definição da estrutura de dados do sistema de informação e podem ser fonte de entrada para popular a base de dados.

A estrutura organizacional (como equipes designadas para desempenhar as funções), além de também contribuir com a estrutura e base de dados, aponta atores, papéis e informações relevantes para o controle de acesso ao sistema de informação.

4. DISCUSSÃO E CONCLUSÃO

Tanto na Engenharia de Sistemas, como na Engenharia de Software, existe um processo de compreensão e definição dos requisitos do sistema. O Conselho Internacional de Engenharia de Sistemas (INCOSE), destaca que esse processo constitui um esforço significativo dentro do processo do desenvolvimento do sistema (INCOSE, 2015).

No caso do desenvolvimento de satélites, no domínio da Engenharia de Sistemas esse esforço inclui a interação com os diversos *stakeholders* do sistema espacial para entendimento das suas necessidades e expectativas quanto ao produto a ser desenvolvido. O engenheiro de sistemas encarregado de desenvolver um satélite realiza uma série de atividades para entendimento dos requisitos, um esforço do qual as informações resultantes também interessam ao engenheiro de software, como foi demonstrado neste trabalho.

O fato é que o resultado do esforço realizado para a engenharia do sistema espacial tem muito significado também para a especificação dos sistemas de informação. Trabalhando isoladamente, o engenheiro de software despenderá um esforço também significativo para explorar os mesmos processos e cenários que são explorados no domínio da Engenharia de Sistemas, isso sinaliza a viabilidade de um trabalho conjunto entre os dois domínios.

Quanto as abordagens de engenharia de sistemas que elegemos para esta pesquisa, SCE e MBSE, mostraram-se apropriadas para o contexto deste trabalho pelos seguintes motivos:

- Na SCE os processos do ciclo de vida do produto são analisados desde o início do projeto, fornecendo informações em tempo hábil para o desenvolvimento de sistemas de informação que irão dar apoio a esses processos;
- A SCE aborda também o desenvolvimento das organizações que implementam os processos do ciclo de vida do produto, o que a torna uma fonte de informação mais

rica em comparação com as abordagens tradicionais de engenharia de sistemas que abordam apenas o desenvolvimento do produto.

- Quanto a MBSE, além de ser uma tendência para o desenvolvimento de sistemas espaciais (SILVA; LOUREIRO, 2020), o uso de modelos é uma prática familiar aos engenheiros de software, o que também favorece a integração entre os domínios da engenharia de sistemas e da engenharia de software.

Concluimos que os artefatos da Engenharia Simultânea de Sistemas resultantes do esforço do engenheiro de sistemas para desenvolver um satélite também contribuem com o engenheiro de software que trabalha para desenvolver sistemas de informação para apoiar processos do ciclo de vida desse satélite. Isso sugere que um trabalho simultâneo e colaborativo entre os dois domínios é viável e possibilita que o esforço realizado no domínio da engenharia de sistemas seja aproveitado e não repetido no domínio da engenharia de software.

5. AGRADECIMENTOS

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

REFERÊNCIAS BIBLIOGRÁFICAS

ESA. **ECSS-M-ST-10C Rev.1: Space project management - Project planning and implementation**. Noordwijk. 2010.

INCOSE. **Incase Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities**. 4th. ed. [S.I.]: Wiley, 2015.

INPE. **INPE / Amazon Mission**. Disponível em: <<http://www.inpe.br/amazonia1/en/>>. Acesso em: 22 ago. 2023.

INPE/LIT. **Ensaio Dinâmico**. 2022. Disponível em: <<http://www.lit.inpe.br/pt-br/vibracao>>. Acesso em: 13 jul.2022.

LOUREIRO, G.; PANADES, W. F.; SILVA, A. Lessons learned in 20 years of application of Systems Concurrent Engineering to space products. **Acta Astronautica**, v. 151, p. 44–52, out. 2018.

ROSING, M. v.; SCHEER, A.-W.; SCHEEL, H. v. **The complete business process handbook: body of knowledge from process modeling to BPM. Volume I**. Waltham, Ma: Elsevier, 2015. ISBN 978-0127999593.

SILVA, A. C.; LOUREIRO, G. Integrated development of space systems – design for ait - design for assembly, integration and testing of satellites - d4ait. In: **2011 IEEE International Conference on Industrial Engineering and Engineering Management**. [S.l.: s.n.], 2011. p. 590–594

SILVA, A. C. d. P.; LOUREIRO, G. Iniciativas de adesão à mbse na área espacial. In: RODRIGUES, A. C.; BARBOSA, A. L.; TEIXEIRA, A. F.; BATISTA, C. L. G.; OLIVEIRA, C. M. d.; CUELLAR, D. A. (Ed.). **Anais... do 11º Workshop em Engenharia e Tecnologia Espaciais**. São José dos Campos: Instituto Nacional de Pesquisas Espaciais (INPE), 2020. ISSN 2177-3114.

SILVA, L. A. **TEPROM - FRAMEWORK DE ENGENHARIA SIMULTÂNEA DE SISTEMAS BASEADO EM MODELOS PARA TESTES FUNCIONAIS ELÉTRICOS EM SATÉLITES**. Tese (Doutorado em Engenharia e Gerenciamento de Sistemas Espaciais) — Instituto Nacional de Pesquisas Espaciais (INPE), 2023.

A.2 Estratégia de integração de MBSE com sistemas de informação que apoiam processos de ciclo de vida de satélites

Artigo apresentado no Congresso Brasileiro de Sistemas em outubro de 2023.

ESTRATÉGIA DE INTEGRAÇÃO DE MBSE COM SISTEMAS DE INFORMAÇÃO QUE APOIAM PROCESSOS DE CICLO DE VIDA DE SATÉLITES

Engenharia de Sistemas, organização, complexidade

Ana Claudia de Paula Silva

Instituto Nacional de Pesquisas Espaciais - INPE

ana.claudia@inpe.br

Geilson Loureiro

Instituto Nacional de Pesquisas Espaciais - INPE

geilson.loureiro@inpe.br

RESUMO

Este trabalho apresenta uma estratégia para integração automatizada de modelos da Engenharia de Sistemas Baseada em Modelos (MBSE) com sistemas de informação que apoiam processos de ciclo de vida de satélites. Os modelos de engenharia construídos nas fases iniciais de desenvolvimento do satélite são fonte de informação para os processos das fases posteriores. Dessa forma, é de interesse que informações contidas nesses modelos estejam disponíveis nos sistemas de informação. Dentro desse contexto, a integração automatizada tem um papel facilitador. O trabalho foi demonstrado com a integração de modelos do satélite Amazonia 1 com um sistema de informação para apoio ao processo de montagem, integração e testes de satélites. A estratégia de integração proposta neste trabalho consiste na definição de uma convenção de modelagem e de um mapeamento entre os modelos da MBSE e dos sistemas de informação. A convenção e o mapeamento viabilizam a integração automatizada entre os modelos. Concluímos que a estratégia apresentada aponta um caminho possível para integração automatizada de modelos da MBSE com sistemas de informação que apoiam processos do ciclo de vida de satélites.

1. INTRODUÇÃO

A Engenharia de Sistemas baseada em modelos (MBSE) é uma opção para desenvolver sistemas complexos, como satélites, em substituição à abordagem tradicional baseada em documentos.

Na MBSE, modelos são a principal fonte de informação e o principal meio de troca de informação entre os envolvidos no processo de engenharia, substituindo nessas funções os documentos utilizados na abordagem tradicional de engenharia de sistemas (FRIEDENTHAL et al., 2014). Modelos são construídos para representar requisitos, estrutura e comportamento dos sistemas (FRIEDENTHAL et al., 2014).

Friedenthal et al. (2014) cita como benefícios da MBSE: a melhoria na comunicação, a redução no risco de desenvolvimento, o aprimoramento da qualidade, o aumento da produtividade e a melhoria na transferência de conhecimento.

Este trabalho apresenta uma estratégia para integração automatizada de modelos da MBSE com sistemas de informação que apoiam processos de ciclo de vida de satélites.

Os processos do ciclo de vida de um satélite vão da sua concepção até o seu descarte. Considerando uma abordagem baseada em modelos para a engenharia do satélite, os modelos construídos nas fases de desenvolvimento do produto são fonte de informação para os processos de todas as fases posteriores. Dessa forma, é de interesse que informações contidas nos modelos de engenharia estejam disponíveis nos sistemas de informação que apoiam os processos do ciclo de vida de satélites. Por essa razão uma integração entre os modelos do satélite e os sistemas de informação mostra-se desejável.

A estratégia de integração proposta neste trabalho é demonstrada com o caso de um sistema de informação em desenvolvimento no Laboratório de Integração e Testes (LIT) do Instituto Nacional de Pesquisas Espaciais (INPE) que tem o objetivo de apoiar o processo de AIT (*Assembly, Integration and Testing* ou Montagem, Integração e Testes) de satélites.

Muitas informações importantes para AIT vem do processo de desenvolvimento do satélite. Sendo esse processo baseado em modelos, torna-se viável buscar uma integração que possibilite que essas informações sejam automaticamente inseridas no sistema de informação.

Embora até agora os satélites que passaram pelo LIT tenham sido desenvolvidos com a abordagem baseada em documentos, há uma tendência na Engenharia de Sistemas Espaciais de migração para a MBSE. Grandes atores da área espacial têm investido nessa abordagem, como NASA, ESA, Airbus, Thales Alenia Space e o próprio INPE (SILVA; LOUREIRO, 2020).

Os benefícios que a adesão a MBSE e a integração dos modelos com o sistema de informação podem trazer para o processo de AIT e, possivelmente, para outros processos do ciclo de vida de satélites, são a motivação para este trabalho.

2. ABORDAGEM

Com base em documentos do satélite Amazonia 1 (INPE, 2022), que foi integrado e testado no LIT/INPE, foram construídos modelos exemplificando a abordagem MBSE para esse satélite. A ferramenta utilizada para a construção dos modelos foi a IBM Rhapsody (IBM, 2023).

Esses modelos foram comparados com o modelo de dados do sistema de informação, em desenvolvimento, que visa apoiar o processo de AIT de satélites do LIT.

A comparação dos modelos foi realizada com a intenção de identificar as informações dos modelos do satélite que são de interesse do sistema de informação, ou seja, informações que constam nos modelos do satélite e que também são previstas de serem armazenadas no sistema de informação.

Identificadas as informações de interesse, foi desenvolvida uma estratégia para integração dos modelos. Definida a estratégia, foi desenvolvido um código em linguagem groovy (APACHE, 2023) para automatizar a integração.

3. RESULTADOS

O processo de montagem, integração e testes (AIT na sigla em inglês) realizado no LIT/INPE pode ser resumido em cinco macro subprocessos (INPE/LIT, 2017):

Definir AIT: são definidas as atividades que serão executadas (desde as macro atividades até o nível de procedimento) e a sequência em que essas atividades devem ser executadas.

Preparar AIT: detalham-se as atividades, passo a passo, identificando e apontando os recursos necessários aos operadores para executá-las, como *setup*, equipamentos do satélite e GSEs (*Ground Support Equipment*).

Gerenciar AIT: o gerente de AIT organiza a execução das atividades, elabora o cronograma, libera tarefas para execução e trata das questões e problemas que ocorrem durante o processo.

Executar AIT: as atividades definidas são efetivamente realizadas e sua realização é documentada.

Gerenciar recursos: os recursos necessários para o processo de AIT, como materiais, ferramentas, pessoal, GSEs, são gerenciados e mantidos em condições de uso.

Para apoiar esse processo de ponta a ponta, está em desenvolvimento no LIT um sistema de informação que visa prover informações confiáveis para tomada de decisão e ser base de conhecimento para AITs futuros.

A estratégia de integração proposta será exemplificada com a integração desse sistema de informação com modelos do satélite Amazonia 1 que foi integrado e testado no LIT.

A linguagem utilizada nos modelos do sistema de informação é a UML. UML é uma linguagem utilizada para modelar softwares baseados no paradigma da orientação a objetos (OMG, 2017). Neste trabalho é apresentado um diagrama de Classes da UML. Uma Classe é um modelo para instanciar objetos (OMG, 2017).

A linguagem dos modelos do satélite utilizado como estudo de caso é a SysML. SysML é uma linguagem de modelagem gráfica de uso geral que pode ser base para diferentes métodos da MBSE (INCOSE, 2015).

Neste trabalho são apresentados um Diagrama de Definição de Blocos (BDD) e um Diagrama de Blocos Internos (IBD) da SysML. Esses são diagramas utilizados para representar a estrutura do sistema (FRIEDENTHAL et al., 2009). O bloco é a unidade básica de estrutura em SysML e pode ser usado para representar hardware, software, instalações, pessoal ou qualquer outro elemento do sistema. (FRIEDENTHAL et al., 2009).

Um BDD foca na definição e descreve a hierarquia do sistema (IBM, 2012). O IBD foca no uso (FRIEDENTHAL et al., 2014) e descreve a estrutura interna de um sistema em termos de suas partes, portas e conectores. Enquanto um Bloco é a definição de um elemento, o uso ou instância desse elemento é chamado Parte (IBM, 2012).

3.1. CLASSES DO SISTEMA DE INFORMAÇÃO

A Figura 1 mostra algumas Classes do sistema de informação com alguns dos seus atributos.

As classes referenciadas para demonstração da estratégia de integração são explicadas a seguir.

SatelliteModel: representa modelos do satélite (como modelo elétrico e modelo de voo, não confundir com modelos da MBSE).

Subsystem: representa subsistemas do satélite.

EquipmentType: representa tipos de equipamento que compõe o satélite, como roda de reação ou sensor solar.

SatelliteEquipment: representa um tipo de equipamento numa posição específica do satélite, como uma roda de reação na posição x ou y.

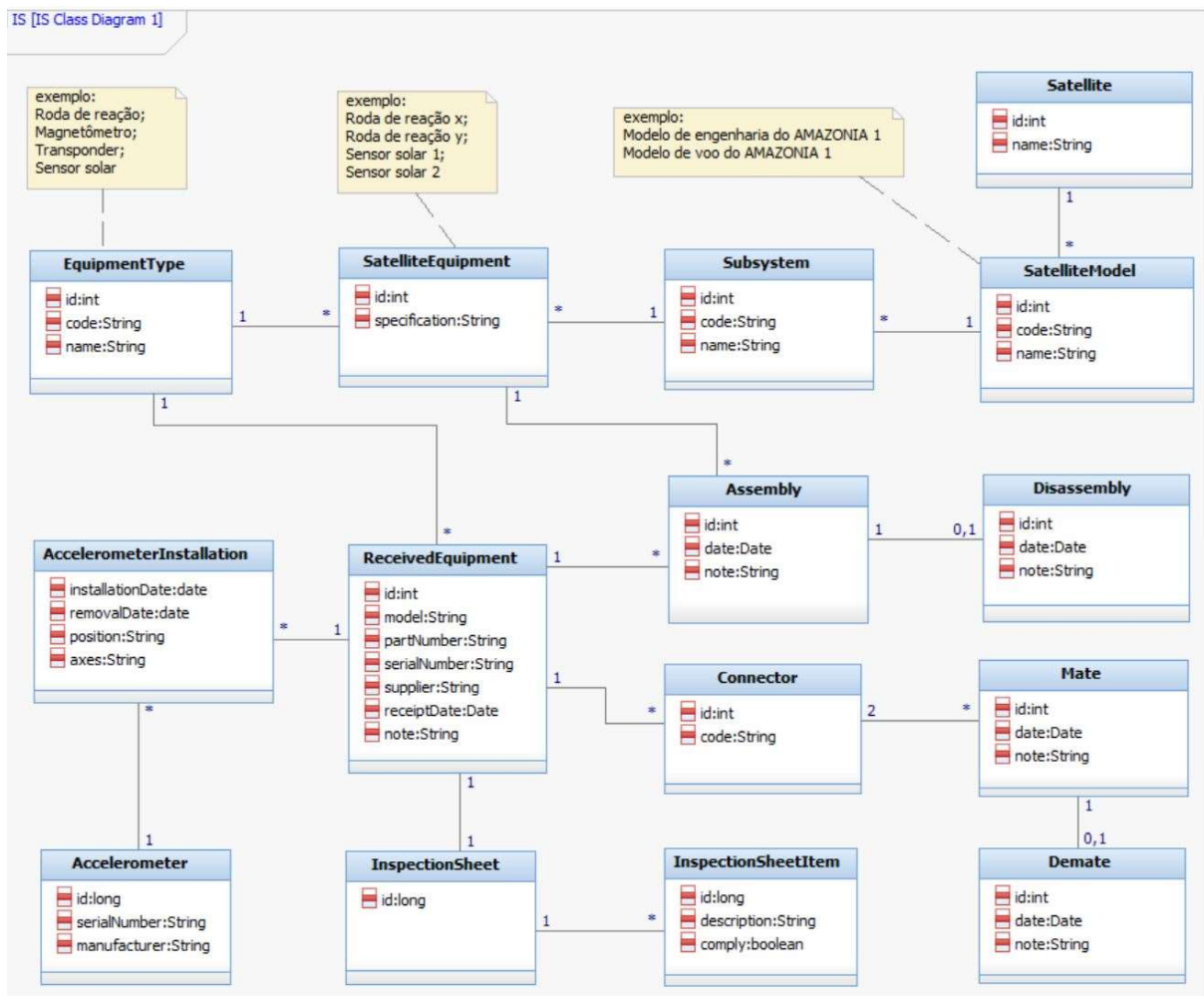


Figura 1- Diagrama de Classes do sistema de informação

3.2. MODELOS DO SATÉLITE

Uma pequena parte do modelo de arquitetura de implementação do satélite está representada nos diagramas das Figura 2 e Figura 3.

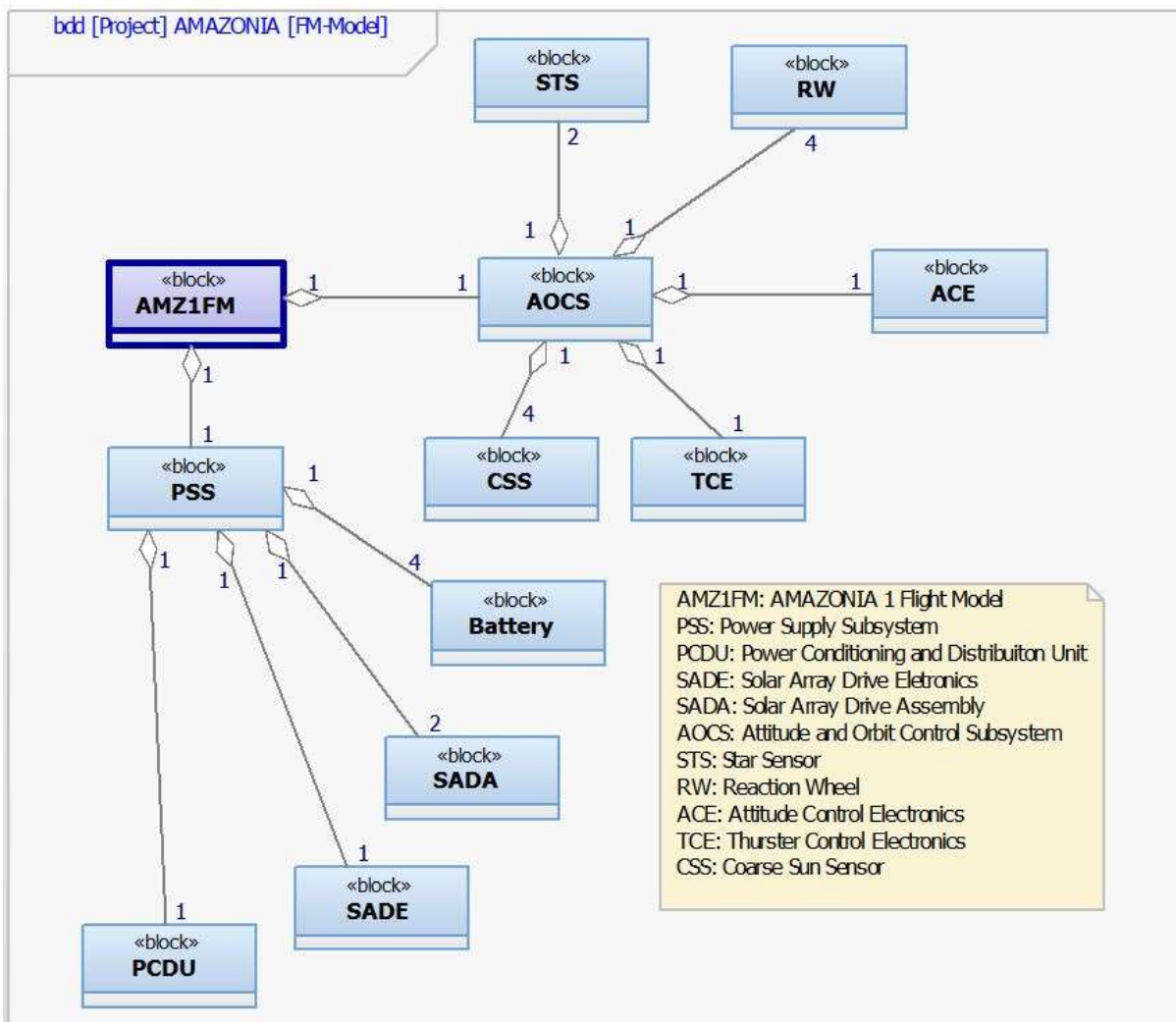


Figura 2 – BDD simplificado do satélite

No BDD da Figura 2 o modelo de voo do satélite (Bloco AMZ1FM) é representado como uma composição dos seus subsistemas e seus subsistemas são representados como uma composição de seus equipamentos. O diagrama apresenta dois subsistemas do satélite: PSS (*Power Supply Subsystem* ou subsistema de suprimento de energia) e AOCS (*Attitude and Orbit Control Subsystem* ou subsistema de controle de atitude e órbita) e alguns dos equipamentos que compõe esses subsistemas, como RW (*Reaction Wheel* ou roda de reação) e ACE (*Attitude Control Electronics* ou Eletrônica de controle de atitude).

De acordo com a notação SysML, no BDD as caixas representam os Blocos; as linhas com um losango em uma extremidade conectando os Blocos representam uma Associação de Partes (que são comumente usadas para decompor um Bloco em suas Partes) e os números na outra extremidade das linhas representam a quantidade de instâncias do Bloco relacionado. Por exemplo, o subsistema PSS é decomposto em 1 PCDU, 1 SADE, 2 SADA e 4 baterias.

A Figura 3 mostra instâncias do Blocos RW e ACE em um IBD do subsistema AOCS. No IBD as caixas representam as Partes e em seu primeiro compartimento aparecem o nome da Parte e o Bloco do qual a Parte é um tipo, no formato: Nome da Parte: Nome do tipo.



Figura 3 - IBD simplificado de um subsistema do satélite

3.3. ESTRATÉGIA DE INTEGRAÇÃO

Podemos observar que a estrutura do satélite é modelada nos diagramas estruturais do satélite, BDD (Figura 2) e IBD (Figura 3), e também no Diagrama de Classes (Figura 1) do sistema de informação, mas de maneiras diferentes.

Os elementos do BDD e IBD não correspondem diretamente aos elementos do Diagrama de Classes (um Bloco não corresponde a uma Classe), mas existe um relacionamento entre os modelos.

Elementos do BDD e IBD correspondem a instâncias das Classes do sistema de informação. O Bloco AMZ1FM corresponde a uma instância da classe *SatelliteModel*. Os Blocos dos subsistemas (AOCS e PSS) correspondem a instâncias da Classe *Subsystem*. Os Blocos que aparecem na decomposição dos subsistemas correspondem a instâncias da classe *EquipmentType*.

Instâncias da Classe *SatelliteEquipment* não aparecem no BDD, mas aparecem no IBD dos Blocos de subsistema.

Embora esses modelos do satélite e o modelo de Classes não tenham relação direta, é possível definir uma estratégia de integração. A estratégia apresentada neste trabalho inclui uma convenção de modelagem e um mapeamento entre os modelos.

CONVENÇÃO DE MODELAGEM

Deve ser definida uma convenção de modelagem para a construção dos modelos do satélite que possibilite a identificação dos elementos de interesse do sistema de informação.

Neste trabalho são utilizados estereótipos da SysML para implementar o link entre os modelos. Os estereótipos permitem classificar elementos com “algo em comum”. Com este recurso, podemos classificar os elementos dos modelos SysML, identificando a qual Classe do sistema de informação eles estão relacionados.

Foi definido que os estereótipos seriam iniciados com o prefixo “is_” (referência a *Information System*) para indicar que os elementos com eles classificados são de interesse do sistema de informação. A Figura 4 mostra IBDs com Partes classificadas com estereótipos. As propriedades, ou atributos, dos elementos também são apontadas nessa figura porque são referenciadas na Tabela 1.

Os seguintes estereótipos foram utilizados: <<is_satellite_equipment>>, <<is_satellite_model>> e <<is_subsystem>>.

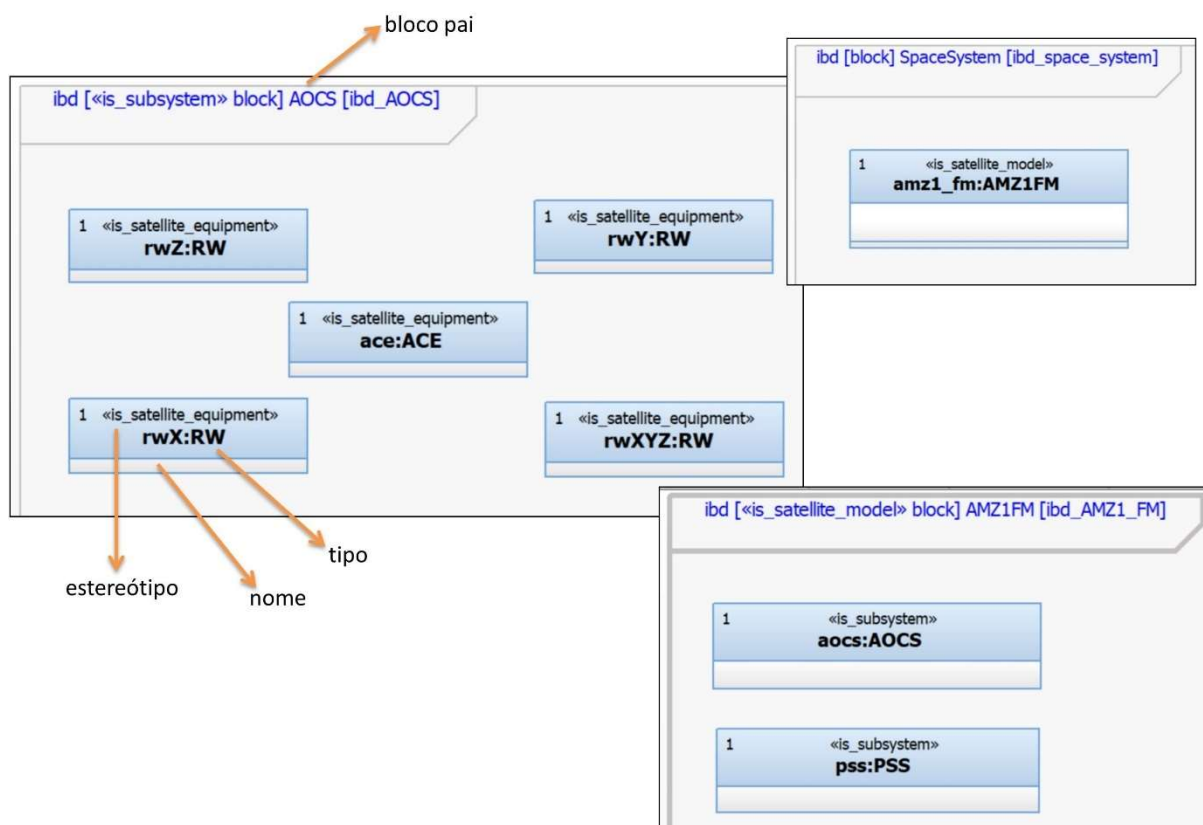


Figura 4 - IBDs com Partes classificadas com estereótipos

MAPEAMENTO

Definida a convenção, é necessário mapear o destino de cada informação do modelo SysML que interessa ao sistema de informação, conforme exemplificado na Tabela 1, na qual é feito um mapeamento para leitura de IBDs para popular Classes do sistema de informação.

Nessa tabela, na coluna "Atributo da Parte", aparecem os nomes dos atributos apontados na Figura 4 e os termos entre parênteses correspondem ao nome dos atributos na ferramenta Rhapsody (IBM, 2023), que foi utilizada para criar os modelos SysML.

De acordo com o mapeamento apresentado na Tabela 1, os elementos do modelo do satélite identificados com o estereótipo “is_subsystem” devem ser alocados na Classe Subsystem do sistema de informação, sendo que, o valor do atributo “nome” do elemento será atribuído ao atributo “code” e o valor do atributo “bloco pai” será atribuído ao atributo “satelliteModel” dessa Classe.

Os elementos do modelo do satélite identificados com o estereótipo “is_satellite_equipment” devem ser alocados nas classes EquipmentType e SatelliteEquipment. Sendo que, o valor do atributo “tipo” do elemento será atribuído ao atributo “code” da Classe EquipmentType. O valor do atributo “nome” será atribuído ao atributo “specification” da Classe SatelliteEquipment, o valor do atributo “tipo” será atribuído ao atributo “equipmentType” e o valor do atributo “bloco pai” será atribuído ao atributo “subsystem” dessa mesma classe.

Os elementos do modelo do satélite identificados com o estereótipo “is_satellite_model” devem ser alocados na Classe SatelliteModel, sendo que o atributo “nome” do elemento será atribuído ao atributo “code” dessa Classe.

Tabela 1- Mapeamento

Estereótipo (no modelo SysML)	Classe (no modelo do sistema de informação)	Atributo da Parte (no modelo SysML)	Atributo da Classe (no modelo do sistema de informação)
is_subsystem	Sybsystem	nome (name)	code
		bloco pai (ofClass.name)	satelliteModel
is_satellite_equipment	EquipmentType	tipo (otherClass.name)	code
	SatelliteEquipment	nome (name)	specification
		tipo (otherClass.name)	equipmentType
bloco pai (ofClass.name)	subsystem		
is_satellite_model	SatelliteModel	nome (name)	code

3.4. INTEGRAÇÃO AUTOMATIZADA

A ferramenta Rhapsody possui uma biblioteca, chamada JAVA API (IBM, 2009), que permite ler e escrever nos modelos. Utilizando essa biblioteca, foi criado um script na linguagem groovy (APACHE, 2023) para ler os modelos SysML, interpretá-los de acordo com o mapeamento definido e gerar entradas para o sistema de informação a partir dos modelos do satélite.

A Figura 5 apresenta o código escrito em groovy que faz uso da biblioteca JAVA API para a leitura dos modelos do Rhapsody. A Figura 6 apresenta o código groovy que cria objetos, a partir dos elementos do Rhapsody, para popular o banco de dados do sistema de informação.

4. DISCUSSÃO E CONCLUSÃO

Este trabalho apresentou uma proposta de convenção de modelagem e mapeamento que possibilitam a integração dos modelos de engenharia de um satélite com modelos de um sistema de informação que apoia um processo do ciclo de vida desse satélite e demonstrou ser viável a automatização dessa integração via software.

Embora a estratégia de integração automatizada apresentada tenha sido demonstrada com um caso de desenvolvimento de um satélite e de um sistema de informação específicos, ela pode ser aplicada a outros casos que envolvam MBSE e sistemas de informação para apoio a processos do ciclo de vida de satélites.

Visto que informações contidas nos modelos do satélite são de interesse dos sistemas de informação que apoiam os processos do seu ciclo de vida, a integração proposta neste trabalho mostra-se um facilitador para o compartilhamento das informações, criando um vínculo que minimiza retrabalho e inconsistências, já que informações dos modelos do satélite podem ser automaticamente inseridas no sistema de informação e mudanças podem ser refletidas automaticamente.

O padrão da convenção de modelagem a ser utilizada e do mapeamento vai depender da linguagem de modelagem escolhida para a engenharia do satélite e da linguagem de modelagem e paradigma adotados para o sistema de informação.

Para a integração automatizada é necessário que sejam utilizadas ferramentas de modelagem que ofereçam recursos que possibilitem a leitura dos modelos. É preciso destacar que atualmente as ferramentas para a prática da MBSE ainda são limitadas e não oferecem muitas facilidades, fator que pode trazer certa dificuldade para a aplicação prática da estratégia proposta.

De qualquer forma, concluímos que a estratégia apresentada neste trabalho aponta um caminho possível para integração de modelos da MBSE com sistemas de informação que apoiam processos do ciclo de vida de satélites.

Como trabalho futuro tem-se a perspectiva de desenvolvimento de um *framework* para padronizar e direcionar a integração automatizada de modelos de engenharia de sistemas com modelos de engenharia de software.

```

/*Classe que lê o projeto do Rhapsody e armazena os elementos de interesse do SI */
class ReadRhapsody {

    private IRPPProject prj //projeto do Rhapsody

    /*Listas para armazenar os elementos de interesse do SI */
    List satelliteEquipments = []
    List subsystems = []
    List satelliteModels = []

    ReadRhapsody(){
        /*Pega o projeto ativo do Rhapsody.
        * Precisa usar o java 32-bit para rodar, porque a biblioteca rhapsody.dll é 32-bits*/
        prj = RhapsodyAppServer.activeRhapsodyApplication.activeProject()
        initialize()
    }

    /* Lê os modelos ativos no Rhapsody e popula as listas com os elementos de interesse do SI */
    private void initialize(){
        //Pega todos os elementos do projeto ativo
        List allElements = prj.nestedElementsRecursive.toList()

        allElements.each { element ->
            //armazena todos os esteriótipos do elemento
            List stereotypes = element.stereotypes.toList()*.name
            /*verifica se o elemento está classificado com esteriótipo existente no mapeamento
            acordado entre o engenheiro de software e o engenheiro de sistemas (Tabela 6.12 da tese)
            e o inclui na respectiva lista
            */
            if ("is_subsystem" in stereotypes)
                subsystems << element
            else if ("is_satellite_equipment" in stereotypes)
                satelliteEquipments << element
            else if ("is_satellite_model")
                satelliteModels << element
        }
    }
}

```

Figura 5 - Código para ler os modelos do Rhapsody


```

1  /* Classe que cria objetos para popular o banco de dados do SI
1  * a partir dos elementos lidos do modelo Rhapsody */
class InformationSystem {
    List<SatelliteModel>    satelliteModels    = []
    List<Subsystem>        subsystems        = []
    List<EquipmentType>    equipmentTypes    = []
    List<SatelliteEquipment> satelliteEquipments = []

1  public InformationSystem(){
    println "Reading Rhapsody Model ..."
    ReadRhapsody rhapsody = new ReadRhapsody()

    //popula subsystems
1  rhapsody.subsystems.each{subsystem ->
    //cria o satelliteModel que contém o subsystem
    SatelliteModel satelliteModel = new SatelliteModel(code:subsystem.ofClass.name)
    //verifica se o satelliteModel já está na lista de satelliteModels, se não adiciona
    if (! satelliteModels.find{ SatelliteModel model-> model.code == subsystem.ofClass.name })
        satelliteModels << satelliteModel
    //cria o subsystem e o adiciona na lista de subsystems
    subsystems << new Subsystem(code: subsystem.name, satelliteModel: satelliteModel)
1  }

    //popula satelliteEquipment
1  rhapsody.satelliteEquipments.each { satelliteEquipment ->
    //cria o equipmentType do satelliteEquipment
    EquipmentType equipmentType = new EquipmentType(code: satelliteEquipment.otherClass.name)
    //verifica se o equipmentType já está na lista equipmentTypes, se não, adiciona
    if (! equipmentTypes.find{ EquipmentType it-> it.code == satelliteEquipment.otherClass.name})
        equipmentTypes << equipmentType
    //obtem o subsystem do satelliteEquipment
    Subsystem subsystem = subsystems.find{ Subsystem it-> it.code == satelliteEquipment.ofClass.name }
    //cria o satelliteEquipment e o adiciona na lista de satelliteEquipments
    satelliteEquipments << new SatelliteEquipment(specification: satelliteEquipment.name,
                                                    type:equipmentType , subsystem: subsystem)
1  }
1  }

```

Figura 6 - Código para popular o sistema de informação

AGRADECIMENTOS

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

REFERÊNCIAS BIBLIOGRÁFICAS

APACHE. **Groovy Language Documentation**. 2023. Acessado em 21 de abril de 2023.

Disponível em: <<http://www.groovy-lang.org/single-page-documentation.html>>.

FRIEDENTHAL, S.; MOORE, A.; STEINER, R. **OMG Systems Modeling Language (OMG SysML™) Tutorial**. [S.l.]: INCOSE, 2009.

FRIEDENTHAL, S.; MOORE, A.; STEINER, R. **A Practical Guide to SysML: The Systems Modeling Language**. 3th. ed. [S.l.]: Morgan Kaufmann, 2014.

IBM. **Rational Rhapsody API Reference Manual**. [S.l.]: IBM Corporation, 2009.

IBM. **Essentials of IBM Rational Rhapsody for Systems Engineers v.7.6.1 - Student Manual vol 1**. [S.l.]: IBM Corporation, 2012.

IBM. **IBM Engineering Systems Design Rhapsody**. 2023. Acessado em 21 de abril de 2023.

Disponível em: << <https://www.ibm.com/products/systems-design-rhapsody>>>.

INCOSE. **IncoSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities**. 4th. ed. [S.l.]: Wiley, 2015.

INPE. **Amazon Mission**. 2022. Accessed July 13, 2022. Disponível em:

<<http://www.inpe.br/amazonia1/en/>>.

INPE/LIT. **Minuta da reunião realizada em 21 de setembro de 2017 no LIT/INPE, LIT25-LIT05-MR-001**. 2017. Disponível no centro de documentação do LIT.

OMG. **Unified Modeling Language - OMG Document Number formal/2017-12-05**. [S.l.]: OMG, 2017.

SILVA, A. C. d. P.; LOUREIRO, G. Iniciativas de adesão à mbse na área espacial. In:

RODRIGUES, A. C.; BARBOSA, A. L.; TEIXEIRA, A. F.; BATISTA, C. L. G.; OLIVEIRA, C. M. d.;

CUELLAR, D. A. (Ed.). **Anais... do 11º Workshop em Engenharia e Tecnologia Espaciais**. São

José dos Campos: Instituto Nacional de Pesquisas Espaciais (INPE), 2020. ISSN 2177-3114.

A.3 Simultaneous and collaborative development of satellites and Information Systems to support their lifecycle processes

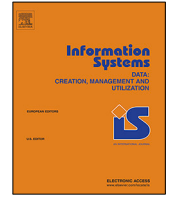
Artigo publicado no Journal Information Systems (ISSN:0306-4379) em janeiro de 2023.

- Classificação Qualis: A2 na área de Engenharias III na data da submissão (10/8/2022);
- JCR: Fator de Impacto 3.180; JCI: 0.93.



Contents lists available at ScienceDirect

Information Systems

journal homepage: www.elsevier.com/locate/is

Simultaneous and collaborative development of satellites and Information Systems to support their lifecycle processes

Ana Claudia de Paula Silva^{*}, Geilson Loureiro

Brazilian Institute for Space Research (INPE), Av. dos Astronautas, 1758, São José dos Campos, 12227-010, Brazil

ARTICLE INFO

Article history:

Received 10 August 2022
 Received in revised form 5 September 2022
 Accepted 6 September 2022
 Available online 15 September 2022
 Recommended by D. Shasha

Keywords:

Information Systems
 Space Systems Engineering
 Satellite development
 MBSE

ABSTRACT

Space Systems Engineering is the domain in which satellites are developed. A lot of information gathered by the systems engineer to develop a space product is also necessary and gathered by the software engineer in the Software Engineering domain to develop the Information Systems that will support the lifecycle processes of that product. Motivated by this reality, this paper aims to demonstrate that the work of the software engineer can be integrated with that of the systems engineer during the Systems Engineering process. We assumed the use of Model-Based Systems Engineering (MBSE) and Systems Concurrent Engineering (SCE) as the approaches to engineering a satellite and looked at the activities performed by the systems engineer to identify how the software engineer's work can be inserted into the Systems Engineering process. We use the AMAZONIA 1 satellite to exemplify and focus on the process of assembly, integration and testing of this satellite. We concluded that simultaneous and collaborative work between the Systems Engineering and Software Engineering domains is feasible and can bring quality and productivity gains for both sides.

© 2022 Elsevier Ltd. All rights reserved.

1. Introduction

This paper aims to demonstrate that the work of the software engineer, which aims to develop Information Systems to support satellite lifecycle processes, can be integrated with that of the systems engineer, which aims to develop the satellites.

Adopting Information Systems to automate the business processes of organizations can improve their efficiency. As defined by [1], a business process is a set of activities or related tasks performed to deliver an expected result, as a product or a service. In the various stages that a satellite passes through from conception to disposal, a set of activities is carried out to deliver an expected result, that is, there is a business process that can benefit from an Information System.

Software Engineering is the domain in which Information Systems are developed, and Space Systems Engineering is the domain in which satellites are developed. This research is relevant because simultaneous and collaborative work between these two domains can bring productivity and quality gains for both.

The question raised in this paper is that looking at the engineering process, it can be observed that a lot of information gathered by the systems engineer to develop a space product is also necessary and gathered by the software engineer in the

Software Engineering domain, to develop the Information Systems that will support the lifecycle processes of this product. This suggests the possibility of simultaneous and collaborative work.

In this work, we study the Space Systems Engineering process to identify how the software engineer's work can be inserted into that process.

The Systems Engineering domain has several work approaches. We chose to work with the Systems Concurrent Engineering (SCE) approach, which appears more favorable to the desired integration between the two domains.

We also consider the use of Model Based Systems Engineering (MBSE) for engineering the satellite, as this is a trend for the development of space systems [2].

The work is exemplified by the engineering process of the AMAZONIA 1 satellite, focusing on Assembly, Integration and Testing (AIT) activities, as examples of lifecycle processes.

AIT activities are a logical and interrelated sequence of events whose primary objective is to achieve a high degree of confidence that the satellite conforms to its specified performance parameters [3]. AIT includes assembling and integrating the various subsystems and performing functional and environmental tests on the satellite as a system [3].

In the AIT process, a set of activities is carried out so that an assembled, integrated and tested satellite is delivered. Although the AIT of satellites has a logical and almost standardized sequence, each organization executes it with its particularities, according to its culture and strategies, which justifies the development of a customized Information System to support this process.

^{*} Corresponding author.

E-mail address: ana.claudia@inpe.br (A.C.d.P. Silva).

Table 1
Number of papers found in Scopus and Web of Science databases on April 13, 2022.

		“Business Process Automation” “Process Automation” “BPA”	“Information Systems” “Information System” “Software tool” “Software tools” (excluding Geographic Information Systems)
“satellite” “satellites” “spacecraft” “space product” “space products” “space system” “space systems”	“Development” “lifecycle” “life cycle” “life-cycle”	0	23 (8 in the last 10 years)
	“AIT” “V&V” “IV&V” “AI&T” “AIV” “assembly” “verification” “validation” “integration”	0	10 (4 in the last 10 years)

This paper is organized as follows: Section 2 presents a literature review, Section 3 introduces the MBSE and SCE approaches, Section 4 presents the results of the work done, and Section 5 discusses the results and presents the conclusion.

2. Literature review

We searched the Web of Science and Scopus databases for works related to the topic Information Systems in the context of space product development and, more specifically, in the context of the satellites Assembly, Integration and Testing (AIT) process.

We used the keywords “Information System(s)” and “Software Tool(s)” (as Information Systems are also often referred to as software tools, although a software tool is not its only component and the term also encompasses several other types of systems). We also use the keywords “Business Process Automation” and “Process Automation”, as process automation implies the use of software tools.

These words were crossed with the terms “satellite development”, “satellite lifecycle” (and its spelling variations) and “satellite AIT”. As AIT activities are also often referred to separately and/or with other acronyms, we alternated the terms “AIT” and “AI&T” (Assembly, Integration and Testing) with the terms “V&V” (Verification and Validation), “IV&V” (Integration, Verification and Validation) and “AIV” (Assembly, Integration and Validation). The term “satellite” has also been interchanged with the more generic variations: “spacecraft”, “space product(s)” and “space system(s)”.

Results containing the term “Geographic Information System” and plural were excluded, as they do not fit the subject of interest of this paper.

The search yielded few results, as can be seen in Table 1. Of the papers returned, disregarding the repetitions, only 11 were published in the last 10 years and, except for one of the authors themselves [4], they correspond to studies that cite the search keywords but do not focus on the Information System, or the system has a purpose different from the topic of interest of this research.

In [4], we demonstrate the relationship of some Systems Engineering models with Software Engineering models, already starting our research on this topic. No other papers related to the topic of interest of this research were found in the research databases used.

We conclude that although Information Systems are widely adopted by several organizations to automate their processes and bring improvements, no studies are aimed at the development of these Information Systems by taking advantage of the modeling effort already undertaken for space product development. This fact justifies this research.

3. Introduction to the MBSE and SCE approaches

3.1. Model-Based Systems Engineering (MBSE)

In the MBSE approach, domain models are the main source of information and the main means of exchanging information among those involved in the engineering process, rather than exchanging information based on documents, which is characteristic of traditional Systems Engineering. Models are built to represent the requirements, structure and behavior of systems [5].

MBSE brings benefits such as enhanced communications, reduced development risk, improved quality, increased productivity and enhanced knowledge transfer [5].

Models can be created in several languages per the choice of the team. One of the most widely used languages nowadays is SysML (Systems Modeling Language) [5].

SysML is a general-purpose graphical modeling language that can support many different MBSE methods [5]. It is an extension of UML that supports the specification, analysis, design, verification and validation of systems [6].

3.2. Systems Concurrent Engineering (SCE)

SCE is an approach to engineering complex systems that brings the lifecycle processes requirements forward to the early stages of product development, targeting not only the product development but also simultaneously the development of the organizations that perform the lifecycle processes of this product [7].

This broader proposal justifies the choice of this approach, as the artifacts generated during the SCE process make it possible to think simultaneously and in advance about the Information Systems that will support each process of the engineering product lifecycle.

3.2.1. SCE overview

SCE integrates both Systems Engineering and Concurrent Engineering for complex product development [8]. It was proposed in 1999 by [8] and has evolved over the years according to lessons learned from its application to various complex systems [7]. The SCE approach, in its most current version, implements the framework shown in Fig. 1, which is called The Total View Framework.

In Fig. 1, the analysis dimension includes the sub-processes of the Systems Engineering process that are applied to the elements of each layer of the product decomposition structure, represented in the structure dimension. The integration dimension contains the elements that must be integrated into the system solution.

The integration dimension uses Concurrent Engineering [7]. Concurrent Engineering is a collaborative approach that brings lifecycle processes requirements forward to the early stages of product development, aiming to simultaneously develop the product and organizations within the scope of the development

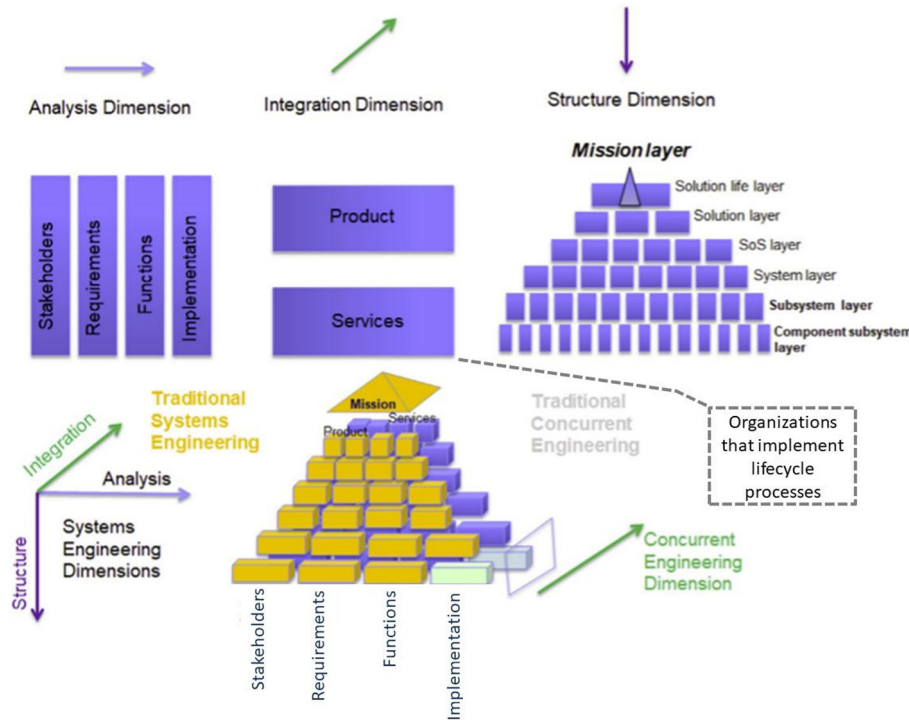


Fig. 1. SCE – The total view framework. Source: Adapted from [7].

effort. However, Concurrent Engineering is traditionally applied only at very low levels of abstraction for the design of parts and their respective lifecycle processes. SCE expands Concurrent Engineering to all layers of the structure dimension [7].

The Total View Framework assumes, therefore, that Systems Engineering processes can be applied not only to product development but also to the development of organizations (services in the figure) that perform the lifecycle processes of this product [7].

3.2.2. Activities proposed by the SCE

Based on a mission statement, the SCE approach starts by identifying needs and measures of effectiveness and from that a mission solution. In the next step, the lifecycle processes of the system of interest to be developed are identified. Each process can be further decomposed into scenarios [7].

For each process or scenario, systems requirements are derived, for both the product and the organization that implements the lifecycle process, through Stakeholder Analysis, Requirements Analysis, Functional Analysis and Implementation Analysis [7].

From the system's functional and non-functional requirements, the functional and implementation architecture are derived for both the product and the organization. For each element of the system architecture, requirements and performance measures are identified and then the detailed system design is arrived at [7].

4. Results

As proposed by Systems Concurrent Engineering (SCE), Section 4.1 presents the mission statement of the AMAZONIA 1 satellite and its lifecycle processes.

Sections 4.2 and 4.3 present SCE product and organization Stakeholder Analysis and SCE organization Functional Analysis respectively, demonstrating that these activities also act as a source of requirements for the Information Systems and that the

software engineer can work on them simultaneously with the systems engineer.

Section 4.4 demonstrates the relationship between the product implementation architecture modeled by the systems engineer and the Information System structure modeled by the software engineer. It also presents a proposal for the integration of Systems Engineering models with Software Engineering models.

Section 4.5 overviews the relationship of SCE models with the development of Information Systems.

To make some points clearer, we will use a simple Information System for order processing as a running example that will be referenced throughout this section.

4.1. AMAZONIA 1 mission and lifecycle processes

AMAZONIA 1 is the first Earth observation satellite entirely designed, assembled, integrated, tested and operated by Brazil. The mission of AMAZONIA 1 is to generate images of the planet to provide data for environmental monitoring, mainly in the Amazon [9].

Fig. 2 shows the lifecycle processes of the satellite, highlighting the Assembly, Integration and Testing (AIT) process and also the Dynamic Testing scenario, which are used to exemplify.

The Dynamic Testing scenario is an offshoot of the Environmental Testing scenario in which the satellite is exposed to controlled conditions in the laboratory to verify and guarantee its ability to withstand the environmental conditions of all phases of its lifetime, from the time of launch until the expected completion of its in-orbit operation. In the Dynamic Testing scenario, the satellite undergoes vibration and acoustic tests [10].

The AIT activities for AMAZONIA 1 were conducted at the Integration and Testing Laboratory (LIT) of the Brazilian Institute for Space Research (INPE). LIT is therefore the organization that implements the AMAZONIA 1 satellite AIT process.

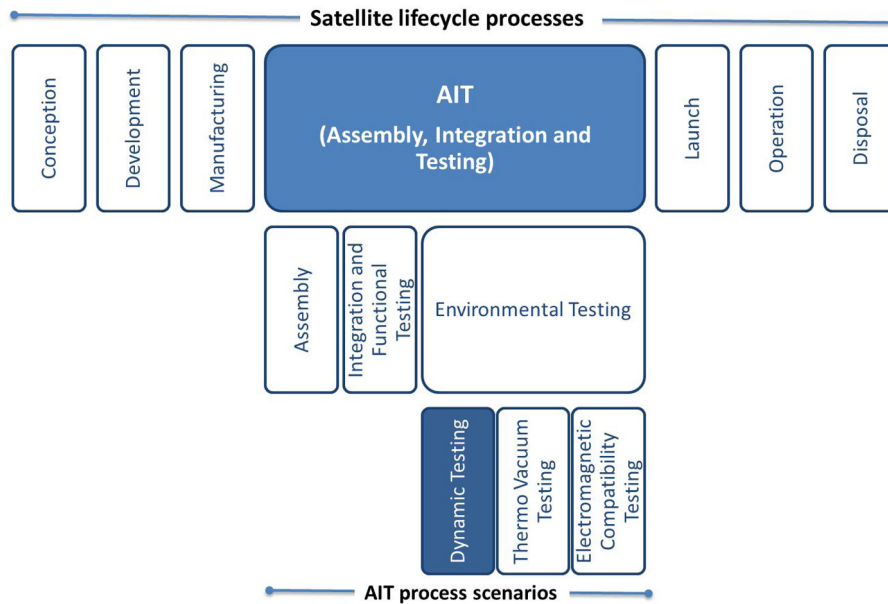


Fig. 2. AMAZONIA 1 lifecycle processes.

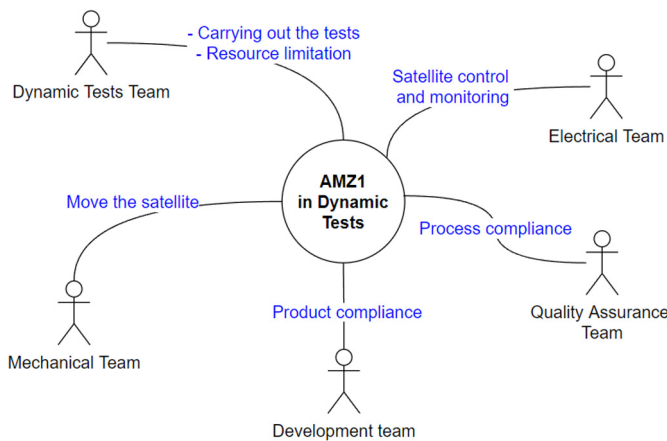


Fig. 3. Examples of the AMAZONIA 1 product stakeholders in the Dynamic Testing scenario and their concerns.

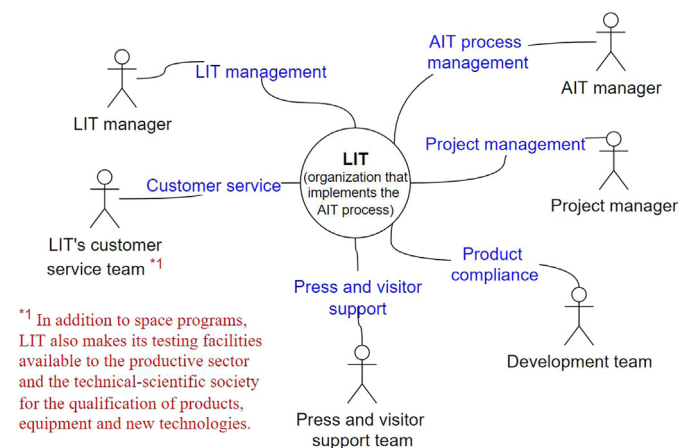


Fig. 4. Examples of stakeholders of the organization that implements the AIT process and their concerns.

4.2. SCE Stakeholder Analysis as a source of requirements for the Information Systems

For the Stakeholder Analysis, SCE proposes identifying the product and organization stakeholders and their concerns (desires, objectives, interests and worries) for each scenario of the system lifecycle process. Product stakeholders are those who affect or are affected by the product in that scenario. Organization stakeholders are those who affect or are affected by the organization implementing that scenario.

Fig. 3 shows examples of the AMAZONIA 1 product stakeholders in the Dynamic Tests scenario and their concerns. Fig. 4 shows examples of the stakeholders of LIT, the organization that implements the AIT process and their concerns.

Continuing the engineering process proposed by SCE, the concerns are translated into needs and then into stakeholder requirements. Stakeholder requirements describe what stakeholders shall be able to do. Figs. 5 and 6 illustrate this procedure.

After identifying stakeholder requirements, the next step for the systems engineer is to derive system requirements from them. System requirements describe what the system shall be

able to do to meet stakeholder requirements. Requirements analysis is carried out on stakeholder and systems requirements.

For requirement RSTK001 (see Fig. 5), for example, for the dynamic testing team to be able to fix the satellite to the shaker, the satellite must have a mechanical interface with the physical characteristics (layout and hole dimensions) of the vibrating table. This interface must also have adequate mechanical-dynamic characteristics for carrying out the planned tests. This will be written as system requirements by the systems engineer.

For requirements RSTK002 and RSTK003, installing and removing accelerometers, the satellite's points defined for measuring vibration levels must be accessible so that the dynamic testing team can install and remove accelerometers, i.e., no physical barrier, such as cabling or equipment, can block access to these points. This will also be written as system requirements by the systems engineer.

While the systems engineer works to translate the stakeholder requirements into system requirements, on the basis of the same information, the software engineer works to identify Use Cases for the Information Systems that will support the lifecycle processes.

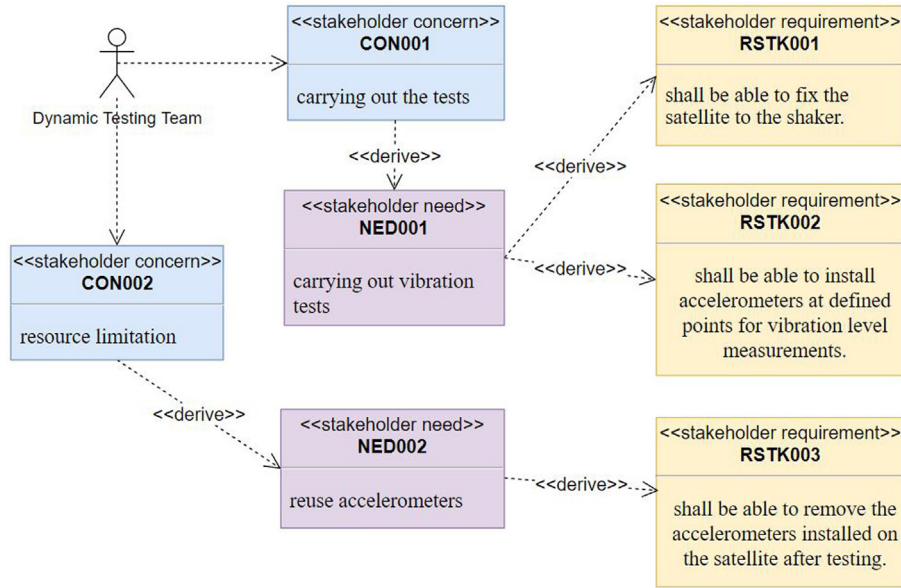


Fig. 5. Unfolding concerns into stakeholder requirements (product stakeholders).

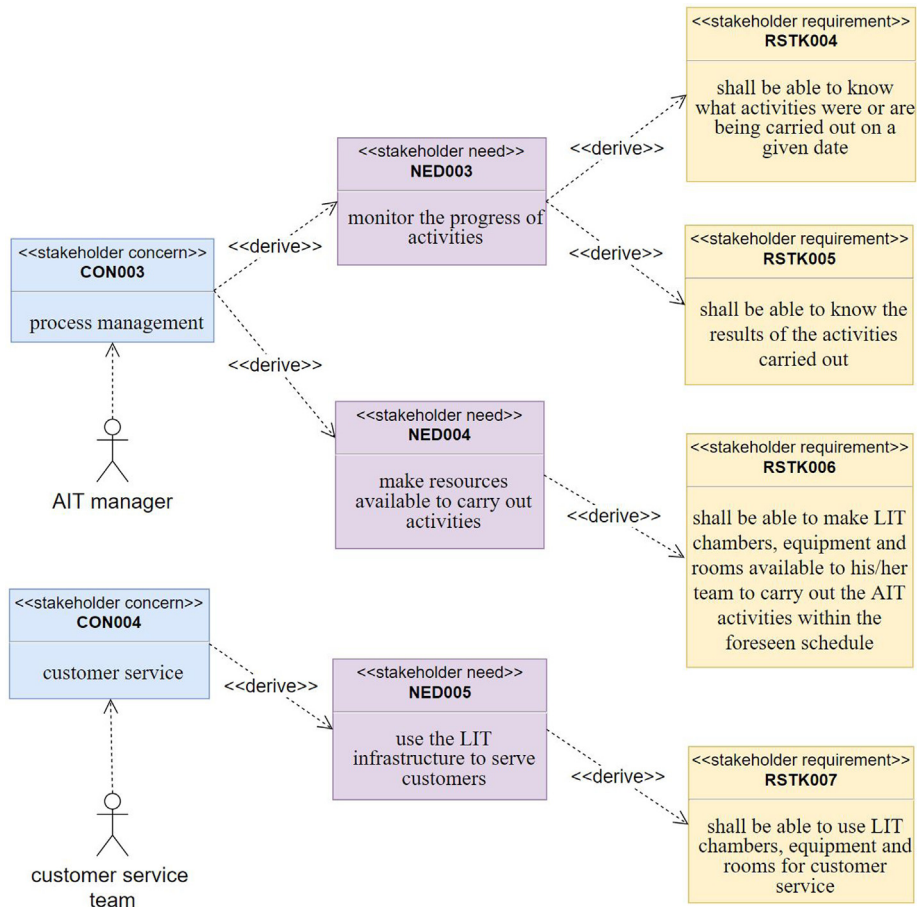


Fig. 6. Unfolding concerns into stakeholder requirements (organization stakeholders).

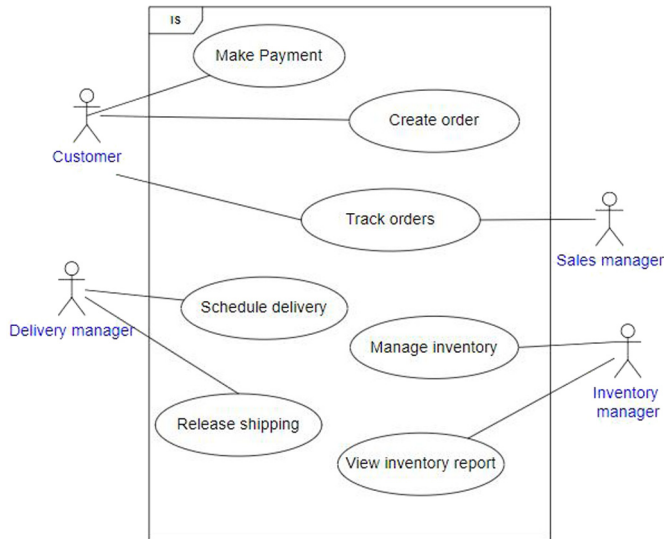


Fig. 7. Order processing Information System – Use Cases.

To make it clearer, before talking about the Information System to support satellite AIT process let us think about our running example (an order processing system). A software engineer working on it will engage enough with the stakeholders to identify their needs and requirements and satisfy their expectations. In Fig. 7 the Use Case diagram expresses in a simplified way the result of the software engineer's effort in this task.

Back to our Information System to support the satellite AIT process. Note in Figs. 3, 4, 5 and 6 that the space system stakeholders are likely actors of the Information System Use Cases and their requirements for the space system are also a source of requirements for the Information System.

For example, for requirement RSTK001, procedures for fixing the satellite on the vibrating table must be available in the Information System.

For RSTK002 and RSTK003 requirements, the features to control the installation and removal of accelerometers must be available in the Information System. A 3D view of the satellite structure highlighting the accelerometer installation points and the points with accelerometers installed can also be provided by the Information System. To provide these features, registering the accelerometers as well as the satellite structure is necessary.

Now, let us look at the organization's stakeholder requirements shown in Fig. 6. For the RSTK004 and RSTK005 requirements, the systems engineer must be concerned with what the LIT must provide to satisfy these requirements, such as delivering activity result reports after the completion of each activity. Remembering that the system solution for SCE encompasses product and organization, in this case, the system requirement needs to be satisfied by the LIT organization.

For these same requirements, the software engineer can identify that the Information System should provide functionalities to register the execution of activities and their results, as well as to allow the AIT manager to view this information. A "Track AIT Plan" Use Case can provide a graphical view of the planning and progress of activities. Returning to the diagram in Fig. 4, we can see that visualizing the results of activities is also of interest to the development team as product compliance is a concern for them.

Regarding the RSTK006 and RSTK007 stakeholder requirements, the systems engineer should be concerned with system requirements that enable the sharing of the infrastructure between AIT activities and activities for customers. For example,

LIT will not be able to schedule dynamic testing services for customers on the dates scheduled for the satellite tests.

At the same time, on the basis of the same information, the software engineer understands that the Information System must provide features for scheduling the use of the infrastructure and that this schedule must be shared between the AIT team and the teams that serve LIT clients. This requires the registration of infrastructure items.

Fig. 8 shows a diagram with the Use Cases identified by the software engineer's analysis during the Systems Engineering process. Note that the systems engineering effort could be leveraged by the software engineer.

4.3. SCE organization functional analysis as a source of requirements for the Information System

SCE proposes that a functional analysis be carried out both for the product, in each process of its lifecycle, and for the organization that implements the processes.

Functional analysis consists of a set of activities that result in a description of the required functional structure and the required functional behavior of the system.

The structure reveals a static view of the system functions, while the behavior is concerned with the temporal, logical, causal and conditional execution of functions.

4.3.1. Functional structure of the organization that implements the AIT process

Carrying out the AIT of satellites is the main function of the organization, which can be broken down into five macro functions [11]:

- Define AIT: the activities that will be executed (from the macro activities to the procedure level) and the sequence of execution are defined.
- Prepare AIT: the activities are detailed step by step, and the resource needed to execute them are identified and pointed out, such as setup, satellite equipment and GSEs (Ground Support Equipment).
- Manage AIT: the AIT manager organizes the execution of activities, prepares the schedule, releases tasks for execution and handles issues and problems that occur in the process.
- Execute AIT: the defined activities are effectively carried out, and their realization is documented.
- Control resources: the resources needed for the AIT process, such as materials, tools, staff and GSEs are managed and kept in usable condition.

Fig. 9 illustrates this functional structure; it shows the macro functions and the high-level flow of information between them. The "satellite log" flow is highlighted because we use it as an example.

The description of the functional structure is also of interest to the software engineer, as the information flow between the processes must be covered by the Information System.

The information flow is certainly a source of requirements for the Information System and a software engineer in charge of developing an Information System, for any purpose, needs to think about it.

Back to our running example (an order processing system). Fig. 10 illustrates some processes to be covered by the Information System and information flow between them. In this case, the Information System should be able to answer questions such as is there items available to fulfill such an order? What is the demand

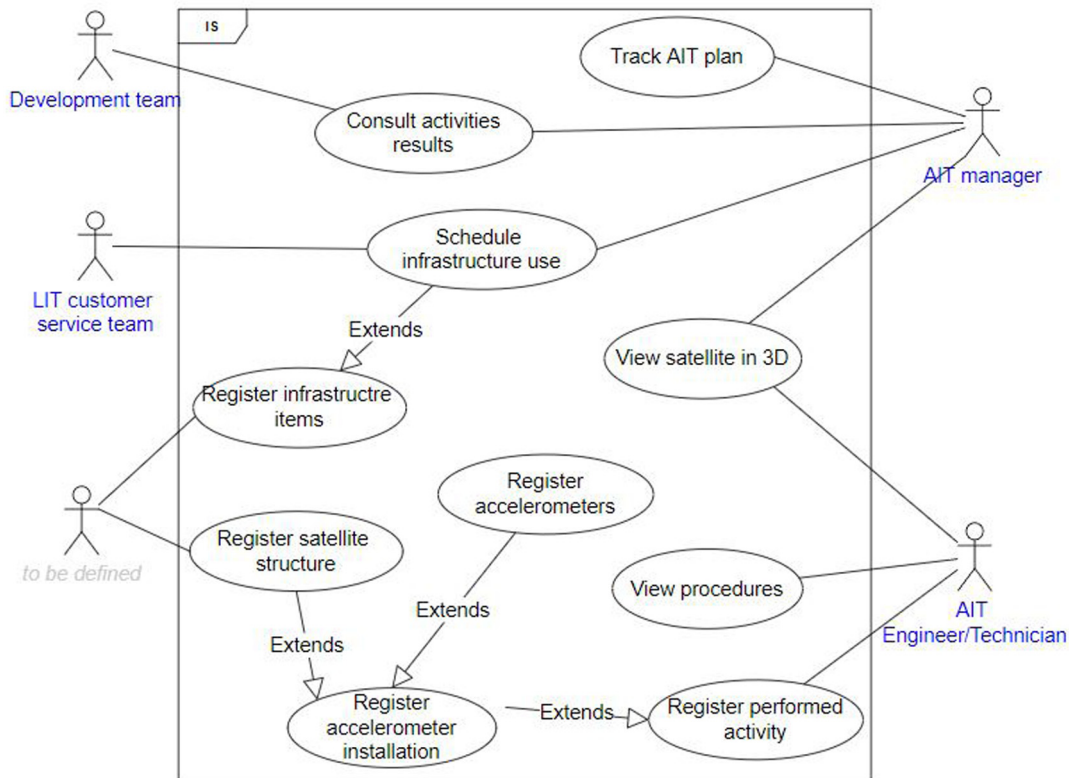


Fig. 8. Use Cases of the Information System.

for a particular item? Does inventory need to be replenished? What items need to be shipped today?

Coming back to the Information System to support the satellite AIT process, although the model shown in Fig. 9 was built to engineering the space system, it also provides the software engineer with a good foundation to work on. Again, the systems engineer's work is leveraged and not repeated by the software engineer.

Let us analyze the "satellite log" flow in the functional structure of the organization illustrated in Fig. 9.

One of the questions that the Information System must answer in the "Manage AIT" macro process is what is already assembled in a given subsystem or what was assembled on a given date. This answer is important, for example, to know which subsystem can already be tested and to analyze failure situations.

For the Information System to provide this answer, a new Use Case can be identified: Register assembly and disassembly. When some equipment is assembled or disassembled on the satellite, the operator shall record this occurrence in the Information System.

To maintain this control, each piece of equipment must be properly identified, which suggests a Use Case for the registration of equipment that arrives at the LIT for AIT.

Likewise, for every other information flow, requirements and Use Cases can be identified.

4.3.2. Functional behavior of the organization that implements the AIT process

The functional behavior of the organization is clearly a source of requirements for the Information System, as it defines the dynamics of the process that will be supported by it. Mapping functional behavior is a common activity for the Information Systems developer and is also an activity proposed by the SCE for the systems engineer.

From the organization's functional behavior, business rules emerge to be implemented in the Information System so that

it meets the normal behavior of the process flow and also the exceptions.

Fig. 11 shows a SysML Activity Diagram representing a small part of the functional behavior of the organization that implements AIT.

An example of a business rule to be implemented by the Information System that supports the AIT process:

- Equipment cannot be assembled on the satellite if it has not passed inspection.
 - The system should alert if the equipment that has not passed inspection is informed by the user when registering an assembly.

Comparing with the order processing system: an item cannot be sold if it is not in stock.

4.4. Product implementation architecture and its relationship with the Information System

Following the SCE approach, from the essential functional architecture, as well as the captured non-functional requirements, an allocable functional architecture is derived, and the system requirements are then allocated to physical elements, resulting in an implementation architecture. This is done simultaneously for the product and organization elements of the system solution. In this paper, we cover the product implementation architecture and use the SysML Block Definition Diagram (BDD) and SysML Internal Block Diagram (IBD) to represent it.

A Block is the basic architectural unit in SysML. As UML Classes, it may include both structural and behavioral features. While a Block is the definition of an element, a use or instance of that element is called a Part [6].

SysML BDD focuses on the definition and captures the system hierarchy [6]. SysML IBD focuses on usage and is usually used

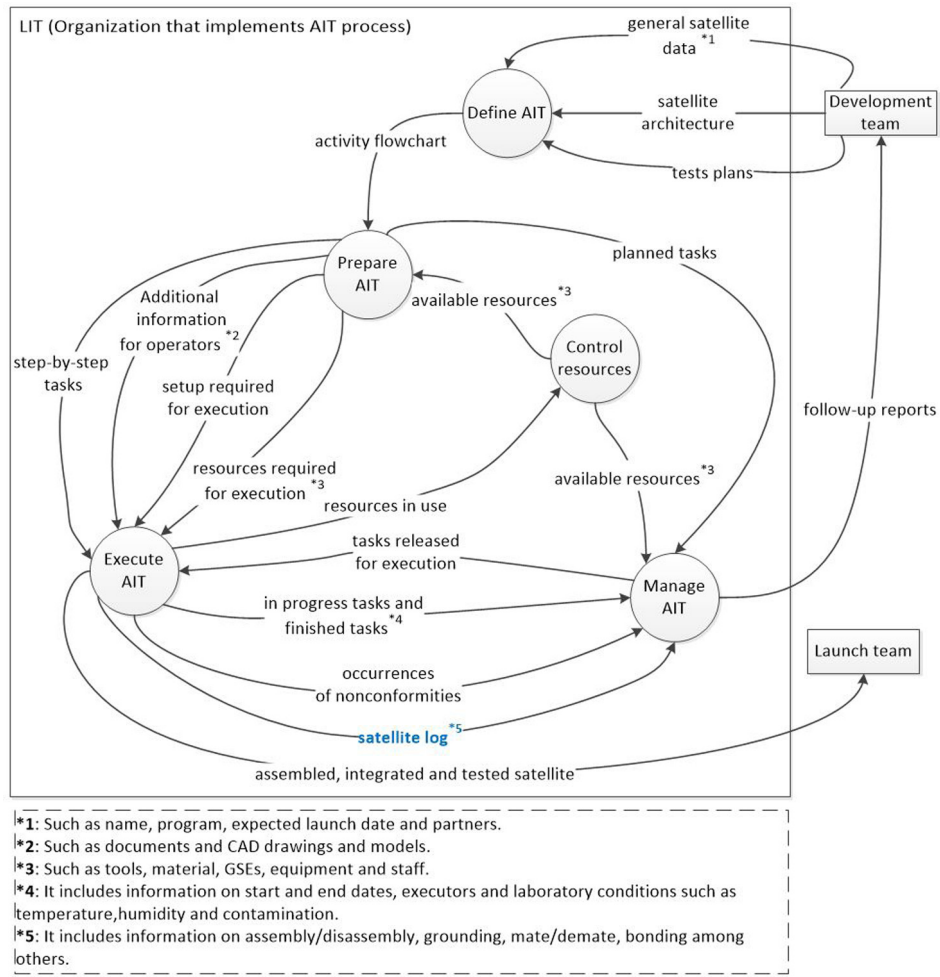


Fig. 9. Functional structure of the organization that implements AMAZONIA 1 AIT process.

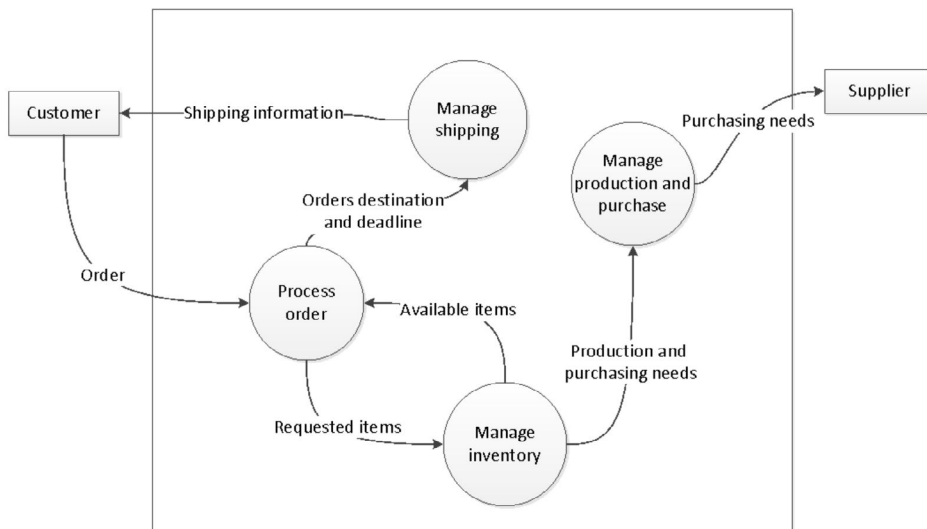


Fig. 10. Some processes of an order processing Information System.

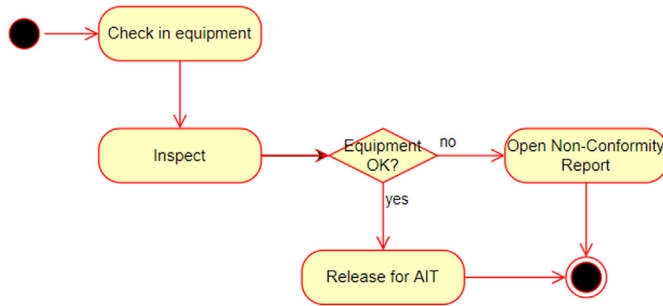


Fig. 11. Part of the functional behavior of the organization that implements the AIT process.

to represent the internal structure of the Blocks, describing how Parts and external references are related in the context of the Block [5].

4.4.1. Product implementation architecture

Fig. 12 shows a BDD that represents, in a simplified way, part of the implementation architecture of the flight model¹ (FM) of the AMAZONIA 1 satellite.

In this diagram, AMAZONIA 1 FM is presented as a composition of its subsystems, and its subsystems as a composition of its parts. Fig. 12 shows two satellite subsystems (AOCS and PSS) and some of the parts of these subsystems.

According to SysML notation, in a BDD, Blocks are represented by boxes and the connection between them is represented by the lines with diamonds at one end. These lines represent a Part Association that decomposes the Block into its Parts. The number of instances, or usage, of Blocks is represented by the number at the other end of the line. For example, the PSS subsystem is decomposed into one PCDU, one SADE, two SADA and four batteries.

Fig. 13 shows an IBD representing, in a very simplified way, part of the internal structure of the AOCS subsystem.

This diagram shows four instances of the RW Block and an instance of the ACE Block. According to the SysML notation, in an IBD, the boxes represent the Parts, and in their first compartment, the name of the Part and the Block that the Part is a type of are shown in the format: Part name: Block name.

4.4.2. An Information System model to support the LIT AIT process

The class diagram, shown in Fig. 14, represents part of an Information System model to meet the requirements of the LIT AIT process.

4.4.3. Linking product implementation architecture models and the Information System

Looking at the diagrams in Figs. 12–14, we can see that the satellite structure is modeled in the structural diagrams (IBD and BDD) in the Systems Engineering domain and also in the Class Diagram in the Software Engineering domain, but in different ways.

We can observe that the elements of the BDD and IBD do not correspond to the structural elements of the Class Diagram, but correspond to the instances of Information System Classes.

The AMZ1FM Block in the BDD corresponds to an instance of the SatelliteModel Class. The subsystems Blocks (AOCS and PSS) correspond to instances of the Subsystem Class.

¹ Developing a satellite can be very expensive and it is common for satellite development to be divided into several stages to reduce risk as it progresses. This process is based on the Philosophy of Models [12] and results in a flight model, that is, a qualified product for space.

Regarding the Blocks that appear in the subsystems Blocks decomposition in the BDD, they do not correspond to instances of the SatelliteEquipment Class, as they do not represent a specific piece of satellite equipment. Instead, these Blocks are equivalent to instances of the EquipmentType Class.

Instances of the SatelliteEquipment Class do not appear in the BDD but appear in the IBD as instances of the subsystems Blocks.

Although these Systems Engineering models and the Software Engineering Classes model have no direct relationship, we can define an integration strategy. In [4], we work with supertypes to identify the Classes of the Information System in which the instances should be allocated, but this strategy had some limitations. Using SysML stereotypes to implement this link proved more appropriate.

Stereotypes allow you to classify elements with “something in common”. With this resource, we can then classify elements of SysML models, identifying which Information System Class they are related to. In this work, we classified the elements Part with stereotypes.

Fig. 15 shows IBDs with Parts classified to exemplify. We use the stereotypes:

- <<is_satellite_equipment >>
- <<is_satellite_model >>
- <<is_subsystem >>

to identify the elements of interest to the Information System. Element properties are also pointed there because we use them in the next steps.

To work with this stereotype strategy, it is necessary to map the destination of each information in the SysML model that is of interest to the Information System, as exemplified in Table 2. In this table, in the column “Part Attribute”, the terms in parenthesis correspond to the name of the attributes in the Rhapsody tool [13], which we used to create the SysML models.

We use a Rhapsody library, JAVA API, which allows reading and writing in the models, and we created a script in the groovy language to read the model, interpret it according to the defined mapping and generate inputs for the Information System from the engineering models.

4.5. SCE models x Information System development – overview

Table 3 overviews the relationship between SCE models and Information Systems development. The analyses of the analysis dimension applied to the elements of the integration dimension result in models that relate to the development of Information Systems as explained in this table.

5. Discussion and conclusion

Information Systems are important tools for increasing the efficiency of processes in organizations. The specificity of the satellite lifecycle processes and the organizations that implement these processes makes the custom development of these systems the most suitable option.

Understanding these processes and developing a customized Information System for the organization require a significant effort from the software engineer. In this article, we have seen that much of the information resulting from the systems engineer’s effort to develop a satellite is also of interest to the software engineer who will develop the Information System, which suggests that simultaneous and collaborative work can save development time and effort.

To specify and design a satellite the systems engineer needs to explore its lifecycle processes. When working on an Information System to support these processes, the software engineer will

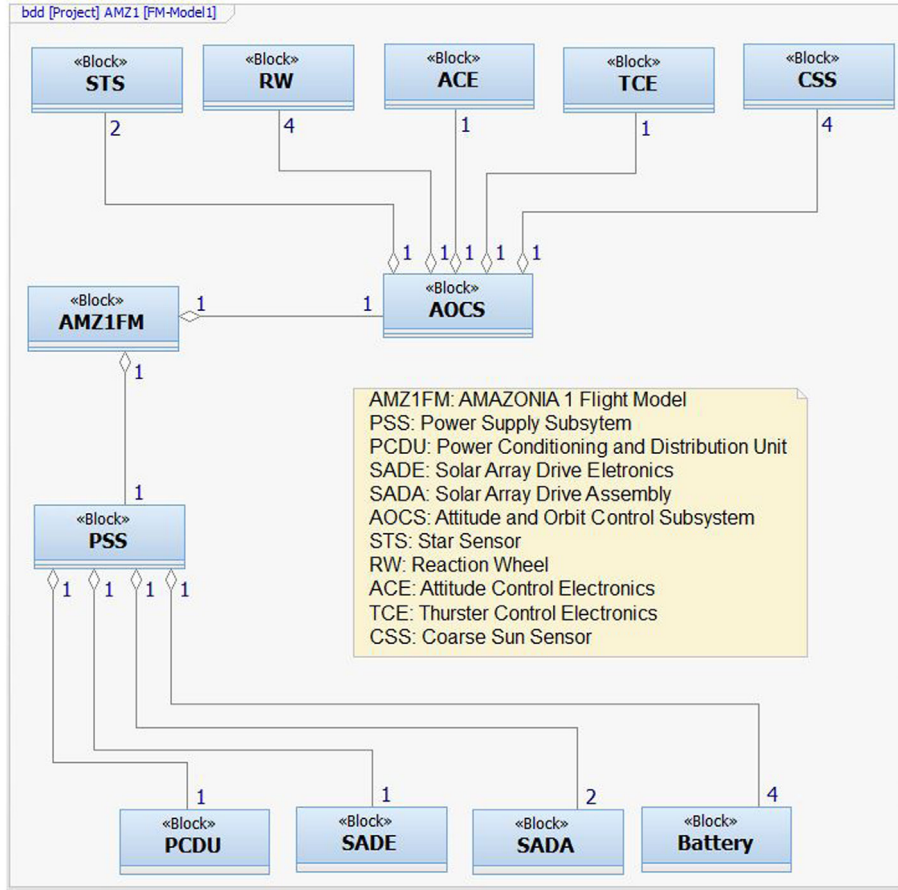


Fig. 12. Simplified SysML BDD for part of the flight model of Amazonia-1 satellite.

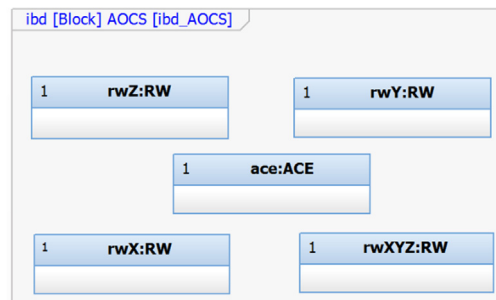


Fig. 13. Simplified SysML IBD representing the AOCs subsystem of AMAZONIA 1.

Table 2
Mapping between the two domains.

Stereotype (in SysML model)	Class (in Information System model)	Part Attribute (in SysML model)	Class Attribute (in Information System model)
is_subsystem	Subsystem	name (name) Parent block (ofClass)	code satelliteModel
is_satellite_equipment	EquipmentType SatelliteEquipment	type (otherClass) name (name) type (otherClass)	code specification equipmentType
is_satellite_model	SatelliteModel	name (name)	code

also need to explore them. The idea is to work together to develop Information Systems taking advantage of the modeling effort already made for the development of space products.

Systems Concurrent Engineering (SCE) is an approach that proves favorable to the integration of the software engineer's work with that of the systems engineer, as in SCE the product

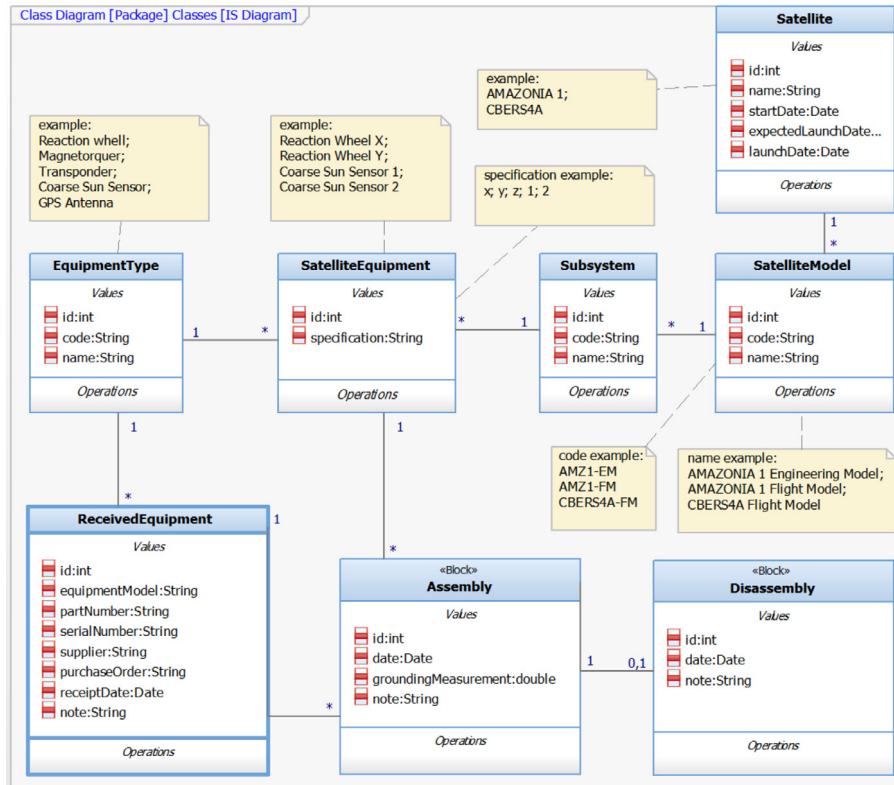


Fig. 14. Simplified UML class diagram representing part of the software structure for an Information System for the AIT process of LIT.

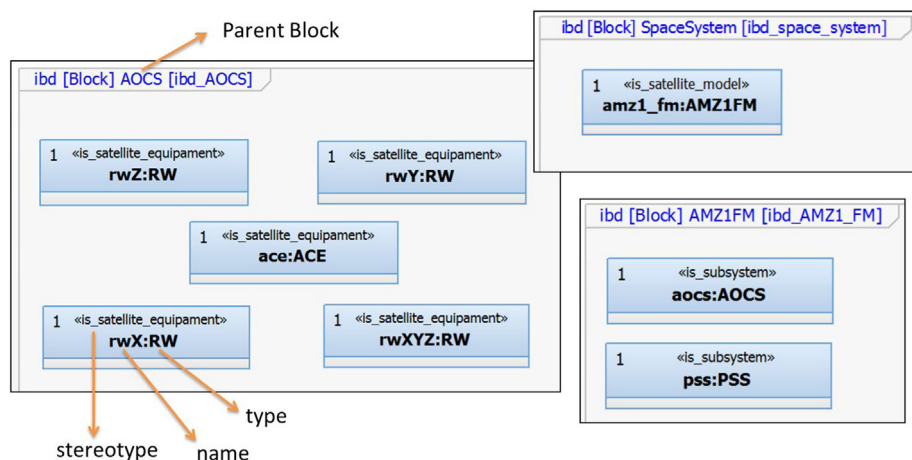


Fig. 15. SysML parts classified with stereotypes.

lifecycle processes are analyzed from the beginning of the project, providing information in a timely manner for the development of Information Systems that will support these processes.

We have seen that information emerging from the activities proposed by the SCE is also a source of requirements for Information Systems.

We also have seen that models generated for the engineering of the space system, considering a model-based approach, can be integrated into the models of Information Systems. Although the focus is different while modeling for Space Systems engineering and Information Systems engineering, these two activities are related and working together can be beneficial.

Another point to be noticed is that the systems engineer's prior knowledge of the Information Systems that will support the execution of the organization's processes can influence the

functional and implementation architecture of that organization, as the organization's functions can be performed in an automated way by the Information System, which can be part of the solution to meet certain stakeholder requirements.

This work assumed the use of Systems Concurrent Engineering (SCE) and Model Based Systems Engineering (MBSE) for the engineering of a satellite and presented examples of the application of these approaches on the case of the AMAZONIA 1 satellite. These examples indicate as follows:

- the work of the software engineer, who aims to develop the Information Systems that will support the satellite's lifecycle processes, can benefit from integration with the work of the systems engineer, who aims to develop the satellite, as the

Table 3
SCE models × Information System development – Overview.

		INTEGRATION DIMENSION	
		PRODUCT	ORGANIZATION
ANALYSIS DIMENSION	STAKEHOLDERS	How can the Information System act to contribute to meeting the requirements of the stakeholders of the space system? This is the main issue for the software engineer during Stakeholder Analysis and Requirements Analysis.	
	REQUIREMENTS	The concerns and requirements related to the space system expressed in these models form a basis for the software engineer to identify requirements for the Information System, as the Information System will be a support tool for the needs of stakeholders to be met.	
	FUNCTIONS	<p>The functional architecture of the product does not influence the Information System, and the Information System does not interfere with it.</p> <p>Product functional models have little or no influence on Information System development.</p>	<p>Functional structure (static view): These models contain the information that flows between the organization's processes. The Information System must be shaped so that this information is available in it. Use Cases are identified based on the information needs expressed in these models.</p> <p>Functional behavior (dynamic view): These models express the way processes are executed in the organization with regard to their temporal, logical, causal and conditional execution. Therefore, they point out business rules for the Information System.</p>
	IMPLEMENTATION	<p>These models express the implementation of the system. That is, in the case of satellites, the physical items (subsystems and their components) designed to perform the functions defined in the functional models of the product.</p> <p>This physical structure of the product is of interest to the Information System, as it will be referenced in it to record various actions, such as what arrived, what was assembled, what was connected, etc.</p> <p>A relationship between these Systems Engineering models and the Information System data model is possible through a mapping (product implementation model x Information System data model) and a code to read the implementation models of the product, interpret them according to the mapping and feed the Information System database.</p>	<p>These models express the implementation of the organization implementing the product lifecycle process. That is, items needed to perform the functions defined in the functional models of the organization, such as buildings, rooms, chambers, laboratories and staff.</p> <p>The organization's physical structure is of interest to the Information System in the sense that, in the case of satellite AIT, for example, the physical spaces and test chambers are items that will have their use managed via the Information System. These models are then a source of data for the Information System and can also be mapped to feed the Information System database.</p> <p>As for the organizational structure, the teams designated to perform the functions point out Use Case actors and relevant information to define access control to the Information System.</p>

information emerging from the Systems Engineering process is also of interest to Information Systems;

- the work of the software engineer can also benefit the work of the systems engineer, as the Information System can become an element of the space system engineering solution.

This concurrent and collaborative work revealed the potential to bring productivity and quality enhancement to both domains.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgments

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

References

[1] M.v. Rosing, A.-W. Scheer, H.v. Scheel, *The complete business process handbook, volume 1 - body of knowledge from process modeling to BPM, 1, 2015.*
 [2] A.C.d.P. Silva, G. Loureiro, *Iniciativas de adesão à MBSE na área espacial*, in: A.C. Rodrigues, A.L. Barbosa, A.F. Teixeira, C.L.G. Batista, C.M.d. Oliveira, D.A. Cuellar (Eds.), *Anais... Do 11º Workshop Em Engenharia E Tecnologia*

- Espaciais, 11. (WETE), Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos, 2020, URL <http://urlib.net/rep/8JMKD3MGPDW34R/433LTT2>.
- [3] A.C. Silva, G. Loureiro, Integrated development of space systems - design for AIT - design for assembly, Integration and Testing of satellites - D4AIT, in: 2011 IEEE International Conference on Industrial Engineering and Engineering Management, 2011, pp. 590–594, <http://dx.doi.org/10.1109/IEEM.2011.6117985>.
- [4] A.C.d.P. Silva, G. Loureiro, The relationship between the model based systems engineering models and information systems to support space products lifecycle processes, in: International Astronautical Congress, Bremen, Germany, 2018.
- [5] S. Friedenthal, A. Moore, R. Steiner, A Practical Guide To SysML: The Systems Modeling Language, third ed, Elsevier, 2015, <http://dx.doi.org/10.1016/c2013-0-14457-1>.
- [6] IBM, Essentials of IBM rational rhapsody for systems engineers v.7.6.1 - student manual vol 1, 2012.
- [7] G. Loureiro, W. Panades, A. Silva, Lessons learned in 20 years of application of Systems Concurrent Engineering to space products, Acta Astronaut. 151 (2018) 44–52, <http://dx.doi.org/10.1016/j.actaastro.2018.05.042>, URL <https://www.sciencedirect.com/science/article/pii/S0094576518304272>.
- [8] G. Loureiro, A Systems Engineering and Concurrent Engineering Framework for the Integrated Development of Complex Products (Ph.D. thesis), Loughborough University, Loughborough . UK., 1999.
- [9] INPE, Amazon mission, 2022, Accessed July 13, 2022. URL <http://www.inpe.br/amazonia1/en/>.
- [10] INPE/LIT, Ensaios dinâmicos, 2022, Accessed July 13, 2022. URL <http://www.lit.inpe.br/pt-br/vibracao>.
- [11] INPE/LIT, Minutes of the meeting held on September 21, 2017 at LIT/INPE, LIT25-LIT05-MR-001, 2017, Available from the LIT documentation center.
- [12] E.C.F.S. STANDARDIZATON, ECSS-E-HB-10-02A: Space engineering - Verification guidelines, Noordwijk, 2010.
- [13] I. Corporation, IBM engineering systems design rhapsody, 2022, Accessed July 13, 2022. URL <https://www.ibm.com/products/systems-design-rhapsody>.

A.4 Análise de *stakeholders* do sistema espacial como fonte de requisitos para os Sistemas de Informação que darão suporte aos processos do seu ciclo de vida

Artigo apresentado no 12^o *Workshop* em Engenharia e Tecnologia Espaciais (WETE) em novembro de 2021.



Análise de stakeholders do sistema espacial como fonte de requisitos para os Sistemas de Informação que darão suporte aos processos do seu ciclo de vida

Ana Claudia de Paula Silva¹, Dr. Geilson Loureiro²

¹ Instituto Nacional de Pesquisas Espaciais, São José dos Campos, SP, Brasil
Aluna de Doutorado do curso de Engenharia e Gerenciamento de Sistemas Espaciais - CSE.

²Instituto Nacional de Pesquisas Espaciais, São José dos Campos, SP, Brasil
Orientador de Doutorado.

ana.claudia@inpe.br

Resumo. *Muitas informações levantadas pelo engenheiro de sistemas, que trabalha para construir o sistema espacial, também são de interesse do engenheiro de software que trabalha para construir os Sistemas de Informação que darão suporte aos processos do ciclo de vida desse sistema. Embora essas atividades costumem ser realizadas independentemente, o potencial de que o esforço realizado no domínio da Engenharia de Sistemas seja também aproveitado no domínio da Engenharia de Software motivou esta pesquisa. Este artigo avaliou a Análise de Stakeholders proposta pela Engenharia Simultânea de Sistemas (SCE) como fonte de requisitos para o desenvolvimento de Sistemas de Informação que darão suporte aos processos do ciclo de vida do sistema espacial. A Análise de Stakeholders foi exemplificada com o caso do satélite AMAZONIA 1 no cenário de ensaios dinâmicos e o resultado dessa análise foi explorado como fonte de requisitos para o desenvolvimento de um Sistema de Informação para apoiar o processo de Montagem, Integração e Testes (AIT) desse satélite. Pode-se observar que o processo de Análise de Stakeholders fornece saídas que também interessam ao desenvolvedor do Sistema de Informação e que as saídas do processo de desenvolvimento do Sistema de Informação podem influenciar positivamente na solução de engenharia do sistema espacial. O resultado apresentado sinaliza que o desenvolvimento simultâneo e colaborativo do sistema espacial e de Sistemas de Informação para apoiar os processos do seu ciclo de vida é viável e vantajoso.*

Palavras-chave: Sistema de Informação, Engenharia de Sistemas Espaciais, Desenvolvimento de Satélites

1. Introdução

Este artigo utiliza o processo de Montagem, Integração e Testes (AIT) do satélite AMAZONIA 1 como caso de estudo para investigar a possível colaboração entre a atividade de análise de *stakeholders* realizada no processo de engenharia do sistema espacial e o desenvolvimento de Sistemas de Informação para apoiar os processos do ciclo de vida desse sistema.

Sistema de Informação (SI) é um conjunto de componentes inter-relacionados que coleta, processa, armazena e dissemina dados e informações. Sistemas de Informação baseados em com-



putador incluem hardware, software, redes, pessoas e procedimentos (STAIR; REYNOLDS, 2018).

Um SI adequado facilita o acesso às informações, elimina ou minimiza possíveis inconsistências, assim como a realização da mesma tarefa por mais de uma vez por diferentes interessados na mesma informação ao longo dos processos de negócio das organizações.

Processo de negócio é um conjunto de atividades ou tarefas relacionadas que são executadas para entregar um resultado esperado, como um produto ou um serviço (ROSING; SCHEER; SCHEEL, 2015). No processo de AIT de satélites, um conjunto de atividades é realizado para que um satélite montado, integrado e testado seja entregue. Essas atividades incluem não somente as atividades de montagem, integração e testes propriamente ditas, mas também todas as outras que possibilitam que essas atividades principais aconteçam.

Sistemas de Informação para atender às necessidades do processo de AIT de ponta a ponta não são encontrados no mercado e, ainda que fossem, um desenvolvimento customizado para as organizações que implementam esse processo seria uma melhor opção, pois um SI desenvolvido sob medida adequa-se aos processos e às características da organização, enquanto ao optar por comprar um sistema de prateleira, a organização acaba por ter que adequar seus processos, sua forma de trabalho, ao sistema. Conforme colocado por (STAIR; REYNOLDS, 2018), um sistema desenvolvido especificamente para o processo de negócio da própria organização tem maior potencial de atender suas necessidades e obter vantagem competitiva.

Muitas informações levantadas no domínio da Engenharia de Sistemas, pelo engenheiro de sistemas que tem como objetivo construir o sistema espacial, também são necessárias no domínio da Engenharia de Software para construir o SI que dará suporte aos processos do ciclo de vida desse sistema. Há potencial para que o esforço realizado no domínio da Engenharia de Sistemas seja aproveitado também no domínio da Engenharia de Software, e este potencial é a principal motivação para esta pesquisa.

Foram realizadas buscas nas bases *Web of Science* e *Scopus* com o intuito de encontrar trabalhos relacionados ao tema desta pesquisa, mas não foram encontrados, com exceção de um trabalho dos próprios autores (SILVA; LOUREIRO, 2018), já abordando esse tema. Foram encontrados alguns artigos da área espacial envolvendo Sistemas de Informação, porém esses trabalhos não têm foco nesses sistemas, ou ainda o sistema tem finalidade diferente do tema de interesse desta pesquisa, como, por exemplo, softwares de simulação (TAN; LABRADOR; TALAMPAS, 2020).

Esta artigo apresenta resultados parciais de uma pesquisa mais abrangente, em andamento, que visa o desenvolvimento simultâneo do sistema espacial e de Sistemas de Informação para apoiar os processos do seu ciclo de vida.

O objetivo deste artigo é avaliar a atividade de análise de *stakeholders* realizada no processo de engenharia do sistema espacial como fonte de requisitos para os Sistemas de Informação que darão suporte aos processos do ciclo de vida desse sistema.

2. Metodologia

Para realização deste trabalho, foi adotada a Engenharia Simultânea de Sistemas (SCE) como abordagem para a engenharia do sistema espacial. O uso da SCE foi exemplificado com o caso do satélite AMAZONIA 1. Para gerar resultados da análise de *stakeholders*, que foram base para o trabalho, foram realizadas entrevistas com dois membros da equipe de AIT do LIT que



participaram do projeto AMAZONIA 1. Um deles respondeu sobre o processo de AIT como um todo e o outro respondeu especificamente sobre as atividades de ensaios dinâmicos. A análise de *stakeholders* foi avaliada como fonte de requisitos com base na experiência dos autores com desenvolvimento de Sistemas de Informação.

3. Resultados e Discussão

3.1. Engenharia Simultânea de Sistemas (SCE)

3.1.1. Visão geral e justificativa para utilização da SCE

Para realização deste trabalho, consideramos o uso da SCE como abordagem para a engenharia do sistema espacial.

SCE é uma abordagem que integra Engenharia de Sistemas e Engenharia Simultânea para o desenvolvimento de produtos complexos (LOUREIRO, 1999). Ela foi proposta em 1999 por (LOUREIRO, 1999) e veio evoluindo ao longo dos anos de acordo com as lições aprendidas com sua aplicação em diversos sistemas complexos (LOUREIRO; PANADES; SILVA, 2018). A abordagem SCE, em sua versão mais atual, implementa o *framework* apresentado na Figura 1, que é denominado *The Total View Framework*



Figura 1. Total View Framework. [Fonte: traduzida de (LOUREIRO; PANADES; SILVA, 2018)]

Na figura 1, a dimensão de análise inclui os subprocessos do processo de Engenharia de Sistemas que são aplicados aos elementos de cada camada da estrutura de decomposição do produto, representados na dimensão de estrutura. A dimensão de integração contém os elementos que devem ser integrados na solução sistema. O *Total View Framework* assume que os processos de Engenharia de Sistemas podem ser aplicados tanto para o desenvolvimento do produto,



quanto das organizações (*serviço* na figura) que executam os processos do ciclo de vida desse produto (LOUREIRO; PANADES; SILVA, 2018). É a dimensão de integração que usa a engenharia simultânea (LOUREIRO; PANADES; SILVA, 2018).

A Engenharia Simultânea é uma abordagem colaborativa que antecipa os requisitos dos processos do ciclo de vida para os estágios iniciais do desenvolvimento do produto, visando o desenvolvimento simultâneo do produto e das organizações que estão dentro do escopo do esforço de desenvolvimento. Porém, tradicionalmente a Engenharia Simultânea é aplicada somente em níveis muito baixos de abstração para o projeto de peças e seus respectivos processos de ciclo de vida. A SCE expande a engenharia simultânea para todas as camadas da dimensão de estrutura (LOUREIRO; PANADES; SILVA, 2018).

Essa proposta de Engenharia de Sistemas mais abrangente justifica a escolha da SCE para este trabalho, já que os artefatos gerados durante o processo da SCE possibilitam pensar simultaneamente e antecipadamente também sobre os Sistemas de Informação que darão suporte a cada processo do ciclo de vida do produto.

3.1.2. Atividades propostas pela SCE

Com base em uma declaração de missão, a abordagem SCE começa identificando necessidades e medidas de efetividade e, a partir disso, uma solução de missão. Em um próximo passo, são identificados os processos do ciclo de vida do sistema de interesse a ser desenvolvido. Cada processo do ciclo de vida pode ser ainda decomposto em cenários.

Para cada processo ou cenário do processo do ciclo de vida, requisitos de sistemas são derivados através da Análise de *Stakeholders* e das análises de contexto Funcional e de Implementação, tanto para o produto como para a organização que implementa o processo do ciclo de vida.

Dos requisitos funcionais e não funcionais do sistema são derivadas a arquitetura funcional e a arquitetura de implementação, tanto do produto como da organização. Para cada elemento da arquitetura do sistema são identificados requisitos e medidas de performance e então, chega-se ao projeto detalhado do sistema.

Das atividades propostas pela SCE, este artigo explora a Análise de *Stakeholders* de produto e de organização como fonte de requisitos para os Sistemas de Informação que darão suporte aos processos do ciclo de vida do produto espacial. As demais análises propostas pela SCE não são exploradas neste artigo e são metas de trabalhos futuros.

3.2. Análise de *Stakeholders* da SCE como fonte de requisitos para o Sistema de Informação - Integração do trabalho do engenheiro de software ao processo de Engenharia de Sistemas

O sistema espacial utilizado como exemplo é o satélite AMAZONIA 1. Este é o primeiro satélite de observação da Terra inteiramente projetado, montado, integrado, testado e operado pelo Brasil. A missão do AMAZONIA 1 é gerar imagens do planeta para fornecer dados para monitoramento ambiental, principalmente na região amazônica (INPE, 2021).

A Figura 2 apresenta os processos do ciclo de vida do satélite AMAZONIA 1, destacando o processo de AIT e o cenário de Ensaios Dinâmicos, que é utilizado para exemplificar a Análise de *Stakeholders*.

O cenário de Ensaios Dinâmicos é um desdobramento do cenário de ensaios ambientais, no qual o satélite é exposto a condições controladas em laboratório com o objetivo de verificar e



garantir sua capacidade de suportar as condições ambientais de todas as fases de sua vida útil, desde o momento do lançamento até o término previsto de sua operação em órbita. No cenário de Ensaios Dinâmicos, o satélite passa por ensaios de vibração e ensaios acústicos (LIT, 2021).

As atividades de AIT para o AMAZÔNIA 1 foram realizadas no Laboratório de Integração e Testes (LIT) do Instituto Nacional de Pesquisas Espaciais (INPE).

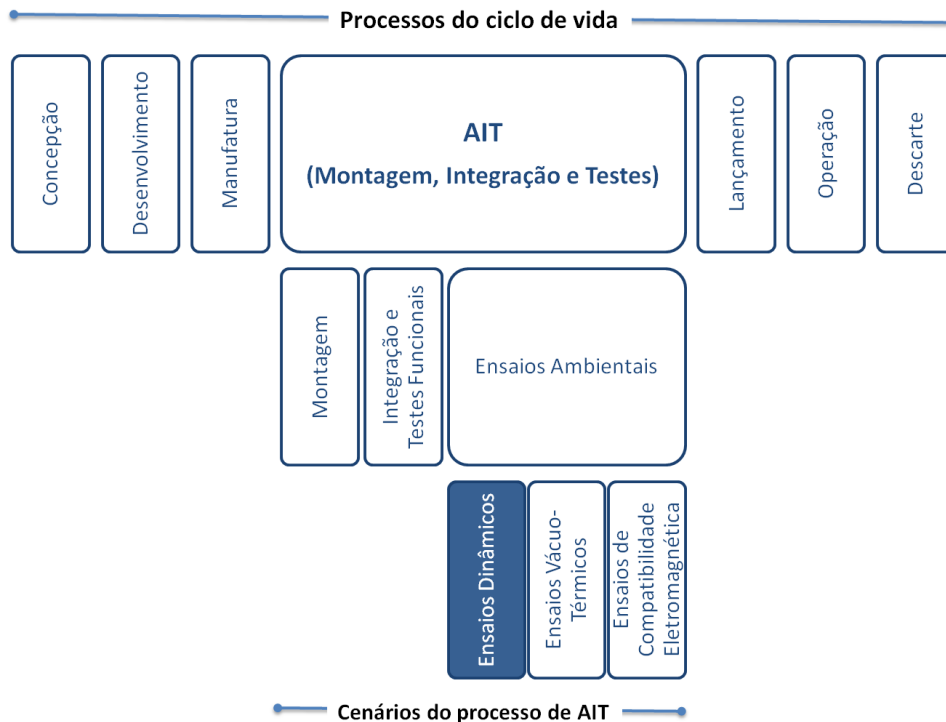


Figura 2. Processos do ciclo de vida do satélite com desdobramento em cenários do processo de AIT

Para a Análise de *Stakeholders*, a SCE propõe que para cada cenário do processo do ciclo de vida do sistema sejam identificados os *stakeholders* do produto e da organização e seus *concerns* (desejos, vontades, objetivos, interesses, preocupações). *Stakeholders* do produto são aqueles que afetam ou são afetados pelo produto naquele cenário. *Stakeholders* da organização são aqueles que afetam ou são afetados pela organização que implementa aquele cenário. A Figura 3 apresenta amostras de *stakeholders* do produto AMAZONIA 1 no cenário de Ensaios Dinâmicos e de seus *concerns*. A Figura 4 apresenta amostras de *stakeholders* do LIT, organização que implementa o processo de AIT, e de seus *concerns*.

Na continuidade do processo de engenharia proposto pela SCE, os *concerns* são transcritos em necessidades e então em requisitos de *stakeholder*. Os requisitos de *stakeholder* descrevem o que os *stakeholders* devem ser capazes de fazer. As Figuras 5 e 6 ilustram esse procedimento.

Após identificar os requisitos de *stakeholder*, o próximo passo para o engenheiro de sistemas é derivar a partir deles os requisitos de sistema. Os requisitos de sistema descrevem o que o sistema deve fazer para que os requisitos de *stakeholder* sejam satisfeitos.

Por exemplo, para o requisito RSTK001, para que a Equipe de Ensaios Dinâmicos consiga fixar o satélite no vibrador, o satélite deverá possuir uma interface mecânica que, além de ser compatível com o sistema de fixação do satélite no veículo lançador, tenha as características

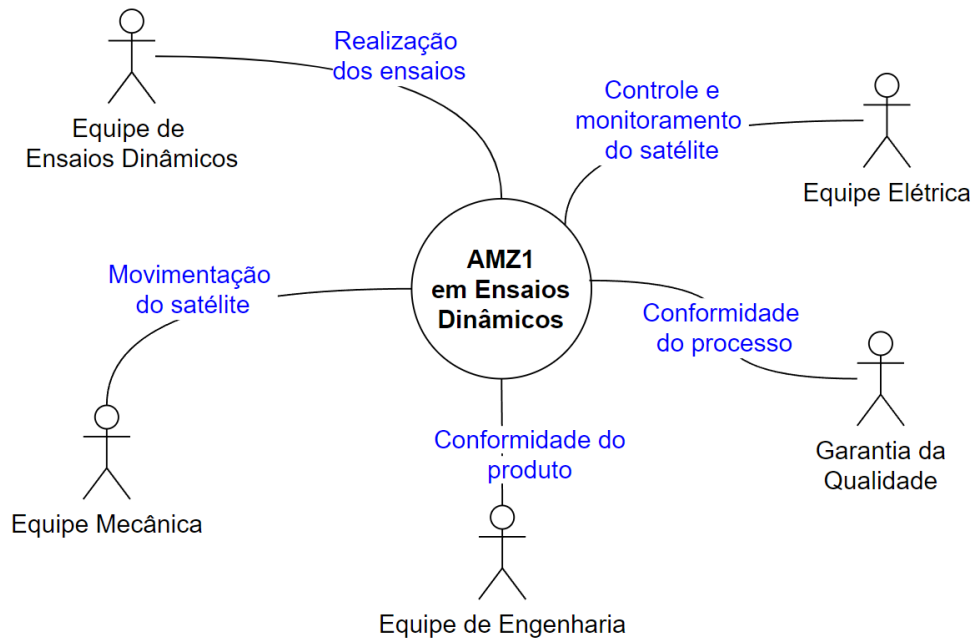


Figura 3. Stakeholders do produto AMAZONIA 1 no cenário de Ensaios Dinâmicos e seus *concerns*

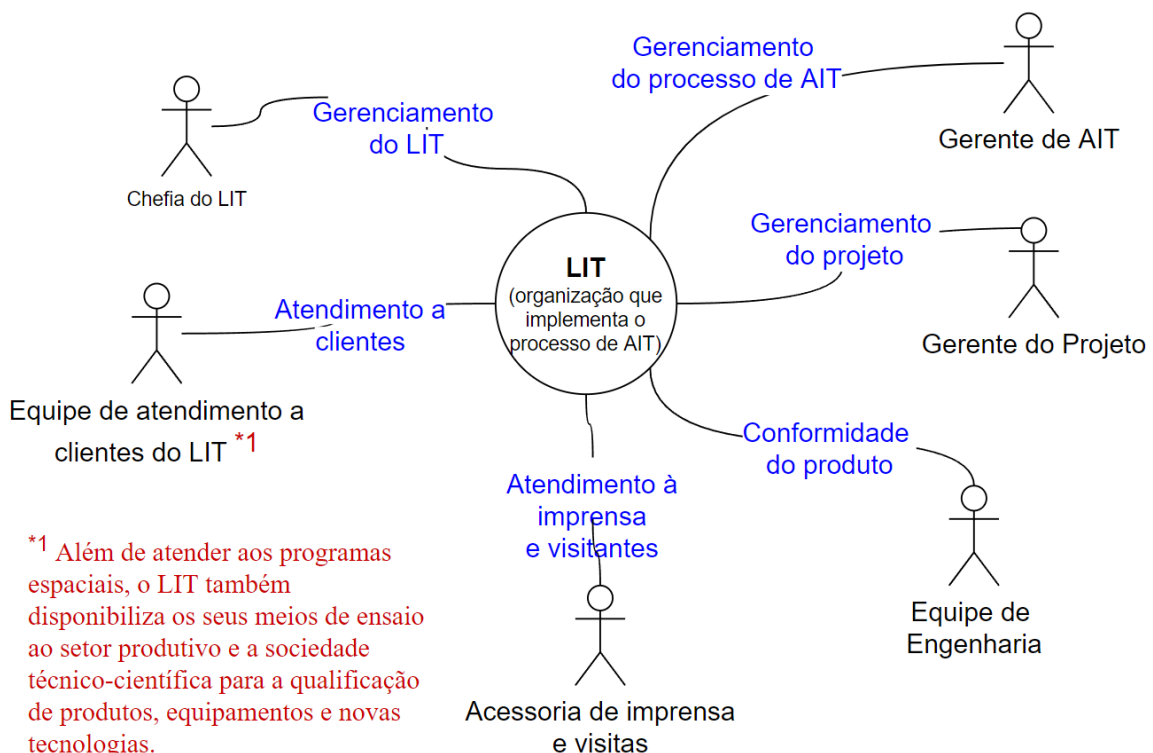


Figura 4. Stakeholders da organização que implementa o processo de AIT e seus *concerns*

físicas (layout de furação e dimensões) da mesa vibratória. Tal interface deverá também ter características mecânico-dinâmicas adequadas para a realização dos ensaios previstos.

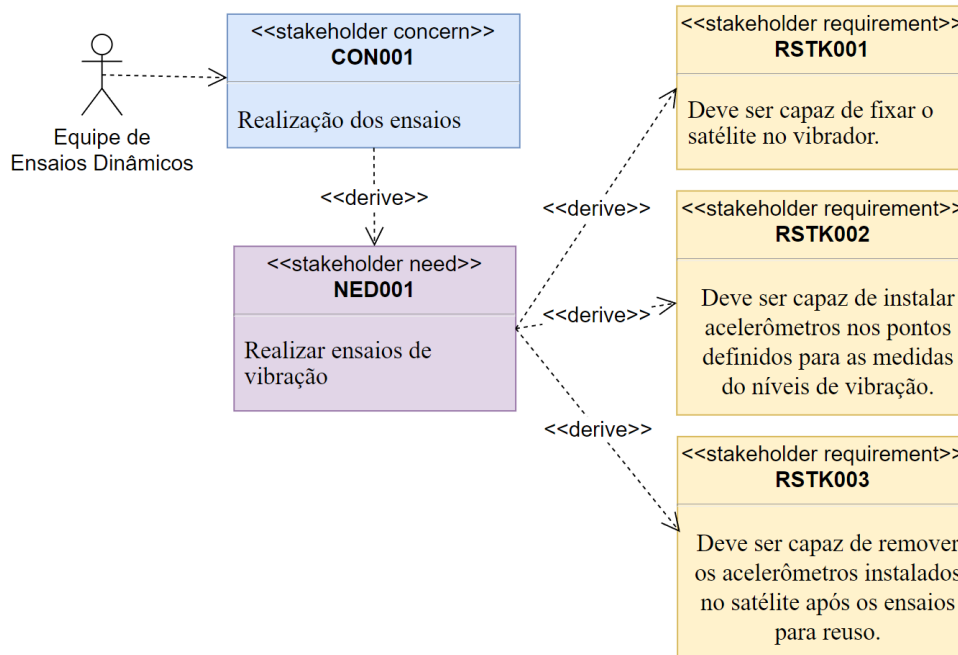


Figura 5. Desdobramento de *concerns* em requisitos de *stakeholder* (*stakeholders* do produto)

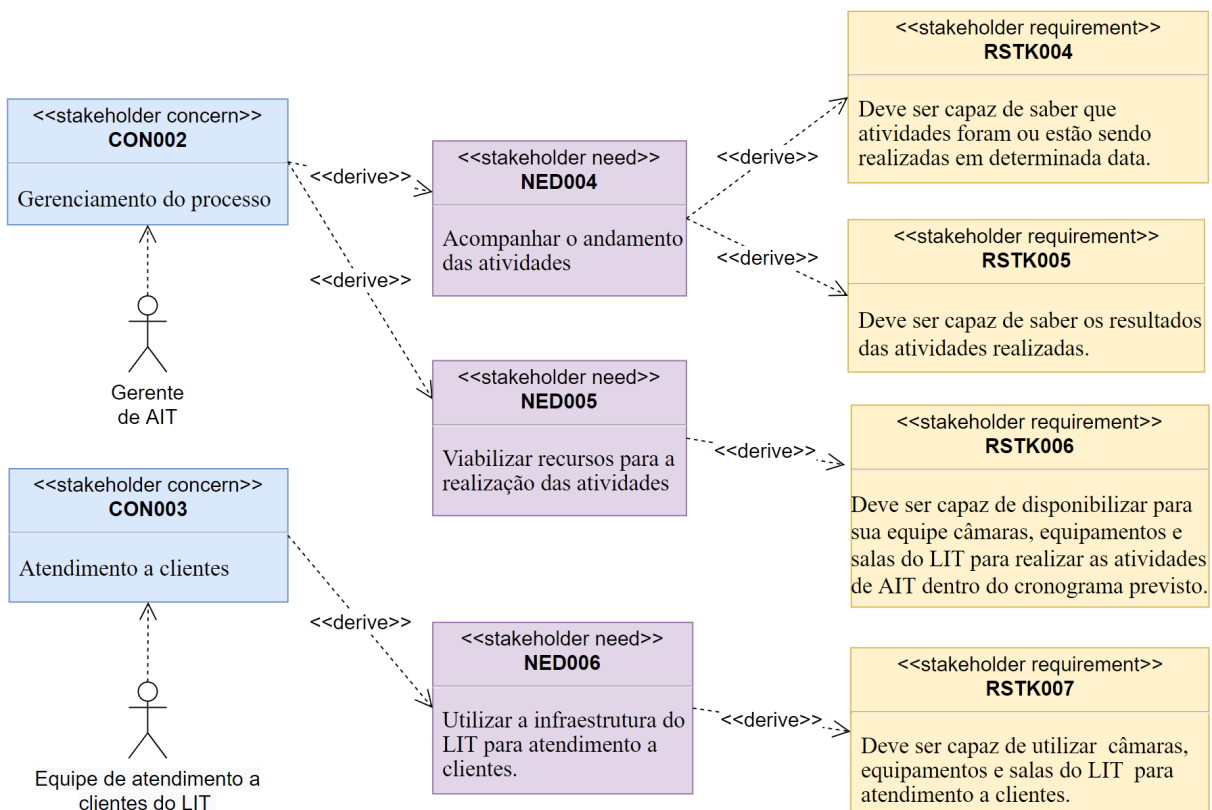


Figura 6. Desdobramento de *concerns* em requisitos de *stakeholder* (*stakeholders* da organização)



definidos para as medidas dos níveis de vibração devem estar acessíveis para que a Equipe de Ensaios Dinâmicos possa instalar e remover os acelerômetros, ou seja, não poderá haver barreira física, como cabeados ou equipamentos bloqueando o acesso a estes pontos.

Enquanto o engenheiro de sistemas trabalha para transformar os requisitos do *stakeholder* em requisitos de sistema, com base nas mesmas informações, o engenheiro de software trabalha para identificar Casos de Uso para o SI que dará suporte ao processo do ciclo de vida em questão. Se outros processos do ciclo de vida também serão contemplados pelo SI, requisitos dos *stakeholders* do sistema espacial em um determinado cenário podem também influenciar o SI nesses outros processos.

Por exemplo, para o requisito RSTK001, as informações das características físicas dos sistemas de vibração do LIT, assim como do sistema de fixação do satélite no veículo lançador (identificado em outro cenário, cenário de Lançamento), deverão estar disponíveis no SI para o projetista da interface mecânica. Para isso, um Caso de Uso para cadastro dos itens da infraestrutura e com campos para prover essas informações pode ser identificado. Nesse caso, requisitos do cenário de Ensaios Dinâmicos estão influenciando o SI que dará suporte ao processo de desenvolvimento.

Após o projeto e fabricação da interface mecânica, o correspondente procedimento para a fixação do satélite sobre a mesa vibratória também deverá estar disponível no SI.

Para os requisitos RSTK002 e RSTK003, funcionalidades para controle da instalação e remoção de acelerômetros devem estar disponíveis no SI. Uma visualização 3D da estrutura do satélite destacando os pontos de instalação dos acelerômetros e quais pontos estão com acelerômetros instalados também poderá ser provida pelo SI. Para prover essas funcionalidades um cadastro de acelerômetros é requerido, assim como da estrutura do satélite.

Agora, vamos analisar os requisitos dos *stakeholders* da organização, mostrados na figura 6. Para os requisitos RSTK004 e RSTK005, o engenheiro de sistema deverá se preocupar com o que o LIT deverá prover para que esses requisitos sejam satisfeitos, como entregar relatórios de resultados das atividades (ensaios, testes) após a conclusão de cada uma delas. Lembrando que a solução sistema para a SCE engloba produto e organização, nesse caso o requisito de sistema é um requisito a ser satisfeito pela organização LIT.

Para esses mesmos requisitos, o engenheiro de software consegue identificar que o SI deverá prover funcionalidades para o registro da execução das atividades e seus resultados, assim como para a visualização desses registros pelo gerente de AIT. Um Caso de Uso "Acompanhar plano de AIT" pode oferecer visualização gráfica do planejamento e andamento das atividades. Voltando ao diagrama da Figura 4, pode-se identificar que visualizar os resultados das atividades também é de interesse da Equipe de Engenharia, já que a conformidade do produto é uma preocupação para ela.

Já para os requisitos RSTK006 e RSTK007, o engenheiro de sistemas deverá se preocupar com requisitos de sistema que possibilitem o compartilhamento da infraestrutura entre as atividades de AIT e as atividades para clientes, como por exemplo, o LIT não poderá agendar serviços de ensaios dinâmicos para clientes nas datas previstas para os ensaios do satélite.

Baseado nesses requisitos, o engenheiro de software entende que o SI deverá prover funcionalidades para agendamento da utilização da infraestrutura e que essa agenda deve ser compartilhada entre a equipe de AIT e as equipes que atendem clientes do LIT. Para isso, o cadastro dos



itens da infraestrutura também é requerido.

A figura 7 mostra um diagrama com os Casos de Uso identificados pela análise do engenheiro de software.

Observe que o conhecimento de que o SI estará disponível para essas finalidades influencia o projeto da organização que poderá usar o SI como solução para satisfazer certos requisitos de *stakeholder*.



Figura 7. Casos de Uso do Sistema de Informação

3.3. Discussão

Sistemas de Informação tem papel importante na otimização de processos nas organizações. Devido à especificidade dos processos do ciclo de vida de satélites e das organizações que implementam esses processos, o desenvolvimento customizado desses sistemas mostra-se uma opção mais adequada.

Para o desenvolvimento de tais Sistemas de Informação é requerido um esforço significativo do engenheiro de software para entender esses processos e as necessidades dos futuros usuários do SI.

Neste artigo, pode-se observar que muitas das informações resultantes do esforço do engenheiro de sistemas na Análise de *Stakeholders* proposta pela SCE para o desenvolvimento do sistema espacial são também de interesse do engenheiro de software, o que sugere que um trabalho simultâneo e colaborativo pode economizar tempo e esforço de desenvolvimento.



Esse trabalho simultâneo e colaborativo mostra-se viável, uma vez que na SCE os processos do ciclo de vida do produto são analisados desde o início do projeto, disponibilizando informações em tempo hábil para desenvolvimento dos Sistemas de Informação.

Um outro ponto a ser observado é que o conhecimento prévio pelo engenheiro de sistemas dos Sistemas de Informação que vão apoiar a execução dos processos da organização pode influenciar o projeto funcional dessa organização, já que funções da organização poderão ser realizadas de forma automatizada pelo SI.

O desenvolvimento simultâneo mostra-se vantajoso, portanto, tanto para as atividades da Engenharia de Sistemas, quanto para as atividades da Engenharia de Software.

4. Conclusão

Este artigo avaliou a Análise de *Stakeholders* da Engenharia Simultânea de Sistemas como fonte de requisitos para o desenvolvimento de Sistemas de Informação que darão suporte aos processos do ciclo de vida do sistema espacial.

O resultado apresentado sinaliza que o desenvolvimento simultâneo e colaborativo do sistema espacial e de Sistemas de Informação para apoiar os processos do seu ciclo de vida é possível e pode economizar tempo e esforço de desenvolvimento.

Agradecimentos: *Agradecemos a colaboração dos tecnólogos do INPE, Homero Anchieta Furquim de Souza e Luiz Alexandre da Silva, que responderam nossas questões sobre o processo de AIT do satélite AMAZONIA 1 realizado no LIT, fornecendo conteúdo para nossa análise.*

Referências

- INPE. *Missão Amazonia*. 2021. Disponível em: <http://www3.inpe.br/amazonia-1>).
- LIT. *Ensaio Dinâmico*. 2021. Disponível em: <http://www.lit.inpe.br/pt-br/vibracao>).
- LOUREIRO, G. *A systems engineering and concurrent engineering framework for the integrated development of complex products*. Loughborough University, 1999. Disponível em: <https://hdl.handle.net/2134/10687>).
- LOUREIRO, G.; PANADES, W.; SILVA, A. Lessons learned in 20 years of application of systems concurrent engineering to space products. *Acta Astronautica*, v. 151, 05 2018.
- ROSING, M. v.; SCHEER, A.-W.; SCHEEL, H. v. *The Complete Business Process Handbook, Volume 1 - Body of Knowledge from Process Modeling to BPM*. [S.l.: s.n.], 2015. v. 1. ISBN 978-0127999593.
- SILVA, A. C. d. P.; LOUREIRO, G. The relationship between the model based systems engineering models and information systems to support space products lifecycle processes. In: *INTERNATIONAL ASTRONAUTICAL CONGRESS*. Bremen, Germany: [s.n.], 2018.
- STAIR, R. M.; REYNOLDS, G. W. *Principles of Information Systems*. 13th. ed. [S.l.]: Cengage Learning, 2018. ISBN 978-1-305-97177-6.
- TAN, V.; LABRADOR, J. L.; TALAMPAS, M. C. Mata: Mission, attitude, and telemetry analysis software for micro-satellites. In: *2020 IEEE REGION 10 CONFERENCE (TENCON)*. [S.l.: s.n.], 2020. p. 614–619. ISSN 2159-3450.

A.5 Iniciativas de adesão à MBSE na área espacial

Artigo apresentado no 11^o *Workshop* em Engenharia e Tecnologia Espaciais (WETE) em agosto de 2020.



INICIATIVAS DE ADESÃO À MBSE NA ÁREA ESPACIAL

Ma. Ana Claudia de Paula Silva¹, Dr. Geilson Loureiro²

¹Instituto Nacional de Pesquisas Espaciais – S.J.Campos, SP – Brasil
Aluna de Doutorado do curso de Engenharia e Gerenciamento de Sistemas Espaciais - CSE

ana.claudia@inpe.br

²Instituto Nacional de Pesquisas Espaciais – S.J.Campos, SP – Brasil
Laboratório de Integração e Testes - LIT

Resumo. *A Engenharia de Sistemas Baseada em Modelos (MBSE) é uma tendência mundial para projetar sistemas complexos. Este trabalho é uma pesquisa bibliográfica com o objetivo de buscar parâmetros para entender os desafios para a transição da abordagem de Engenharia de Sistemas tradicional, baseada em documentos, para a MBSE na área espacial. Foram identificadas e discutidas iniciativas de adesão à MBSE em duas das principais agências espaciais, NASA e ESA, e em duas das principais indústrias envolvidas no ramo espacial, Thales Alenia Space e Airbus Defense and Space, além de iniciativas no INPE (Instituto Nacional de Pesquisas Espaciais). A experiência desses grandes atores da área espacial mostra que essa não é uma transição fácil e rápida, porém os promissores benefícios da MBSE têm impulsionado muitas pesquisas e desenvolvimentos nessa área.*

Palavras-chave: Engenharia de Sistemas Baseada em Modelos; Engenharia de Sistemas Espaciais; MBSE

1. Introdução

A Engenharia de Sistemas Baseada em Modelos, ou *Model Based Systems Engineering* (MBSE), é uma tendência mundial para projetar sistemas complexos. Nessa abordagem, modelos são a principal fonte de informação e o principal meio de troca de informação entre os envolvidos no processo de engenharia, substituindo nessas funções os documentos utilizados na abordagem tradicional de engenharia de sistemas. Modelos são construídos para representar requisitos, estrutura e comportamento dos sistemas [Friedenthal et al. 2015].

Modelos podem ser criados em várias linguagens, sob a escolha da equipe. Uma das linguagens mais utilizadas atualmente é a SysML.

SysML é uma linguagem de modelagem gráfica de uso geral que pode ser base para diferentes métodos da MBSE [INCOSE 2015]. SysML suporta a especificação, análise, *design*, verificação e validação de sistemas [IBM 2012].

A MBSE traz benefícios como: melhoria na comunicação, redução no risco de desenvolvimento, qualidade aprimorada, aumento da produtividade e melhoria na transferência de conhecimento [Friedenthal et al. 2015].



A Engenharia de Sistemas de produtos espaciais é uma tarefa desafiadora. Considerando os potenciais benefícios da MBSE, produtos espaciais, por serem produtos complexos e com altos custos envolvidos em seu desenvolvimento, são bons candidatos para aplicação dessa abordagem.

Com base em pesquisas bibliográficas, neste trabalho, são apresentadas iniciativas de transição para MBSE em duas das principais agências espaciais, NASA (*National Aeronautics and Space Administration*) e ESA (*European Space Agency*), e em duas das principais indústrias envolvidas no ramo espacial, Thales Alenia Space e Airbus Defense and Space, além de iniciativas no INPE (Instituto Nacional de Pesquisas Espaciais) a respeito da MBSE.

O objetivo desta pesquisa é buscar parâmetros para entender os desafios para a transição da abordagem de Engenharia de Sistemas tradicional, baseada em documentos, para a abordagem de Engenharia de Sistemas Baseada em Modelos na área espacial.

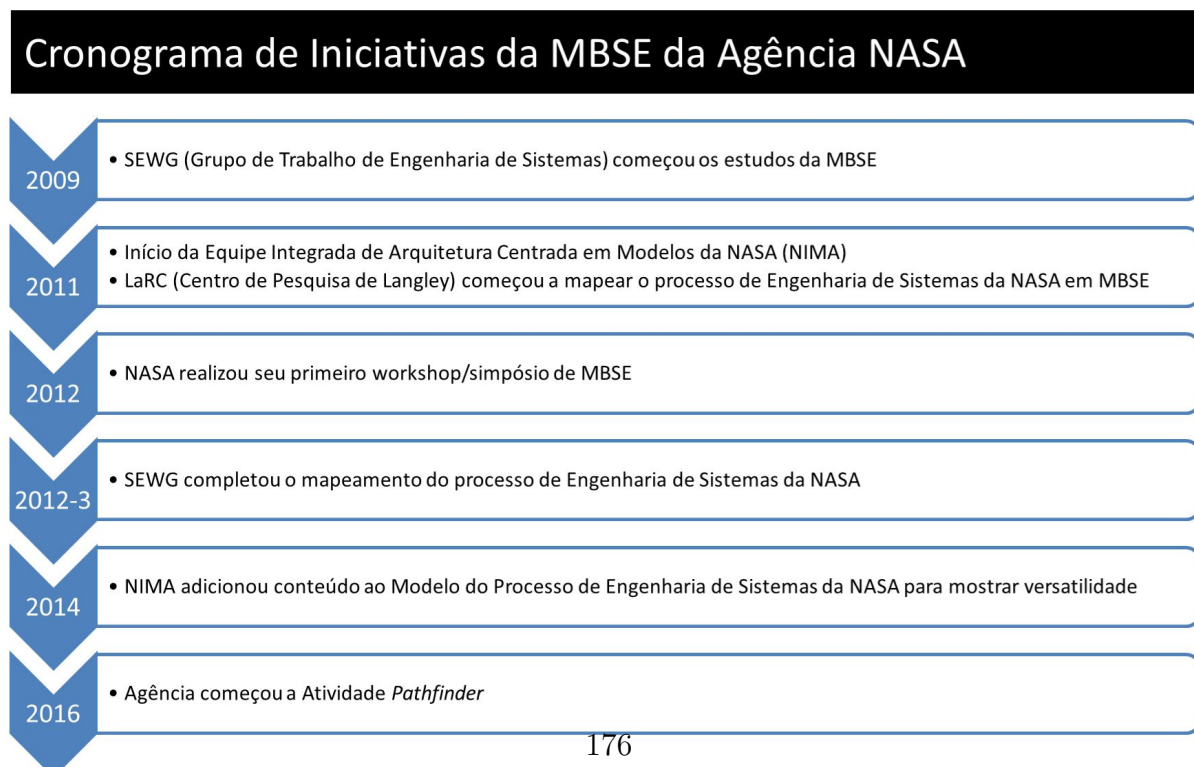
2. Metodologia

Esta é uma pesquisa bibliográfica. Por meio de pesquisa em artigos científicos de revistas e congressos e em publicações em sites e blogs oficiais das agências e indústrias abordadas neste artigo, foi realizado um estudo para identificação de iniciativas de transição da abordagem tradicional de Engenharia de Sistemas para a MBSE. As informações coletadas foram, então, comparadas e discutidas.

3. Resultados e Discussão

3.1. MBSE na NASA

A Figura 1 apresenta em uma linha do tempo as iniciativas da NASA relativas à MBSE.





O avanço do uso da MBSE nas aplicações da NASA e o desejo de capturar lições aprendidas para guiar os próximos passos motivaram o estabelecimento, em 2016, de um estudo chamado MBSE Pathfinder [Weiland and Holladay 2017].

Conduzido pelo Centro de Engenharia e Segurança da NASA, esse estudo avaliou a facilidade com que a MBSE poderia ser implantada e aplicada em quatro áreas de interesse. Quatro equipes trabalhando em paralelo por oito meses demonstraram potenciais benefícios da utilização da MBSE para seus stakeholders e para a NASA como um todo [Holladay et al. 2019], como mostrado na Tabela 1.

Tabela 1: Benefícios da MBSE demonstrados pelas equipes do Pathfinder 1 Fonte: Elaborada pela autora com as informações extraídas de [Holladay et al. 2019]

Área de interesse	Benefícios demonstrados
Uso e reuso da arquitetura da missão para uma campanha de missões humanas-para-Marte	A equipe demonstrou como os princípios de desenvolvimento ágil (semelhantes aos usados para engenharia de software) podem ser usados para criar modelos de sistema.
Fabricação aditiva para desenvolvimento de motores de foguetes.	A equipe gerou automaticamente requisitos e especificações de desempenho atualizados para permitir que os engenheiros rastreiem com mais facilidade as rápidas mudanças do sistema inerentes ao processo de fabricação aditiva.
Projeto de elemento de missão de um <i>Mars lander</i> (veículo para pousar em Marte).	A equipe gerou um subconjunto do pacote de revisão técnica para uma Revisão Preliminar de projeto.
Sombreamento do fluxo da missão de um projeto de foguete de sondagem.	A equipe modelou com sucesso as informações da Revisão de Projeto de uma missão anterior, descobrindo informações ausentes e inconsistências nos planos de teste.

A abordagem adotada para o MBSE Pathfinder forneceu uma série de lições aprendidas e construiu uma comunidade de usuários forte e colaborativa [Weiland and Holladay 2017].

Em 2017 foi estabelecida a parte 2 do MBSE Pathfinder com os objetivos de implementar as lições aprendidas na parte 1, focar no desenvolvimento de produtos de engenharia de sistemas ao longo do ciclo de vida, prover integração com várias ferramentas analíticas, como CAD, e demonstrar e comunicar valor para os projetos e Centros [Holladay et al. 2019].

Como na Parte 1 do MBSE Pathfinder, na Parte 2 os participantes também foram organizados em equipes virtuais representando vários centros da NASA. Dessa vez foram seis equipes em seis diferentes áreas de interesse. Quatro equipes continuaram o trabalho em relação ao ano anterior e duas novas equipes com dois novos tópicos foram acrescentadas: a equipe do ISRU (*in-situ resource utilization*) e Arquitetura de Habitat Integrada e a equipe do Adaptador de Carga SLS [Holladay et al. 2019].



Após essas duas experiências, a equipe de liderança do MBSE Pathfinder mudou o nome da iniciativa para Iniciativa de Infusão e Modernização de Engenharia de Sistemas Baseada em Modelo (MIAMI), a fim de distinguir seu escopo expandido do trabalho do MBSE Pathfinder de 2016 e 2017. O foco mudou de “preparação para transição” para “implementação”. Vários programas e projetos na NASA passaram a aplicar a MBSE [Holladay et al. 2019].

[Holladay et al. 2019] apresenta uma figura (Figura 2) com um resumo do planejamento para implantação da MBSE na NASA de 2016 a 2020. Na parte superior da figura estão as *tag lines* para cada ano e na metade superior são mostradas breves descrições da abordagem de implementação. As áreas do MBSE Pathfinder, na seção inferior, indicam a estreita relação com as missões reais da NASA. O centro da figura destaca parcerias com outras organizações. A parceria MBMA (*Model-based Mission Assurance*) é realizada com o Escritório de Segurança e Garantia de Missão da NASA, e a parceira DoD e *Digital Engineering Strategy Tiger Team* são parcerias com o Departamento de Defesa. O marco em 2020 é uma implantação direcionada de infraestrutura, processos e ferramentas, comunidade de usuários e projetos piloto.

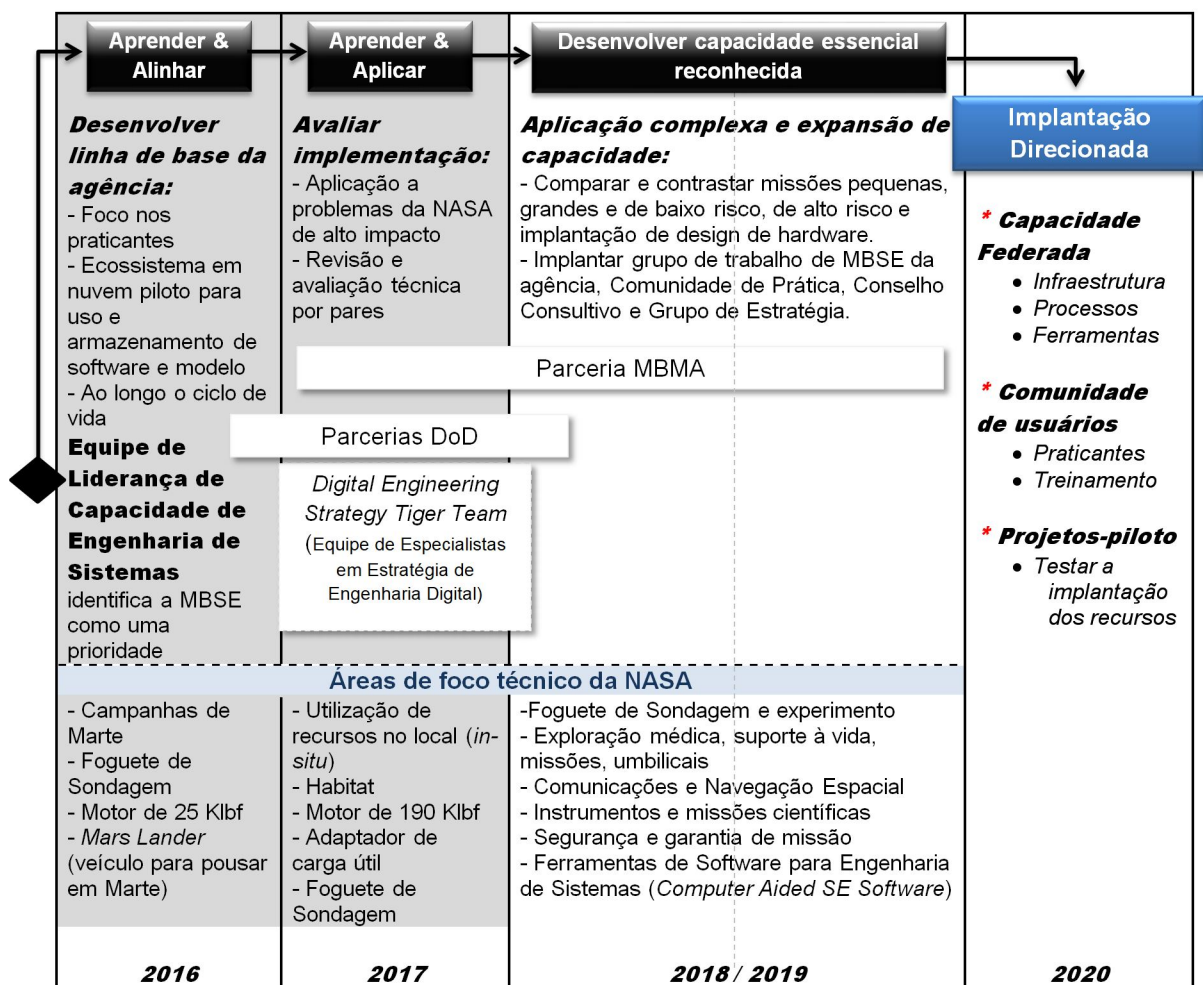


Figura 2. Resumo do Planejamento de Cinco Anos (2016 a 2020) da NASA para implantação da MBSE. Fonte: Traduzida de [Holladay et al. 2019]

3.2. MBSE na ESA

Na ESA a MBSE foi implementada como uma inovação dentro da missão, também inovadora, e.Deorbit, uma missão para remover detritos espaciais da órbita Sol-Síncrona entre 600 e 800km



de altitude. Considerada uma missão complexa envolvendo requisitos de robótica, orientação, navegação e controle, teve início em 2012, tendo um primeiro modelo, um modelo da arquitetura física, sido implementado durante a Pré-Fase A no CDF (*Concurrent Design Facility*) [WOLAHAN and BIESBROEK 2017], uma instalação de ponta onde especialistas de várias disciplinas aplicam métodos de engenharia simultânea ao projeto de futuras missões espaciais [ESA 2020].

Após o estudo de viabilidade interna, contratos industriais da Fase A foram firmados com três grandes Integradores de Sistemas (LSIs) europeus, Airbus Defense and Space, OHB e Thales Alenia Space. Como parte da Fase A, foi solicitado aos contratados usar MBSE para modelar as arquiteturas físicas e funcionais [Jessica 2017].

Após a conclusão bem-sucedida da Revisão Preliminar de Requisitos, foi recomendado estender o uso da MBSE também ao nível do sistema [Jessica 2017].

A Fase B1 começou em 2015 e terminou em 2017. Nessa fase foram feitos contratos com a Airbus e OHB, que foram incentivadas pela ESA a aplicar a MBSE sempre que possível. Cada contratado adotou diferentes ferramentas e metodologias para aplicação da MBSE, porém com os mesmos propósitos [Jessica 2017]:

- Analisar o problema e definir seus limites;
- Derivar e gerenciar os requisitos;
- Identificar os recursos/funções exigidos pelo sistema;
- Desenvolver a arquitetura funcional, lógica e física;
- Seguir os métodos de verificação/validação necessários;
- Iniciar simulações diretamente do modelo MBSE;
- Estabelecer uma troca de dados de “verdade única” no nível do sistema.

Esses foram os primeiros exemplos da ESA dos estudos da Fase B1, usando a MBSE, incluindo a arquitetura física da plataforma e a carga útil, juntamente com a arquitetura funcional e lógica da missão, com links diretos para algumas das ferramentas usadas para simulações [Jessica 2018].

3.3. MBSE na Thales Alenia Space

A Thales Alenia Space fornece soluções de alta tecnologia para telecomunicações, navegação, observação da Terra, gerenciamento ambiental, exploração, ciência e infraestrutura orbital [Thalesgroup 2020].

Em 2005 a Thales começou a investir na MBSE visando à melhoria do seu processo de engenharia. Investiu fortemente nos aspectos metodológicos e de ferramentas, tendo como parte dos resultados o método de desenvolvimento baseado em modelo chamado Arcadia e uma ferramenta de suporte a esse método, chamada Capella, que foi disponibilizada em 2014 em todas as unidades da Thales pelo mundo [Bonnet et al. 2016].

A solução Arcadia/Capella é inspirada nos conceitos da SysML, porém com meios de expressão reduzidos. Essa simplificação é possível devido ao seu escopo mais preciso. Arcadia/Capella não cobre todo o espectro das atividades de design, o foco está no projeto de arquitetura (justificação de componentes/interfaces por meio da análise funcional, avaliação não funcional antecipada da arquitetura e preparação das atividades de integração e validação), excluindo modelagem comportamental de baixo nível ou simulação [Bonnet et al. 2016] e [Eclipse 2017].

A simplificação ou especialização dos conceitos¹⁷⁹ provenientes da SysML foi pensada visando uma curva de aprendizado facilitada [Eclipse 2017] que pudesse atingir os engenheiros de sistemas que não possuem experiência em engenharia de software, já que as origens da SysML na



orientação-a-objetos mostraram-se claramente um obstáculo para sua adoção pelos engenheiros de sistemas da Thales que não eram familiares com o mundo de desenvolvimento de software [Bonnet et al. 2016].

O objetivo final da Arcadia/Capella não é ter especialistas em modelagem, mas sim promover uma mudança cultural, direcionada para a MBSE, nas práticas dos engenheiros de sistemas [Bonnet et al. 2016] e [Eclipse 2017].

Podemos dizer que alguns conceitos da solução Arcadia/Capella são especializações de conceitos da SysML para melhor representar a realidade da Thales. Foi feita uma interpretação da especificação SysML de forma a prover os engenheiros de todos os meios de expressão que eles precisassem, porém evitando sobrecarregá-los com complexidade desnecessária, ajudando-os a seguir as metodologias da Thales e unificando a maneira como as arquiteturas de software e de sistemas são modelados na Thales [Calio et al. 2016].

O metamodelo da Capella se diferencia da SysML em vários conceitos, sendo as duas principais diferenças o gerenciamento da análise funcional e o gerenciamento de tipos e instâncias [Bonnet et al. 2016].

Quanto ao gerenciamento da análise funcional, [Bonnet et al. 2016] coloca que a análise funcional não é estritamente suportada pela SysML, já que ela não define os conceitos de “funções” e “hierarquia de funções”. E que apesar da abordagem orientada a função poder ser implementada na SysML por meio do diagrama de atividades e dos diagramas de Blocos e de Blocos Internos (a árvore funcional sendo capturada como uma hierarquia de blocos, com o fluxo de dados sendo representado como diagramas de Blocos Internos), não é uma opção que atenda aos requisitos de simplicidade do Arcadia.

Quanto ao gerenciamento de tipos e instâncias, o conceito de Partes da SysML fica oculto na solução Arcadia/Capella, onde cada componente é considerado como uma instância por padrão, permitindo uma modelagem simplificada [Bonnet et al. 2016].

Para ilustrar as diferenças entre Arcadia/Capella e SysML, a seguir é apresentada uma comparação entre os dois principais diagramas de estrutura da SysML e os diagramas oferecidos pela solução Arcadia/Capella [Eclipse 2017]. Uma comparação mais completa é apresentada por [Eclipse 2017].

3.3.1. Diagrama de Blocos SysML x Arcadia/Capella

Arcadia/Capella utiliza dois diagramas para representar o que a SysML representa no seu diagrama de Blocos:

- o Diagrama de Detalhamento de Componentes que mostra a hierarquia de componentes através de uma árvore gráfica; e
- o Diagrama de Interface de Componente que mostra o relacionamento de composição entre componentes por meio de contenção gráfica e relacionamentos entre componentes e interfaces através de portas. Nesse diagrama as propriedades do componente não são exibidas graficamente.

3.3.2. Diagrama de Blocos Internos SysML x Arcadia/Capella

O Diagrama de Blocos Internos da SysML captura a estrutura interna de um bloco em termos de propriedades e conectores entre propriedades.



Esse diagrama é dedicado a modelar a estrutura interna de um bloco. Um Bloco pode ser decomposto em Partes, que são instâncias de outros Blocos. Blocos são “definições” e Partes são instâncias dessas definições.

Um Bloco que define uma bicicleta, por exemplo, tem duas Partes: “roda dianteira” e “roda traseira”, que são definidas, ou tipificadas, pelo Bloco “roda”. A definição de “roda” pode ser reutilizada muitas vezes no sistema através do conceito de Parte.

Apesar desse paradigma bloco/parte também poder ser utilizado na solução Arcadia/Capella, por padrão Arcadia/Capella é configurado para uma modelagem controlada por instância, ou seja, componentes e funções são considerados por padrão como instâncias ou usos.

Essa abordagem baseou-se no retorno proveniente da experiência dos engenheiros de sistemas com a modelagem. Eles não se mostraram necessariamente confortáveis com o fluxo de trabalho de criação de elementos de definição primeiro (“blocos” ou “componentes”) para depois referenciá-los a partir de elementos de uso específico (“partes”). Além disso, distinguir as diferentes ocorrências de cada elemento e fornecer propriedades ou valores para cada ocorrência é fundamental para realizar análises não funcionais, como análise de segurança, que fazem parte do projeto arquitetônico da Capella.

3.4. MBSE na Airbus Defence and Space

Segundo [Sharples 2018], a Airbus Defense and Space tem uma rigorosa abordagem de MBSE que é utilizada em todos os seus projetos. E, segundo [Gregory et al. 2020], a Airbus liderou vários projetos buscando desenvolver técnicas de MBSE e aplicá-las a projetos do mundo real, sendo o mais significativo deles o desenvolvimento do modelo em SysML da missão e.Deorbit da ESA.

Em 2012 a Airbus sentiu a necessidade de uma ferramenta para dar suporte aos novos métodos baseados nos princípios da MBSE. Anteriormente, contava com ferramentas COTS disponíveis, que considerava bem estabelecidas e com bons resultados, porém complexas o suficiente para limitar sua implantação. A nova ferramenta deveria ser essencialmente flexível o suficiente para se adaptar ao vocabulário e aos conceitos da Engenharia de Sistemas da Airbus e para se ajustar facilmente às necessidades em evolução, à medida que os processos e métodos amadureciam gradualmente [Magnet 2016].

Para suprir essa necessidade, a Airbus desenvolveu e implantou uma ferramenta customizada, baseada na ferramenta de código aberto Eclipse Papyrus [Eclipse 2020] que fornece, entre outras coisas, suporte completo a SysML [Magnet 2016].

A Papyrus ofereceu a capacidade de criar rapidamente uma ferramenta personalizada que restringe alguns dos recursos gerais da SysML permitindo que os usuários trabalhem apenas com conceitos específicos do domínio. A linguagem SysML foi, portanto, personalizada para um formato mais facilmente compreensível pelos projetistas [Magnet 2016].

A ferramenta desenvolvida, chamada FAST, dá suporte às análises funcional e operacional, utilizando os diagramas de Atividades e de Blocos Internos da SysML fornecidos nativamente pela Papyrus. FAST permite a definição das funções necessárias para operar o sistema, bem como sua decomposição funcional e suas interfaces [Magnet 2016].

A extensibilidade geral da Papyrus também permitiu a inclusão de outras ferramentas, como verificações internas de consistência e geração automática de documentos [Magnet 2016].



3.5. MBSE no Instituto Nacional de Pesquisas Espaciais (INPE)

No INPE, uma metodologia de Engenharia de Sistemas Baseada em Modelos foi adotada na missão EQUARS (*Equatorial Atmosphere Research Satellite*) em desenvolvimento. EQUARS é uma missão que tem como objetivo “*promover o avanço do conhecimento científico em Aeronomia Equatorial, com ênfase no entendimento dos fenômenos físicos que perturbam o comportamento médio do plasma ionosférico*” [Hoffmann et al. 2018].

Além da aplicação prática na missão EQUARS, vários projetos desenvolvidos no INPE foram e estão sendo utilizados como base para pesquisas envolvendo a MBSE. Algumas dessas pesquisas são citadas a seguir:

- [Coicev and Loureiro 2019] apresenta uma proposta de utilização de MBSE e SysML aplicados a um estudo de caso de análise de um componente de um Equipamento de Suporte Elétrico de Solo (EGSE) típico utilizado na Montagem, Integração e Testes (AIT) de satélites com o objetivo de descrever o fluxo de processos utilizado nas análises, fornecendo um pano de fundo metodológico para a aplicação da notação SysML de forma prática para desenvolvimento de EGSEs.
- [Franco 2018] propõe um modelo de referência da interface entre o satélite e o veículo lançador, em linguagem SysML, para servir como base a futuros esforços na utilização da MBSE para definição e controle desta interface.
- [Aquino et al. 2018] realizou um projeto de Iniciação Científica que tem como objetivo a implementação da MBSE para o nano satélite AESP-14, primeira plataforma CubeSat brasileira, desenvolvido pelo ITA (Instituto Tecnológico de Aeronáutica) com apoio do INPE.
- [Silva and Loureiro 2018], usando como exemplo o processo de Montagem, Integração e Testes (AIT) de satélites, fez uma análise da possível colaboração entre as atividades de modelagem da MBSE e as atividades de desenvolvimento de Sistemas de Informação para apoiar os processos do ciclo de vida de um produto espacial.
- [Burger 2018] apresenta em sua tese de doutorado um framework conceitual que considera o uso da MBSE para fornecer entradas para o planejamento da Montagem, Integração e Testes (AIT) de satélites.

3.6. Discussão

Esforços das grandes agências e indústrias da área espacial para implementar a MBSE sinalizam a confiança em que seus benefícios são promissores. Porém as experiências da NASA, ESA, Airbus e Thales Alenia Space mostram também que essa não é uma transição fácil e rápida. É um projeto de anos que envolve esforços de convencimento e adaptação.

A escolha da metodologia e das ferramentas para dar suporte à metodologia escolhida mostrou-se bem particular para cada organização. Embora existam ferramentas comerciais, o desenvolvimento customizado que busca preservar a cultura de engenharia de cada organização parece ser uma tendência. Mesmo a consagrada linguagem SysML ainda não representa um ponto de intersecção entre os engenheiros das diversas organizações. Nas experiências da Airbus e Thales Alenia, por exemplo, podemos perceber que optaram por desenvolvimento de ferramentas customizadas que adaptaram a linguagem SysML para seus processos de engenharia. Ao menos no caso da Thales Alenia, ficou claro que a opção pela customização se deu como uma forma de diminuir a distância entre a realidade da formação de seus engenheiros e a nova abordagem proposta. Esse é um ponto a ser pensado, se o mais adequado seria mesmo uma customização da linguagem, o que garantiu uma adesão mais suave e rápida, ou se o mais adequado seria que



os engenheiros aderissem à visão orientada a objetos que é a base da SysML, o que garantiria a longo prazo uma unicidade entre os modelos das várias organizações.

Percebemos que a customização aconteceu de forma distinta, cada organização de acordo com sua realidade prática. Visto que um projeto espacial normalmente é feito em parceria com outras organizações, o uso de diversas linguagens e ferramentas em diferentes fases ou em diferentes subsistemas, a depender da organização responsável pela etapa/subsistema, pode ser um fator que dificulte a efetiva migração para MBSE de todo o ciclo de vida de um produto espacial de forma integrada.

A despeito disso e de todas as dificuldades e desafios dessa transição para MBSE, os promissores benefícios têm impulsionado uma série de pesquisas e desenvolvimentos nessa área.

4. Conclusão

Este trabalho apresentou iniciativas de alguns dos principais atores da Engenharia de Sistemas Espaciais no que diz respeito à migração de uma abordagem tradicional de engenharia de sistemas, baseada em documentos, para a abordagem baseada em modelos, conhecida como MBSE (*Model Based Systems Engineering*). A customização de metodologias e ferramentas, para aproximá-las da realidade de cada organização, parece uma tendência que garante um caminho mais suave de transição, porém pode ser um dificultador para a adesão à MBSE de forma integrada em todo ciclo de vida de um projeto espacial, que normalmente envolve desenvolvimento em parceria com várias organizações. Embora a transição para MBSE na área espacial seja desafiadora, os benefícios promissores têm impulsionado pesquisas e desenvolvimentos na área.

Referências

- Aquino, E. R., Loureiro, G., and Burger, E. E. (2018). Mbse para engenharia de sistema de cubesats. In *SEMINÁRIO DE INICIAÇÃO CIENTÍFICA E INICIAÇÃO EM DESENVOLVIMENTO TECNOLÓGICO E INOVAÇÃO (SICINPE)*, São José dos Campos, SP.
- Bonnet, S., Voirin, J., Exertier, D., and Normand, V. (2016). Not (strictly) relying on sysml for mbse: Language, tooling and development perspectives: The arcadia/capella rationale. In *2016 Annual IEEE Systems Conference (SysCon)*, pages 1–6.
- Burger, E. E. (2018). *A conceptual MBSE framework for satellite AIT planning*. Doutorado em engenharia e gerenciamento de sistemas espaciais, Instituto Nacional de Pesquisas Espaciais (INPE).
- Calio, E., Giorgio, F. D., and Pasquinelli, M. (2016). Deploying model-based systems engineering in thales alenia space italia. In *CIISE - 2nd INCOSE Italia Conference on Systems Engineering*, pages 112–118, Turin, Italy.
- Coicev, M. and Loureiro, G. (2019). Mbse i&’ sysml applied to the development of egse for sattelites assembly, integration and testing (ait) - a practical case. *International Journal of Advanced Engineering Research and Science*, 6:391–397.
- Eclipse (2017?). Equivalences and differences between sysml and arcadia/capella. Acesso em: 29 mai. 2020.
- Eclipse (2020?). Eclipse papyrus. Acesso em: 29 mai. 2020.
- ESA (2020). What is the cdf? Acesso em: 29 mai. 2020.



- Franco, R. (2018). *Um modelo de referência MBSE da interface entre satélite e um veículo lançador escolhido usando AHP*. Mestrado em engenharia e gerenciamento de sistemas espaciais, Instituto Nacional de Pesquisas Espaciais (INPE).
- Friedenthal, S., Moore, A., and Steiner, R. (2015). *A Practical Guide to SysML: The Systems Modeling Language*. Elsevier, 3th edition.
- Gregory, J., Berthoud, L., Tryfonas, T., Rossignol, A., and Faure, L. (2020). The long and winding road: Mbse adoption for functional avionics of spacecraft. *Journal of Systems and Software*, 160. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0164121219302274>. Acesso em: 29 mai. 2020.
- Hoffmann, L. T., Branco, M. S. A., Branco, R. H. F., Gobbi, D., Silva, C. M. Z., and Perondi, L. F. (2018). A engenharia de sistemas espaciais da missão equars. In *SIMPÓSIO BRASILEIRO DE GEOFÍSICA ESPACIAL E AERONOMIA*, Santa Maria, RS.
- Holladay, J. B., Knizhnik, J., Weiland, K. J., Stein, A., Sanders, T., and Schwindt, P. (2019). Mbse infusion and modernization initiative (miami): “hot” benefits for real nasa applications. In *2019 IEEE Aerospace Conference*, pages 1–14.
- IBM (2012). *Essentials of IBM Rational Rhapsody for Systems Engineers v.7.6.1 - Student Manual vol 1*.
- INCOSE (2015). *Incose Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*. Wiley, 4th edition.
- Jessica (2017). Applying mbse to a space mission. Acesso em: 29 mai. 2020.
- Jessica (2018). From active debris removal to in-orbit servicing: The legacy of e.deorbit. Acesso em: 29 mai. 2020.
- Magnet, A. (2016). Designing freedom of flight. Acesso em: 29 mai. 2020.
- Parrott, E. (2016). The value of successful mbse adoption. Disponível em: <https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20170001634.pdf>. Acesso em: 29 mai. 2020.
- Sharples, R. (2018). *Continuous Model Based System Engineering (MBSE) Improvement via Human System Integration and Customer Change*, pages 49–54.
- Silva, A. C. d. P. and Loureiro, G. (2018). The relationship between the model based systems engineering models and information systems to support space products lifecycle processes. In *INTERNATIONAL ASTRONAUTICAL CONGRESS*, Bremen, Germany.
- Thalesgroup (2020). Space for life. Acesso em: 29 mai. 2020.
- Weiland, K. J. and Holladay, J. (2017). Model-based systems engineering pathfinder: Informing the next steps. *INCOSE International Symposium*, 27(1):1594–1608. Disponível em: <https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20180006165.pdf>. Acesso em: 29 mai. 2020.
- WOLAHAN, A. and BIESBROEK, R. (2017). Model based systems engineering applied to esa’s e.deorbit mission. In *68th International Astronautical Congress 2017*, Adelaide, Australia.

A.6 The relationship between the model based systems engineering models and information systems to support space products lifecycle processes

Artigo apresentado no *International Astronautical Congress* (IAC) em outubro de 2018.

IAC-18,D1,4A,4,x45081

The relationship between the Model Based Systems Engineering models and Information Systems to support space products lifecycle processes

Silva, A.^{a*}, Loureiro, G^b

^a *Brazilian Institute for Space Research (INPE), Av. dos Astronautas, 1758, São José dos Campos, Brazil, 12227-010, ana.claudia@inpe.br*

^b *Brazilian Institute for Space Research (INPE), Av. dos Astronautas, 1758, São José dos Campos, Brazil, 12227-010, geilson@lit.inpe.br*

* Corresponding Author

Abstract

Motivated by the needs of the Integration and Testing Laboratory (LIT) of the Brazilian Institute for Space Research (INPE), this paper aims to analyze how the information of space products life cycle processes is determined from the models used for their development, considering a model-based approach. With a focus on Space Systems Engineering, exemplified by the satellite Assembly, Integration and Testing (AIT) process, we analyzed the relationship between Systems Engineering models with a future Information System to support the product lifecycle process. There are various business processes to perform the satellite life cycle phases. Integrated with each other, they deliver the final result. A business process is a set of activities or related tasks that are performed to deliver an expected result. Information Systems are a good way to improve the business processes performance. AIT is one of the satellite life cycle processes. Doing AIT is so complex and a lot of equipment, people, tasks and information are involved in it. So far, at LIT/INPE there is not an Information System that supports the whole AIT process practiced there, and the storage and exchange of AIT information are document based. The use of documents as a primary source of data makes it difficult to retrieve information during the process, requires a lot of effort from the team involved in it, and sometimes causes delays in activities. There is a worldwide trend towards a shift from the document-centric to the model-centric approach for engineering complex systems. Model Based Systems Engineering (MBSE) improves Systems Engineering practices using models to represent the system in various aspects. Models are built to represent requirements, structure and behaviour of systems. These models, among other advantages, become consistent and reliable sources of information to support the various phases of the product lifecycle. Concerning LIT's case, the MBSE benefits as well as the adoption of Information Systems can improve the AIT process. We must consider that in order to build good models that faithfully represent a system it is necessary effort and skilled workforce. Equivalent effort and workforce are needed to build Information Systems that faithfully meet the needs of a business process. There is a relationship between these two activities and collaboration between them can be helpful to reduce development effort.

Keywords: MBSE, Information Systems, Space Products, SysML, UML

Acronyms/Abbreviations

AIT: Assembly, Integration and Testing
BDD: Block Definition Diagram
CBERS: China-Brazil Earth Resources Satellite
IBD: Internal Block Diagram
INPE: Brazilian Institute for Space Research
IS: Information System
LIT: Integration and Testing Laboratory
MBSE: Model Based Systems Engineering
SysML: System Modelling Language
UML: Unified Modelling Language

Testing (AIT) is one of these phases. Doing AIT involves a lot of equipment, people, tasks and information, what makes it a complex process.

Currently the Integration and Testing Laboratory (LIT) of the Brazilian Institute for Space Research (INPE) is working on AIT of two satellites, Amazonia-1 and CBERS 4A.

The AIT team at LIT/INPE uses a series of tools to help them to manage the many tasks they have to perform and all information that flows through the whole AIT process. So far, at LIT/INPE the storage and exchange of AIT information are document based. There are too many documents involved in the AIT process. The use of documents as a primary source of data makes it difficult to retrieve information during the

1. Introduction

There are various business processes to perform the satellite life cycle phases. Assembly, Integration and

process, requires a lot of effort from the team involved in it, and sometimes causes delays in activities.

Motivated by the need to improve the way the information is stored, manipulated and exchanged during the AIT process and taking advantage of the AIT activities that are taking place there, LIT/INPE started the project of an Information System (IS) to support the AIT process. This project is led by the Information Systems Development team at LIT, which has a close collaboration of the AIT team.

As well as adopting an IS to support the business process of the AIT phase, using a model-based approach to engineer the satellites that arrives at LIT for AIT would also bring great benefits to the LIT activities and this is worth considering as a future project.

In this context, being aware that building good models that faithfully represent a system demands effort and skilled workforce and that equivalent effort and workforce are needed to build Information Systems that faithfully meet the needs of a business process, this work begins a study on the relationship between these two activities, using the satellites AIT process as example. This study aims to analyze how the information of space products life cycle processes is determined from the models used for their development, considering a model-based approach.

Many of the current works in the area of Space Systems Engineering are looking to improve the practice of Systems Engineering by exploring Model Based Systems Engineering (MBSE). In the space community, we find works that focus on the construction of methods, languages and tools to model complex systems such as [1] and [2]; we also find works applying some language, tool or methodology to model the structure and/or behaviour of some specific space product and its subsystems, such as [3] and [4]. This paper has another focus. This paper focuses on the identification of the relationship between the modelling made from the perspective of the systems engineer, who is concerned with building the space product, and the modelling made from the perspective of the software engineer who is concerned with building an IS to support the product lifecycle business processes.

Section 2 presents important concepts for understanding the article. Section 3 presents, at a high level, the business process to perform the AIT activities at LIT. Section 4 presents the relationship between an IS system model for LIT's AIT business process case, and a simplified engineering model of a satellite assembled, integrated and tested at LIT. Section 5 discusses the results of this research. Section 6 presents the next steps to continue this research, and section 7 concludes the article.

2. Important concepts

2.1 MBSE

MBSE is a world trend to engineer complex system. It improves Systems Engineering practices using models to represent the system in various aspects.

In the MBSE approach, domain models are the main source of information and the main means of exchanging information among those involved in the engineering process, rather than exchanging information based on documents. Models are built to represent requirements, structure and behaviour of systems [5]. These models become consistent and reliable sources of information to support the various phases of the products lifecycle. MBSE brings benefits like enhanced communications, reduced development risk, improved quality, increased productivity and enhanced knowledge transfer [6].

Models can be created in several languages at the direction of the team. One of the most widely used languages nowadays is SysML.

2.2 Business Processes and Information Systems

A business process is a set of activities or related tasks that are performed to deliver an expected result, as a product or a service [7]. A good way to improve business processes performance is adopting Information Systems to support them. An IS is a system that collect, process, store, and disseminate data and information and that provide a feedback mechanism to monitor and control its operation [8]. An adequate IS facilitates access to information and eliminates possible inconsistencies and the need to redo tasks related to information management.

The main components of Information Systems are custom built software. The first thing one needs to build this kind of software is to understand the business process that will be supported by it and the information that flows in such process. In some points of the process certain information will be required by the IS for that, in other points, it can be provided to be used, for example, to assist decision-making or to perform other operations.

2.3 UML

UML is a visual language, adopted internationally, much used to model software systems based on the object orientation paradigm [9]. UML in its version 2.5 has 14 diagrams, some to model the structure of the systems and some to model the behaviour of the systems [10]. The Class Diagram and the Object Diagram from UML are used in this paper. They are structural UML diagrams.

A Class specifies a classification of Objects and the features that characterize the structure (attributes) and behaviour (operations) of those Objects. Objects of a Class contain values for each attribute of the Class, according to the characteristics of the attribute, for example its type and multiplicity [10].

A Class Diagram presents Classes of the system and the relationship between them. An Object Diagram shows Objects and data values; it corresponds to an instance of the Class Diagram, showing the state of the system at a given point in time [11].

2.4 SysML

SysML is a graphical modelling language, an extension of UML, used to model complex systems. It is a general-purpose modelling language that can support many different MBSE methods [5]. SysML supports the specification, analysis, design, verification and validation of systems [12].

SysML defines three categories of diagrams: structural, that describes the static structure of a system, behavioural, that shows the dynamic view of the systems, and Cross-Cutting, that enables the modelling of relationships across both structural and behavioural elements [12].

The Block Definition Diagram (BDD) and the Internal Block Diagram (IBD), that are used in this paper, are used to represent the structure of the system.

A Block is the basic architectural unit in SysML and, as UML Classes, it may include both structural and behavioural features. Blocks have structural properties named Parts. While Block is the definition of an element, Part is a usage or instance of that element. [12].

SysML BDD focus on definition and captures the system hierarchy and the definition of a Block in terms of its properties, operations and references. It shows high-level structure but also external interfaces and flow [12].

SysML IBD focus on usage and describes the internal structure of the system in terms of its Parts, Ports, and Connectors [5].

3. LIT's AIT Business Process and its information

The satellite life cycle processes are proposed by NASA and ESA. According to ESA [13], the satellite life cycle phases are:

- Phase 0: Mission Analysis/Needs Identification;
- Phase A: Feasibility;
- Phase B: Preliminary Definition (Project and Product);
- Phase C: Detailed Definition (Product);
- Phase D: Production / Ground Qualification Testing;
- Phase E: Utilization;
- Phase F: Disposal.

According to the nomenclature adopted by ESA, the AIT activities of a satellite take place during Phase D of the product life cycle.

There is a business process to perform each satellite life cycle phase, and the execution of these various business processes lead to the expected result.

The AIT process is complex, and many activities are carried out so that its purpose is fulfilled. Aiming to build an IS to support LIT's AIT business process, a initial model for the LIT's AIT business process was built, according to the high-level activities pointed out by the LIT's AIT team. This model starts with 5 macro processes: *AIT Definition*, *AIT Preparing*, *AIT Management*, *AIT Execution* and *Resources Control* [14]. Each of these macro processes broken down into lower level processes or activities. A brief description of each of these macro processes follows.

- **AIT Definition:** Here one defines the activities that will be executed (from the macro activities, through all the levels of unfolding, to the level of procedure) and the sequence in which these activities are to be executed.
- **AIT Preparing:** Here one defines the information that the operator needs to perform each activity, such as setup, required items, and step-by-step operations.
- **AIT Management:** Here the AIT manager organizes the execution of tasks, prepares the schedule, releases tasks for execution and deals with the issues and problems that occur during the process.
- **AIT Execution:** Here the tasks are effectively carried out and records are made about the activity itself (such as executor and date of execution), about the laboratory conditions (such as temperature, humidity and contamination) and about interurrences (as non-conformities and unexpected events). During the AIT Execution, lessons learned should be recorded.
- **Resources Control:** Here, the resources needed for the AIT process are controlled (such as materials, tools, GSEs, staff).

From these macro processes, the high-level information that is generated and used by each of them was identified, arriving at a set of information that is important for the AIT process. Fig. 1 summarizes it. Like the macro processes, certainly this high-level information can also be broken down in more detail.

4. The relationship between a IS model and models of MBSE

As a basis for the comparison between the models, part of an IS model for the LIT's AIT business processes, and two simplified engineering models of the Amazonia-1 satellite are used.

Amazonia-1 is the first Earth observation satellite entirely designed, integrated, tested and operated by Brazil. It is a polar orbiting satellite that will generate images of the planet every 5 days. Its objective is to

provide data for environmental monitoring, especially in the Amazon region [15].

4.1 Modelling an IS for LIT's AIT business process

Currently, the information arriving from Phase C to LIT, shown in Fig. 1, comes in documents. LIT's AIT process also happens based on documents and there is no IS to support the whole process. However, considering an IS for this purpose, the information that arrives at LIT, as well as the information generated at LIT itself, would have to be inserted in this IS so that it could be provided by the IS throughout the process.

This IS would have to be structured so that this information could be entered and retrieved.

To exemplify how part of a model of an IS for meeting the needs of the LIT's AIT process would be, consider the *satellite log* information. As shown in Fig. 1, this useful information flows from the macro process *AIT Execution* towards the macro process *AIT Management*.

Satellite log is high-level information. Included in its decomposition, among many other items of information, are those about assembly and disassembly of parts of the satellite, that will be used as an example.

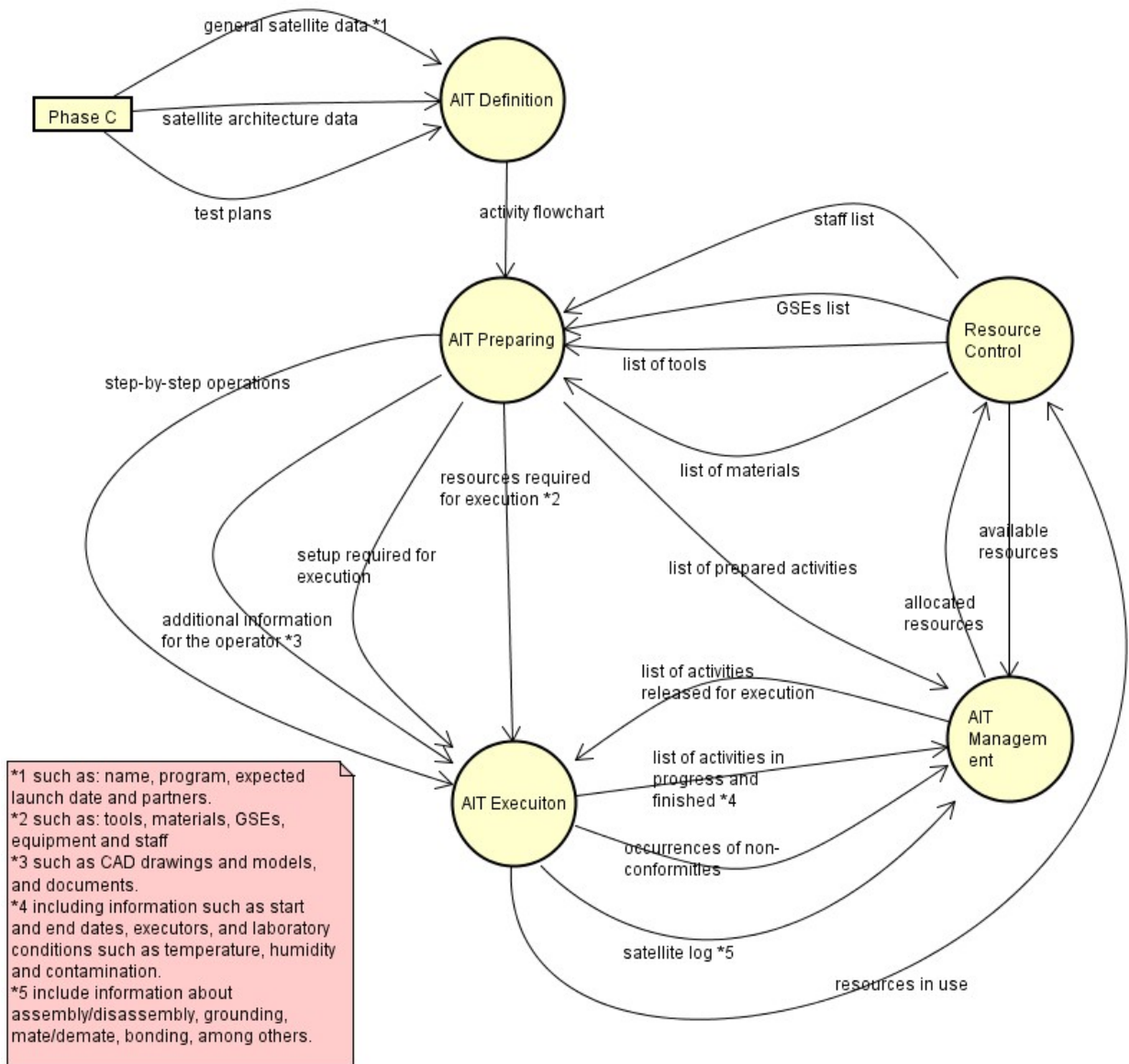


Fig. 1. LIT's AIT macro processes and their information

One of the questions that the IS should answer to the AIT manager, in the macro process *AIT Management*, is what is already assembled on a given subsystem or what was assembled on a certain date. This is important for the manager to let him know, for example, which subsystem can already be tested or so that he, or others, can analyse fault situations.

For the IS to provide this answer, three activities must be performing within the *AIT Execution* macro process, two of them are *register assembly* and *register disassembly*.

When some equipment is assembled or disassembled on the satellite, the operator shall record in the IS that this has occurred. This equipment must be one that was received at LIT for AIT and must be associated in the IS with the satellite part to which it corresponds. For this association to be possible, a third activity must happen first: *register entry of equipment into LIT*. In this activity the received equipment is registered in the IS with its proper identification (Part Number, Serial Number), among other information. Also, the satellite parts need to be previously registered in the IS, as well as the satellite subsystems, and the relationship between them. Fig. 2 illustrates this small part of the AIT business processes model.

For the IS to be able to support this process and to answer the question of the AIT manager, a structure such as that shown in Fig. 3 is required.

Fig. 3 shows a UML Class Diagram representing part of a software structure for an IS for the LIT's AIT business process. According to the UML notation, in Fig. 3 the boxes represent the Classes. Their first compartment is used to name the Class and the second one is used to show its attributes. The lines connecting the Classes represent the relationship between them. The number, and the "*" (which means several) at the ends of the lines indicate the number of possible instances in the relationship, for example, a satellite model may have multiple subsystems and a subsystem must belong to a satellite model.

4.2 MBSE approach to engineering the satellite Amazonia-1

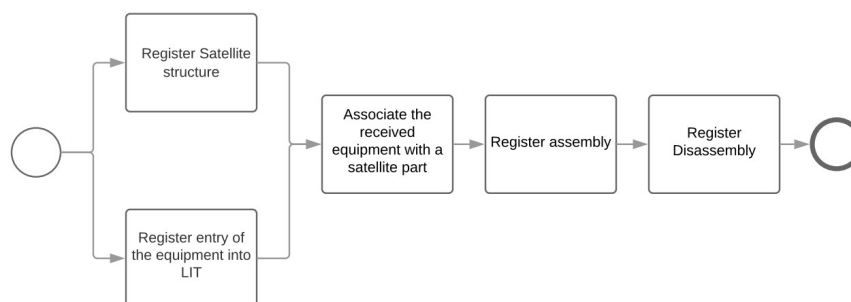


Fig. 2. Part of the LIT's AIT business process

As mentioned earlier, information for the AIT process currently arrives at LIT by documents rather than by models. Taking MBSE as the way to engineering a system, models are built to represent system requirements, structure and behaviour. Considering a model-based approach rather than the document-based approach to engineering the space products that will be assembled, integrated and tested at LIT, information for AIT would arrive in models, for example in SysML models.

Taking the Amazonia-1 satellite as an example, it could be represented in a SysML BDD modelled as a composition of its subsystems, and its subsystem could be represented as a composition of its parts. A simplified example of a BDD for the electrical model of the Amazonia-1 satellite is showed in Fig. 4. Fig. 4 shows three of the subsystems of the Amazonia-1: *AOCS*, *OBDH* and *PSS*; and some of the parts of these subsystems. According to the SysML notation, in Fig. 4 the boxes represent the Blocks, and their first compartment is used to name them. The lines with a solid diamond at one end connecting Blocks represent a Part Association, that are commonly used to decompose a Block into its Parts. The numbers in the other end of the lines represent the number of instances or usages of the related Block. For example, the *PSS* subsystem are decomposed into 1 *PCDU*, 1 *SADE*, 2 *SADA* and 4 *Batteries*.

In a model-based approach, using SysML, Internal Block Diagrams could be used to represent the internal structure of each subsystem, describing how parts and external references are related in the context of the subsystem Block. For this purpose, instances of the subsystem parts would appear in the IBD, as exemplified in Fig. 5, which shows instances of the *RW* Block and *ACE* Block in an IBD of the *AOCS* subsystem. According to the SysML notation, in an IBD the boxes represent the Parts, and in their first compartment appear the name of the Part and the Block that the Part is a type of, in the format: *Part name: Type name*.

4.3 A relationship between the two models

We have seen, as an example, that information on the assembly and disassembly of the satellite parts are important for the AIT manager and should be provided to him by the IS. We have also seen which tasks of the process are required for this information to be available in the IS: *register satellite structure*, *register receipt of equipment*, *register assembly* and *register disassembly*. Fig. 3 shows a model of the Software Engineering domain that allows the IS to support this process.

We can see that the satellite subsystems and the satellite parts modelled in the Class Diagram (Fig. 3) are also modelled in the satellite structural diagrams (Fig. 4 and Fig. 5) in the domain of Systems Engineering.

If we instantiate the model presented in the Class Diagram (Fig. 3) for the case of Amazonia-1, some examples of Objects are those shown in Fig. 6. Fig. 6 shows an Object Diagram from UML. Remembering

that an Object Diagram shows instances of Classes. In practice, if we use, for example, a relational database to implement the IS, these Objects are data on tables and not database structure.

Comparing the BDD (Fig. 4) with the instances of the Class Diagram represented in the Object Diagram (Fig. 6), it can be noticed that the structural model of Fig. 4 is in part an instance of the Class Model (Fig. 3). The subsystem Blocks in the BDD are equivalent to instances of the *Subsystem* Class. In the case of the Blocks that appear in the decomposition of the subsystem Blocks, they do not correspond to instances of the *SatellitePart* Class, since they do not represent a specific satellite part. Instead, these Blocks are equivalent to instances of the *EquipmentType* Class. Instances of the *SatellitePart* Class do not appear in the BDD but appear in the IBD as Parts of the subsystems Blocks.

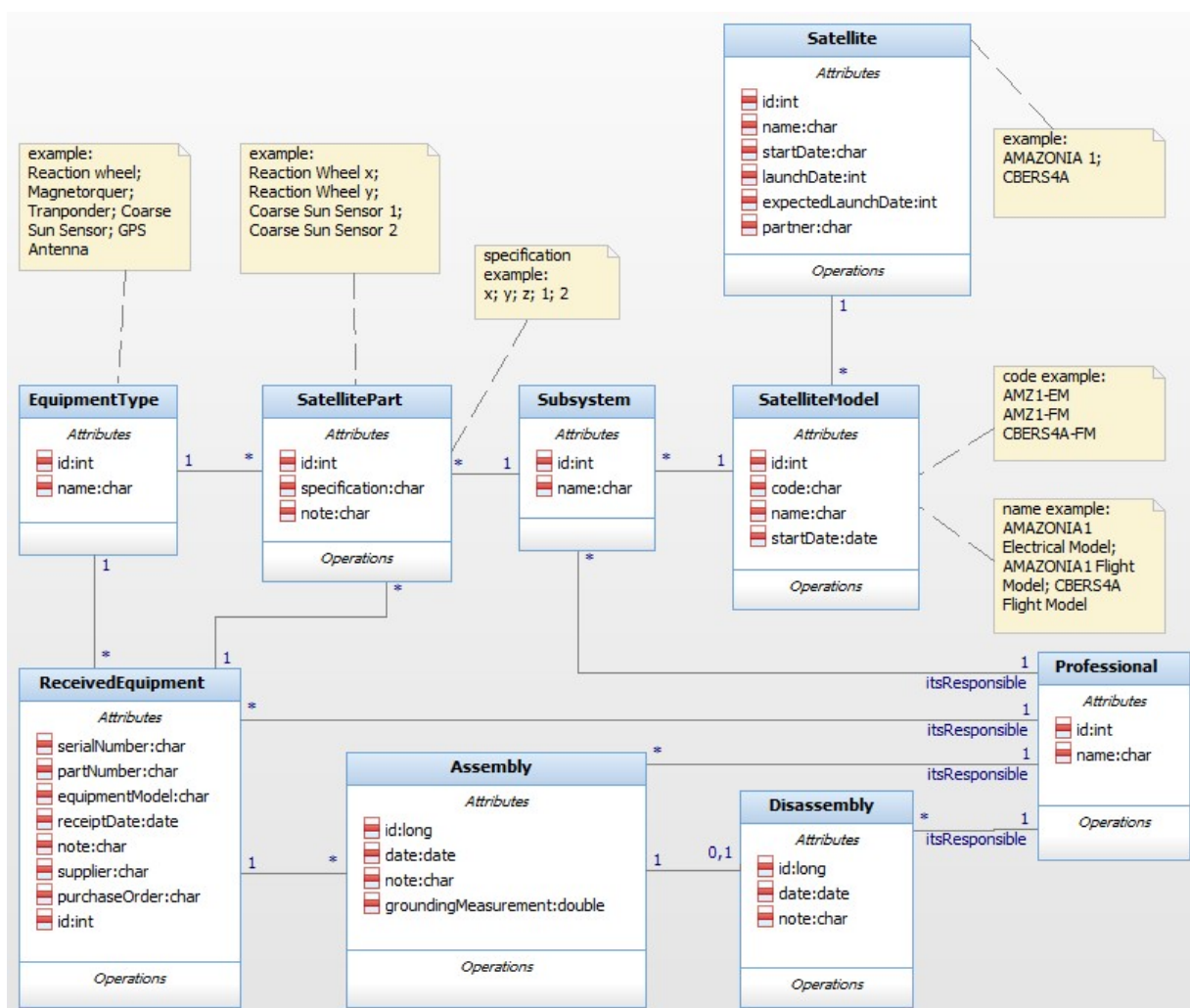


Fig. 3. Simplified UML Class Diagram representing part of the software structure for an IS for the LIT's AIT business process

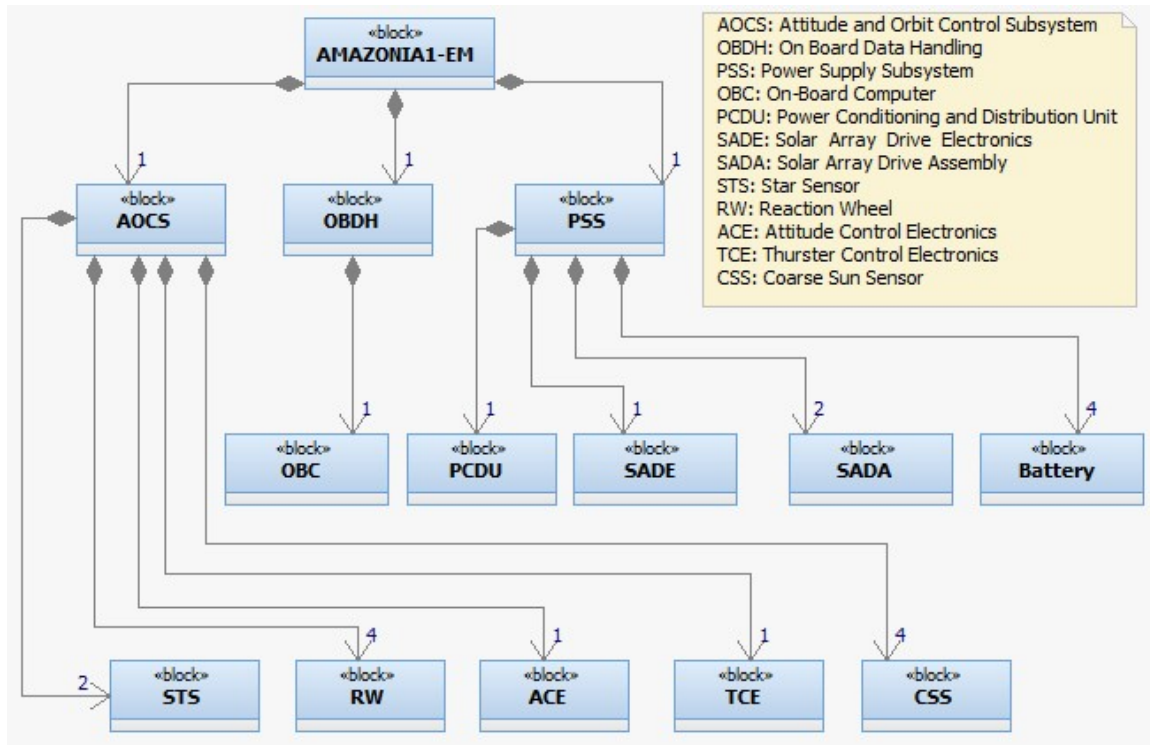


Fig. 4. Simplified SysML BDD for part of the electrical model of Amazonia-1 satellite

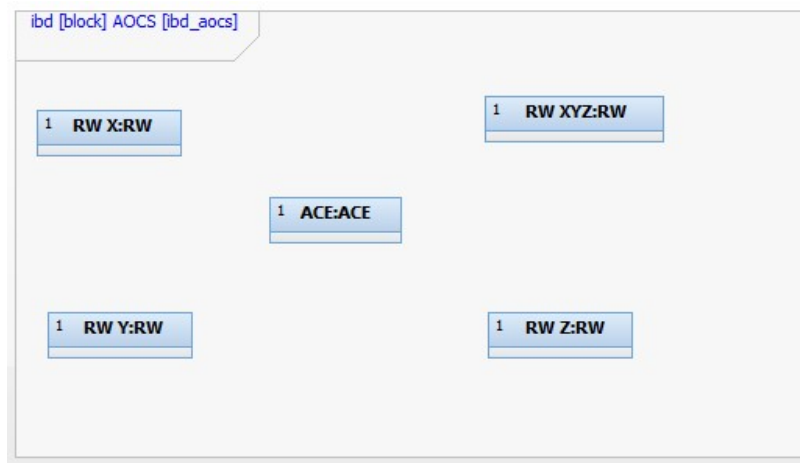


Fig. 5. Simplified SysML IBD for part of Amazonia-1 satellite

In the IBD, the types of the Parts are equivalent to instances of the *EquipmentType* Class of the Class Diagram (Fig. 3), and it is possible to relate each Part of the IBD with instances of the *SatellitePart* Class. Similarly, in an IBD of the *Satellite* Block, the subsystems would appear as Parts and would correspond to the instances of the *Subsystem* Class.

In these examples, each element in BDD corresponds to one or more instances (depending on the multiplicity) of the *SatelliteModel*, *Subsystem* or *EquipmentType* Class.

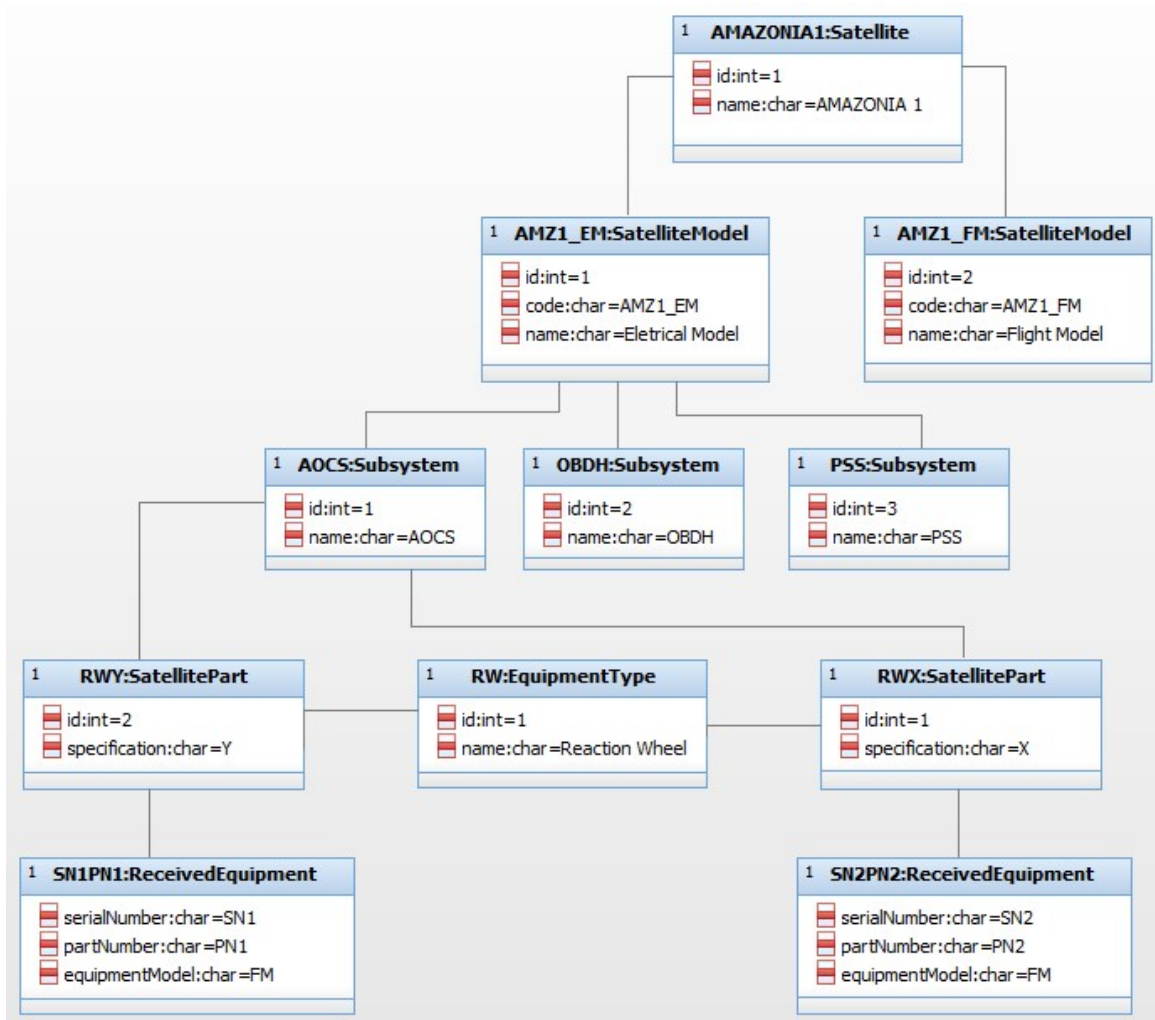


Fig. 6. Simplified UML Object Diagram representing instances of the Class Diagram showed in Fig. 3

Fig. 7 illustrates an approach to model these examples in the Systems Engineering domain in a way that the model could be reused in the field of Software Engineering for the construction of the IS. Using a super type for the Blocks would allow you to identify where your instances should be allocated to the IS domain: *EquipmentType*, *Subsystem*, or *SatelliteModel* Class. For a modelling transition, the super type would define the Classes in the IS domain, and the subtypes the Objects. As not all Systems Engineering domain super types will necessarily influence the IS, a higher-level type (or Block), like *ISLink*, could be created to identify super types of interest to the Software Engineering domain.

5. Discussion

The examples explored in this paper are only a small part of all the work that needs to be done to work with a model-based approach and to have an IS that supports the AIT process. The work is vastly larger.

We could see that some models of information that compose the structure of the IS are determined by models that would be constructed to engineering the satellites, considering a model-based approach. Although one has different focus when modelling to engineering the product and when modelling to engineering an IS, there is a relationship between these two activities.

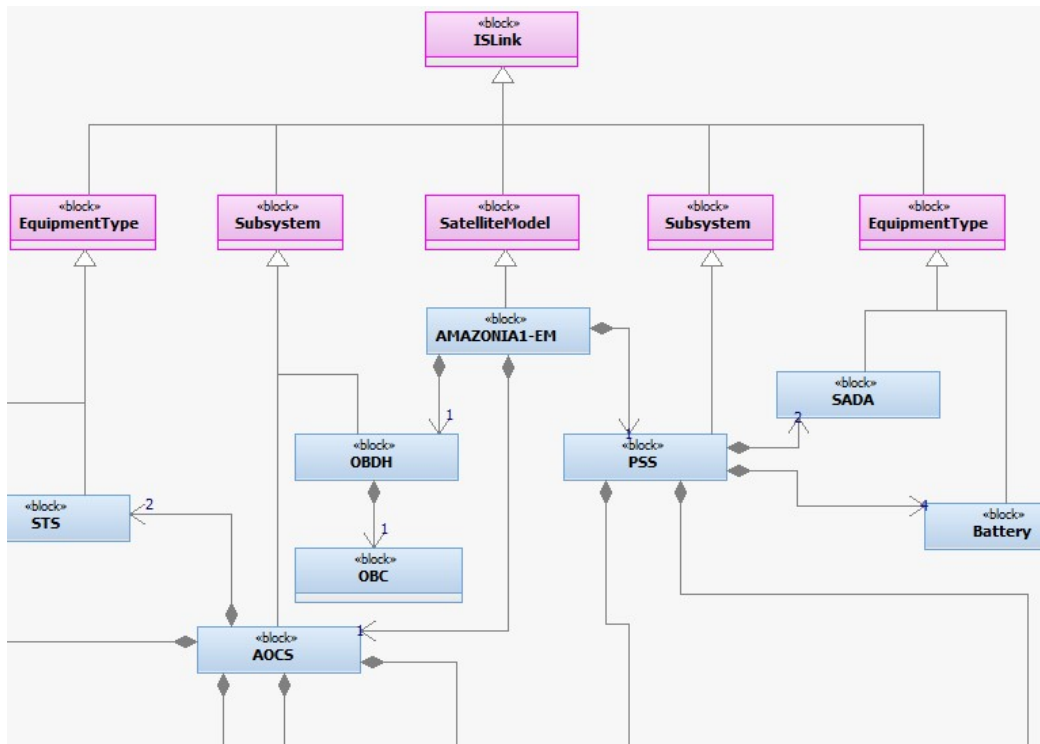


Fig. 7. An approach to model in the Systems Engineering domain in a way that the model could be reused in the Software Engineering domain

We must consider that both to build models that faithfully represent a space product, and to build Information Systems that faithfully meet the needs of the business processes to perform each space product life cycle phase, demands effort and skilled workforce. Looking at the examples, there is a sense that much of the information required for the IS is determined by the Systems Engineering domain models and that, therefore, collaboration between these two activities can be useful in reducing the development effort.

Finding a link between the models built in the field of Systems Engineering and those built in the field of Software Engineering can also bring other benefits. For example, having a structural model for satellite engineering, and knowing that part of this model is a source of information for the IS, the ideal would be that the IS was directly affected by changes in this model. This would contribute to the consistency of the information and to eliminate rework when feeding the IS with information.

It seems interesting to explore this relationship and the possible collaboration between these activities.

6. Further work

A small part of the Amazonia-1 satellite structural model was used as an example for the scope of this article. Many more models, other structural models and

models of other aspects of the system, still need to be explored in future work.

Aiming to arrive at a collaborative approach between modelling of systems for engineering of space products and modelling IS to support the business processes that perform the phases of the life cycle of the space products, as next steps we intend to explore:

- the unfolding of the business process model that performs AIT in LIT;
- the unfolding of the information that flows in this business process;
- the unfolding and complementation of the structural models presented in this work;
- other MBSE models and their relationship to the information flowing in the AIT process.

7. Conclusions

With a focus on Space Systems Engineering, exemplified by the satellite AIT process performed at LIT/INPE, this work presents and compares a model (Class Diagram), usually built in the domain of Software Engineering, to model the structure of software systems that use the object orientation paradigm, and two models (BDD and IBD), usually built in the field of Systems Engineering considering a model-based approach, to model the structure of systems. From the analysis of the relationship between

these models it was possible to observe that the collaboration between the modelling activities performed in the field of Software Engineering and those performed in the field of Systems Engineering can be helpful to reduce development effort.

This is an initial study; further work is needed to arrive at a collaborative approach to model space products to engineering them, and to model IS to support the business process that perform the phases of the life cycle of the space products.

Acknowledgements

The authors would like to thank the LIT's AIT team that provided expertise that greatly assisted the research; and also Fundep (Research Development Foundation) for sponsoring Dr. Loureiro's participation at IAC 2018.

References

- [1] P. M. Fischer, D. Ludtke, C. Lange, F. C. Roshani, F. Dannemann, A. Gerndt, Implementing model-based system engineering for the whole lifecycle of a spacecraft, CEAS Space J (2017) 351-365.
- [2] S. Bonnet, J. Voirin, D. Exertier and V. Normand, Not (strictly) relying on SysML for MBSE: Language, tooling and development perspectives: The Arcadia/Capella rationale, 2016 Annual IEEE Systems Conference (SysCon), Orlando, FL, 2016, 18-21 April, pp.1-6.
- [3] D. Kaslow, B. Ayres, P. T. Cahill, L. Hart and R. Yntema, A Model-Based Systems Engineering (MBSE) approach for defining the behaviors of CubeSats, 2017 IEEE Aerospace Conference, Big Sky, MT, 2017, 4-11 March, pp.1-14.
- [4] W. A. dos Santos, B. Leonor, S. Stephany, A Knowledge-Based and Model-Driven Requirements Engineering Approach to Conceptual Satellite Design, ER '09 Proceedings of the 28th International Conference on Conceptual Modeling, Gramado, RS, 2009, 9-12 November, pp. 487-500.
- [5] INCOSE, Systems Engineering Handbook A Guide for System Life Cycle Processes and Activities, fourth ed., John Wiley & Sons, New Jersey, 2015.
- [6] S. Friedenthal, A. Moore, and R. Steiner, A Practical Guide to SysML: The Systems Modeling Language, 3rd ed. Morgan Kaufmann, Waltham, 2015.
- [7] M. Rosing, A. W. Scheer, H. Scheel, A Practical Guide to SysML: The Complete Business Process Handbook Body of Knowledge from Process Modeling to BPM. Morgan Kaufmann, Waltham, 2015.
- [8] R. Stair, G. Reynolds, Fundamentals of Information Systems, Ninth Edition. Cengage Learning, Boston, 2018.
- [9] G. T. A. Guedes, UML 2 Uma abordagem prática, 3rd ed. Novatec Editora Ltda., São Paulo, 2018.
- [10] OMG, OMG Unified Modeling Language Version 2.5.1, OMG Document Number formal/2017-12-05, 2017.
- [11] A. C. Melo, Desenvolvendo Aplicações com UML 2.2, Third Edition. Brasport, Rio de Janeiro, 2011.
- [12] IBM Corporation, Essentials of IBM Rational Rhapsody v761 for Systems Engineers Student Manual vol 1, 2012.
- [13] ECSS, Space project management - Project planning and implementation, ECSS-M-ST-10C rev, 1, 2009.
- [14] INPE/LIT, Minutes of the meeting held on September 21, 2017 at LIT/INPE, LIT25-LIT05-MR-001, 2017.
- [15] INPE, Missão Amazonia 1, <http://www3.inpe.br/amazonia-1/>, (accessed 08.09.18).

PUBLICAÇÕES TÉCNICO-CIENTÍFICAS EDITADAS PELO INPE

Teses e Dissertações (TDI)

Teses e Dissertações apresentadas nos Cursos de Pós-Graduação do INPE.

Manuais Técnicos (MAN)

São publicações de caráter técnico que incluem normas, procedimentos, instruções e orientações.

Notas Técnico-Científicas (NTC)

Incluem resultados preliminares de pesquisa, descrição de equipamentos, descrição e ou documentação de programas de computador, descrição de sistemas e experimentos, apresentação de testes, dados, atlas, e documentação de projetos de engenharia.

Relatórios de Pesquisa (RPQ)

Reportam resultados ou progressos de pesquisas tanto de natureza técnica quanto científica, cujo nível seja compatível com o de uma publicação em periódico nacional ou internacional.

Propostas e Relatórios de Projetos (PRP)

São propostas de projetos técnico-científicos e relatórios de acompanhamento de projetos, atividades e convênios.

Publicações Didáticas (PUD)

Incluem apostilas, notas de aula e manuais didáticos.

Publicações Seriadas

São os seriados técnico-científicos: boletins, periódicos, anuários e anais de eventos (simpósios e congressos). Contam destas publicações o Internacional Standard Serial Number (ISSN), que é um código único e definitivo para identificação de títulos de seriados.

Programas de Computador (PDC)

São a seqüência de instruções ou códigos, expressos em uma linguagem de programação compilada ou interpretada, a ser executada por um computador para alcançar um determinado objetivo. Aceitam-se tanto programas fonte quanto os executáveis.

Pré-publicações (PRE)

Todos os artigos publicados em periódicos, anais e como capítulos de livros.