



Ministério da
**Ciência, Tecnologia
e Inovação**



sid.inpe.br/mtc-m21b/2014/02.14.14.56-TDI

UMA ABORDAGEM ORIENTADA A MODELOS APLICADA À OTIMIZAÇÃO DE LAYOUT DE EQUIPAMENTOS EMBARCADOS EM SATÉLITES

Michele Medeiros

Dissertação de Mestrado do
Curso de Pós-Graduação em
Engenharia e Tecnologia Espaciais/
Gerenciamento de Sistemas
Espaciais, orientada pelos Drs.
Walter Abrahão dos Santos, e
Fabiano Luis de Sousa, aprovada
em 26 de fevereiro de 2014.

URL do documento original:

<<http://urlib.net/8JMKD3MGP5W34M/3FP3T65>>

INPE
São José dos Campos
2014

PUBLICADO POR:

Instituto Nacional de Pesquisas Espaciais - INPE

Gabinete do Diretor (GB)

Serviço de Informação e Documentação (SID)

Caixa Postal 515 - CEP 12.245-970

São José dos Campos - SP - Brasil

Tel.:(012) 3208-6923/6921

Fax: (012) 3208-6919

E-mail: pubtc@sid.inpe.br

CONSELHO DE EDITORAÇÃO E PRESERVAÇÃO DA PRODUÇÃO INTELLECTUAL DO INPE (RE/DIR-204):**Presidente:**

Marciana Leite Ribeiro - Serviço de Informação e Documentação (SID)

Membros:

Dr. Antonio Fernando Bertachini de Almeida Prado - Coordenação Engenharia e Tecnologia Espacial (ETE)

Dr^a Inez Staciarini Batista - Coordenação Ciências Espaciais e Atmosféricas (CEA)

Dr. Gerald Jean Francis Banon - Coordenação Observação da Terra (OBT)

Dr. Germano de Souza Kienbaum - Centro de Tecnologias Especiais (CTE)

Dr. Manoel Alonso Gan - Centro de Previsão de Tempo e Estudos Climáticos (CPT)

Dr^a Maria do Carmo de Andrade Nono - Conselho de Pós-Graduação

Dr. Plínio Carlos Alvalá - Centro de Ciência do Sistema Terrestre (CST)

BIBLIOTECA DIGITAL:

Dr. Gerald Jean Francis Banon - Coordenação de Observação da Terra (OBT)

REVISÃO E NORMALIZAÇÃO DOCUMENTÁRIA:

Marciana Leite Ribeiro - Serviço de Informação e Documentação (SID)

Yolanda Ribeiro da Silva Souza - Serviço de Informação e Documentação (SID)

EDITORAÇÃO ELETRÔNICA:

Maria Tereza Smith de Brito - Serviço de Informação e Documentação (SID)

André Luis Dias Fernandes - Serviço de Informação e Documentação (SID)



Ministério da
**Ciência, Tecnologia
e Inovação**



sid.inpe.br/mtc-m21b/2014/02.14.14.56-TDI

UMA ABORDAGEM ORIENTADA A MODELOS APLICADA À OTIMIZAÇÃO DE LAYOUT DE EQUIPAMENTOS EMBARCADOS EM SATÉLITES

Michele Medeiros

Dissertação de Mestrado do
Curso de Pós-Graduação em
Engenharia e Tecnologia Espaciais/
Gerenciamento de Sistemas
Espaciais, orientada pelos Drs.
Walter Abrahão dos Santos, e
Fabiano Luis de Sousa, aprovada
em 26 de fevereiro de 2014.

URL do documento original:

<<http://urlib.net/8JMKD3MGP5W34M/3FP3T65>>

INPE
São José dos Campos
2014

Dados Internacionais de Catalogação na Publicação (CIP)

Medeiros, Michele.

M467u Uma abordagem orientada a modelos aplicada à otimização de layout de equipamentos embarcados em satélites / Michele Medeiros. – São José dos Campos : INPE, 2014.

xxv + 141 p. ; (sid.inpe.br/mtc-m21b/2014/02.14.14.56-TDI)

Dissertação (Mestrado em Engenharia e Tecnologia Espaciais/Gerenciamento de Sistemas Espaciais) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2014.

Orientadores : Drs. Walter Abrahão dos Santos, e Fabiano Luis de Sousa.

1. MBSE. 2. MDO. 3. layout. 4. satélite. 5. infraestrutura de software. I.Título.

CDU 629.7:004.414.23



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/).

Aprovado (a) pela Banca Examinadora
em cumprimento ao requisito exigido para
obtenção do Título de **Mestre** em

**Engenharia e Tecnologia
Espaciais/Gerenciamento de Sistemas
Espaciais**

Dra. Ana Maria Ambrosio



Presidente / INPE / São José dos Campos - SP

Dr. Fabiano Luis de Sousa



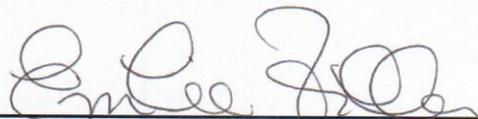
Orientador(a) / INPE / SJCampos - SP

Dr. Walter Abrahão dos Santos



Orientador(a) / INPE / São José dos Campos - SP

Dra. Emília Villani



Convidado(a) / ITA / São José dos Campos - SP

Este trabalho foi aprovado por:

maioria simples

unanimidade

Aluno (a): **Michele Medeiros**

São José dos Campos, 26 de Fevereiro de 2014

“Can I Play with Madness”.

IRON MAIDEN
em *“Seventh Son of a Seventh Son album”*, 1988

*A meus pais Edson e Shimako, à minha irmã Milene e à
minha princesa de quatro patas Lady*

AGRADECIMENTOS

Ao Instituto Nacional de Pesquisas Espaciais - INPE, pela oportunidade de estudar, inicialmente como aluna de matérias isoladas pelas quais pude conhecer as linhas de pesquisa do INPE. Agradeço em especial ao professor Dr. Nilson Sant' Ana que me incentivou a fazer o mestrado.

Aos meus orientadores, pelo conhecimento disponibilizado, além de todo o incentivo e compreensão. Agradeço especialmente ao prof. Dr. Walter Abrahão dos Santos por todo o seu apoio durante as mudanças de escopo que ocorreram neste trabalho e pela ajuda durante este difícil percurso. Ao prof. Dr. Fabiano Luis de Sousa, pela paciência em ensinar conceitos totalmente novos para mim e pela oportunidade de participar nas reuniões e discussões do grupo do PSOTLayout.

À gerência do LIT pelo apoio administrativo, em especial aos meus colegas dos departamentos: DSI (Desenvolvimento de Sistemas de Informação) e LSIS (Laboratório de Engenharia Simultânea de Sistemas) pelas discussões técnicas, material didático, licenças de software e principalmente pelo grande incentivo técnico e emocional. Aqui não posso deixar de citar a Ana Claudia de Paula Silva e o Luiz Alexandre da Silva que abriram barreiras e tornaram possível que eu pudesse terminar este trabalho de mestrado.

Agradeço à biblioteca do INPE, por fornecer meios para a pesquisa e elaboração desta dissertação, através da digital e dos estilos fornecidos para diversos editores. Também agradeço pelas informações tão gentilmente fornecidas pelos seus funcionários, em especial à Yolanda Ribeiro da Silva Souza, por toda ajuda e incentivo.

À secretária da ETE, Edleusa Aparecida Ferreira, pela extrema competência, sempre bem educada e pronta para ajudar.

Agradeço a colaboração, entre LIT(LSIS) - ETE - CAP, que proporcionou as informações e o ambiente necessários para a elaboração deste trabalho.

Finalmente, a todos aqueles que de alguma forma contribuíram para esta dissertação.

RESUMO

Atualmente, nota-se um sensível aumento em número e complexidade a bordo de satélites exigindo melhoria de métodos e processos em Engenharia de Sistemas Espaciais. Este trabalho tem como foco a interoperabilidade de ferramentas em ambientes computacionais de engenharia, neste trabalho, denominada Optima que apoiem o projeto de satélites. Essa necessidade é mais premente na fase de projeto conceitual, pois envolvem decisões que podem influenciar todo o ciclo de vida de um sistema. Aplicações computacionais permitem estudo de cenários que podem envolver simulações, escolha de arquiteturas viáveis de sistemas e complexos cálculos de otimização em diversas disciplinas da engenharia de sistemas espaciais. Geralmente, isso requer lidar com a interoperabilidade na troca de dados entre as diversas ferramentas de software tradicionais de cada disciplina podendo limitar ações de engenharia concorrente. Visando contribuir neste cenário, este trabalho lida com duas abordagens conjuntas, a saber: (1) Engenharia de Sistemas Baseada em Modelos (MBSE) utilizando a linguagem SysML tratada por um aplicativo denominado SatBudgets estendido e (2) Otimização de Projeto Multidisciplinar (MDO) empregando um algoritmo de otimização, evolutivo, denominado MGEO. O problema de layout de equipamentos embarcados em satélites é apresentado como estudo de caso limitado ao cenário bidimensional para fins de escopo. Este problema ocorre tipicamente durante a fase conceitual do desenvolvimento. A partir de requisitos de alto nível da arquitetura de missão da Plataforma Multimissão (PMM) do INPE, é feita a decomposição e derivação de requisitos que influenciarão o problema de layout. Neste processo são definidas três funções-objetivo a serem otimizadas pelo algoritmo MGEO. Adicionalmente, este processo envolveu a interoperação e reuso de artefatos entre duas ferramentas comerciais: uma de Gestão de Requisitos e outra para Modelagem SysML. A metodologia adotada neste trabalho poderá ser estendida para uma grande variedade de problemas com domínio similar de aplicação.

A MODEL-BASED APPROACH APPLIED TO THE LAYOUT OPTIMIZATION PROBLEM OF EQUIPMENT ON-BOARD SATELLITES

ABSTRACT

Nowadays there has been a significant increase in number and complexity on board satellites which requires improvement in methods and processes in Space System Engineering. This dissertation focuses on one such aspect, more specifically regarding the interoperability of tools for computational engineering environments, here named Optima, which support the design of satellites and have to meet constraints on time, cost and quality, among others. This need is more remarkable in the conceptual design phase since it involves decisions that may influence the entire system life cycle. Computing applications allow the study of scenarios involving simulations, choice of viable system architectures and complex optimization calculations on the various disciplines of space systems engineering. Typically, this requires dealing with interoperability in data exchange between the various traditional software tools of each particular discipline which may hinder actions on concurrent engineering. In order to contribute to this scenario, this work deals jointly with two approaches, namely: (1) Model-Based Systems Engineering (MBSE) using the SysML language handled by an extended application called SatBudgets and (2) Multidisciplinary Design Optimization (MDO) using a evolutionary algorithm called MGEO already available. The layout problem of on-board equipments in satellites is presented as a case study limited to a two-dimensional scenario for scope reasons. This problem typically occurs during the conceptual phase of development. Starting from high-level requirements of the mission architecture for the Brazilian Space Multimission Platform (MMP), the decomposition and derivation of requirements that will influence the layout problem were done. From this process, three objective functions were defined for the MGEO optimization. Additionally, this process involved the interoperation and reuse of artifacts between two commercial tools: one for Requirements Management and another for SysML modeling. The methodology adopted here can be extended to a wide range of problems with similar application domain.

LISTA DE FIGURAS

	<u>Pág.</u>
1.1 Contexto geral do trabalho	4
1.2 Mapa dos conceitos e tecnologias usadas neste trabalho e aplicadas ao desenvolvimento de satélites.	7
2.1 Mapa mental usado para a estruturação do estudo teórico.	9
2.2 Fluxo de trabalho para exploração de conceitos de missões espaciais.	11
2.3 <i>Framework</i> usada para Otimização de Projeto de Sistema Multidisciplinar (<i>Multidisciplinary System Design Optimization - MSDO</i>).	14
2.4 Sysml.	18
2.5 Exemplo de XML.	21
2.6 Manipulação de XML.	21
2.7 Níveis de Abstração.	23
2.8 SOA como um estilo de arquitetura.	24
2.9 Serviços Abertos para Comunicações do Ciclo de Vida.	27
2.10 Modelo de Arquitetura	29
2.11 Arquitetura do Modelo de Dados Conceitual Global da ECSS para um projeto.	31
2.12 Fluxo de trabalho da ferramenta SatBudgets.	33
2.13 Integração e execução entre SatBudgets e SpaceESB.	34
3.1 Visão geral com estrutura dos tópicos deste capítulo.	37
3.2 Elementos da Missão	39
3.3 Decomposição do Sistema	41
3.4 Objetivos e Requisitos de uma missão de satélite.	43
3.5 Origem de Requisitos para o subsistema de Estrutura.	44
3.6 <i>Layout</i> dos equipamentos da PMM.	46
3.7 Alguns fatores e elementos que influenciam do projeto de estruturas e do <i>layout</i> de equipamentos em satélites	47
3.8 Influência do Sistema de Lançamento no <i>layout</i>	48
3.9 Influência da escolha de órbita no <i>layout</i>	49
3.10 Influência da radiação espacial no <i>layout</i> de equipamentos	50
3.11 As soluções de compromisso (<i>trade-offs</i>) durante a caracterização da missão	51
3.12 Exemplo de acoplamento do <i>layout</i> do subsistema de estruturas com os outros subsistemas.	52
4.1 Fluxo de Trabalho do SatBudgets.	58

4.2	Fluxo de Trabalho do PJLayout.	59
4.3	Exemplo das distâncias para um equipamento posicionado sobre o painel.	60
4.4	Exemplo de rotação do equipamento sobre o painel.	61
4.5	Processo simplificado usando as interfaces das ferramentas de software.	62
4.6	Fluxo de Trabalho do Optima.	63
5.1	Visão geral com estrutura dos tópicos deste capítulo.	68
5.2	Diagrama de Contexto da PMM	69
5.3	Diagrama estrutural de blocos da PMM	70
5.4	Diagrama de Hierarquia de Subsistemas da PMM	71
5.5	Diagrama Estrutural de Blocos de Restrição com inclusão do bloco de otimização MGEO	72
5.6	Tela da estrutura inicial do DOORS com módulo “Requisitos de Sistema”	73
5.7	Requisitos da PMM inseridos no DOORS	73
5.8	Conexão de Modelos a Requisitos usando o Rhapsody Gateway	75
5.9	Diagrama de Blocos com Decomposição de Requisitos da Missão da PMM	76
5.10	Acoplamento de Lançamento - <i>Layout</i>	77
5.11	Acoplamento de Arquitetura e Configuração Elétrica - <i>Layout</i>	78
5.12	Acoplamento com a Arquitetura Mecânica - <i>Layout</i> : área de posicionamento	79
5.13	Acoplamento com Arquitetura Mecânica - <i>Layout</i> : centro de massa com rastreadabilidade.	80
5.14	Acoplamento com Órbita - <i>Layout</i> : posição relativa ao sol.	81
5.15	Acoplamento com Sistema de Telecomunicações - <i>Layout</i> : posiciona- mento de antenas.	82
5.16	Acoplamento com Subsistema de Controle de Atitude e Determinação de Órbita - <i>Layout</i> : eixos principais de inércia.	83
5.17	Acoplamento com Subsistema de Propulsão - <i>Layout</i> : posição do tanque de combustível.	84
5.18	Acoplamento com Subsistema de Potência - <i>Layout:hotspot</i>	85
5.19	Acoplamento com Estrutura - <i>Layout</i> : distribuição de área disponível.	86
5.20	Acoplamento com Controle Térmico - <i>Layout: hotspot</i>	87
5.21	Acoplamento com Carga Útil - <i>Layout</i> : EMI.	88
5.22	Acoplamento com Compatibilidade Eletromagnética - <i>Layout</i> : EMI/EMC.	89
5.23	Dados dos equipamentos da PMM.	90
5.24	Dados dos equipamentos da PMM (continuação).	91
5.25	Breve Diagrama de Blocos com os equipamentos da PMM.	93
5.26	Diagrama de Blocos com os oito equipamentos da PMM usados na oti- mização para escopo reduzido de layout bidimensional.	94

5.27	Fronteira de Pareto representada em um diagrama de bolhas de cada resultado/arquitetura encontrado pelo algoritmo MGEO exibido na interface gerada.	96
5.28	Tabela com os Resultados encontrados pelo algoritmo MGEO exibido na interface gerada. (Parte 1 de 3). (continua)	98
5.29	Tabela com os Resultados encontrados pelo algoritmo MGEO exibido na interface gerada. (Parte 2 de 3). (continua)	99
5.30	Tabela com os Resultados encontrados pelo algoritmo MGEO exibido na interface gerada. (Parte 3 de 3). - Conclusão	100
5.31	Vistas de uma das soluções (Arquitetura id=1) de <i>layout</i> encontradas pelo algoritmo	101
5.32	Soluções otimizadas de <i>layout</i> encontradas pelo algoritmo MGEO	102
B.1	Decomposição da missão da PMM.	120
B.2	Decomposição dos Requisitos Gerais da PMM.	121
B.3	Decomposição dos Requisitos de Arquitetura e Configuração Elétrica da PMM.	122
B.4	Decomposição dos Requisitos de Arquitetura Mecânica da PMM.	123
B.5	Decomposição dos Requisitos do Controle de Órbita e Atitude.	124
B.6	Decomposição dos Requisitos dos Modos de Operação.	125
B.7	Decomposição dos Requisitos das Funções da PMM.	126
B.8	Decomposição dos Requisitos dos Serviços de Telecomunicações da PMM.	127
B.9	Decomposição dos Requisitos de Alvo de Apontamento da PMM.	128
B.10	Decomposição dos Requisitos de Manipulação e Armazenamento dos Dados Embarcados da PMM.	129
B.11	Decomposição dos Requisitos de Propulsão da PMM.	130
B.12	Decomposição dos Requisitos de Suprimento de Potência da PMM.	131
B.13	Decomposição dos Requisitos de Estrutura da PMM.	132
B.14	Decomposição dos Requisitos de Controle Térmico da PMM.	133
B.15	Decomposição dos Requisitos de Interface com a Carga Útil da PMM.	134
B.16	Decomposição dos Requisitos Operacionais da PMM.	135
B.17	Decomposição dos Requisitos Específicos da PMM.	136

LISTA DE TABELAS

	<u>Pág.</u>
3.1 Associações e restrições de fatores.	42
5.1 Parâmetros das variáveis de projeto.	95
5.2 Parâmetros das variáveis do algoritmo MGEO.	95

LISTA DE ABREVIATURAS E SIGLAS

ALM	–	<i>Application Lifecycle Management</i>
API	–	<i>Application Programming Interface</i>
CAD	–	<i>Computer-Aided Design</i>
CASE	–	<i>Computer Aided Systems Engineering</i>
CDF	–	<i>Concurrent Design Facility</i>
DLR	–	<i>German Aerospace Center</i>
DOM	–	<i>Document Object Model</i>
DOORS	–	<i>Dynamic Object Oriented Requirements System</i>
ECSS	–	<i>European Cooperation for Space Standardization</i>
ESA	–	<i>European Space Agency</i>
GSFC	–	<i>The Goddard Space Flight Center</i>
IBM	–	<i>International Business Machines</i>
ID	–	<i>Identifier</i>
IEEE	–	<i>Institute of Electrical and Electronics Engineers</i>
IDM	–	<i>Integrated Design Model</i>
INCOSE	–	<i>International Council on Systems Engineering</i>
INPE	–	<i>Instituto Nacional de Pesquisas Espaciais</i>
ISO	–	<i>International Standards Organization</i>
JDOM	–	<i>Java Document Object Model</i>
MBSE	–	<i>Model Based Systems Engineering</i>
MDE	–	<i>Model-Driven Engineering</i>
MDA	–	<i>Model-Driven Architecture</i>
MDO	–	<i>Multidisciplinary Design Optimization</i>
MSDO	–	<i>Multidisciplinary System Design Optimization</i>
NASA	–	<i>National Aeronautics and Space Administration</i>
NP	–	<i>Nondeterministic Polynomial time</i>
OCS D	–	<i>Open Concurrent Design Server</i>
OMG	–	<i>Object Management Group</i>
OSLC	–	<i>Open Services for Lifecycle Communications</i>
PDF	–	<i>Portable Document Format</i>
PMM	–	<i>Plataforma Multi-Missão</i>
RFP	–	<i>Request For Proposal</i>
SAX	–	<i>Simple API for XML</i>
SDK	–	<i>Software Development Kit</i>
SMLDP	–	<i>Satellite Module Layout Design Problem</i>
SOA	–	<i>Service Oriented Architecture</i>
SOAP	–	<i>Simple Object Access Protocol</i>
SOI	–	<i>Service Oriented Integration</i>
SysML	–	<i>Systems Modeling Language</i>
STEP	–	<i>Standard for the Exchange of Product model data</i>

UUID – *Universally Unique Identifier*
UML – *Unified Modeling Language*
TI – *Tecnologia da informação*
WSH – *Windows Scripting Host*
XML – *Extensible Markup Language*
XMI – *XML Metadata Interchange*

SUMÁRIO

	<u>Pág.</u>
1 INTRODUÇÃO	1
1.1 Motivação	2
1.2 Descrição do Problema	3
1.3 Objetivos	4
1.4 Metodologia de Pesquisa	6
1.5 Organização do texto	8
2 FUNDAMENTAÇÃO TEÓRICA	9
2.1 Engenharia de Sistemas Espaciais	9
2.2 Otimização de Projeto Multidisciplinar (<i>MDO - Multidisciplinary Design Optimization</i>)	12
2.3 Engenharia Dirigida a Modelos (<i>MDE - Model Driven Engineering</i>) e a Linguagem de Modelagem SysML)	15
2.4 Ambientes Colaborativos para Desenvolvimento de Sistemas	18
2.4.1 Interoperabilidade entre Sistemas de Software	19
2.4.2 O Padrão STEP ISO 10303	19
2.4.3 Os Formatos XML e XMI para Troca de Dados	20
2.4.4 SOA e SOI	22
2.4.5 Serviços via OSLC	25
2.4.6 Visão de Interoperabilidade pela ESA	28
2.4.7 Alguns Trabalhos Relacionados	32
3 METODOLOGIA E DESENVOLVIMENTO	37
3.1 Visão Geral	37
3.2 Processo de desenvolvimento de projetos espaciais	38
3.2.1 Fase Conceitual	38
3.2.2 Missões Espaciais	38
3.2.3 Desenvolvimento de Satélites	42
3.2.4 Subsistema de Estruturas - Configuração e Acondicionamento	43
3.2.5 O problema do <i>Layout</i> de Equipamentos Embarcados em Satélites	45
3.2.6 Alguns Fatores que Influenciam o <i>Layout</i>	47
3.3 O uso da linguagem SysML	53
3.4 O uso proposto da MDO	53

4	IMPLEMENTAÇÃO DA SOLUÇÃO MBSE - MDO	55
4.1	Seleção de Ferramentas Comerciais para Engenharia de Sistemas	55
4.1.1	A ferramenta IBM® Rational® DOORS®	56
4.1.2	A ferramenta IBM® Rational® Rhapsody®	56
4.1.3	A ferramenta IBM® Rational® Rhapsody® Gateway®	56
4.2	As Ferramentas Desenvolvidas no INPE	57
4.2.1	O Aplicativo Satbudgets	57
4.2.2	O Aplicativo PJLayout e o Algoritmo MGEO	58
4.3	Infraestrutura Prototipada de Software e Tecnologias Usadas	61
5	MODELAGEM DO ESTUDO DE CASO E RESULTADOS	67
5.1	Diagrama de Contexto	68
5.2	Diagramas estruturais de blocos	70
5.3	Interoperabilidade entre Modelos e Requisitos	72
5.4	Decomposição de Requisitos da PMM	75
5.5	Acoplamento e Rastreabilidade de Requisitos PMM ao Problema de <i>Layout</i>	76
5.5.1	Acoplamento com o Segmento de Lançamento.	76
5.5.2	Acoplamento com a Arquitetura e Configuração Elétrica.	77
5.5.3	Acoplamento com a Arquitetura Mecânica.	78
5.5.4	Acoplamento com Requisitos de Órbita.	80
5.5.5	Acoplamento com o Subsistema de Telecomunicações.	81
5.5.6	Acoplamento com o Subsistema de Controle de Atitude e Determinação de Órbita - <i>Layout</i>	82
5.5.7	Acoplamento com o Subsistema de Propulsão.	83
5.5.8	Acoplamento com o Subsistema de Potência.	84
5.5.9	Acoplamento com o Subsistema de Estrutura.	85
5.5.10	Acoplamento com o Subsistema de Controle Térmico.	86
5.5.11	Acoplamento com a Carga Útil.	87
5.5.12	Acoplamento com o Subsistema de Cablagem.	88
5.6	Redução do Escopo da Otimização.	89
5.7	Parametrização da Execução do MGEO.	95
5.8	Resultados Obtidos na Otimização MBSE-MDO	95
5.9	Análise dos Resultados	102
6	CONSIDERAÇÕES FINAIS	103
6.1	Conclusões	103
6.2	Principais Contribuições	104
6.3	Trabalhos Futuros	104

REFERÊNCIAS BIBLIOGRÁFICAS	107
APÊNDICE A - UTILIZAÇÃO PASSO A PASSO DO FLUXO DE TRABALHO DO OPTMA	115
APÊNDICE B - DIAGRAMAS DE REQUISITOS DA PMM	119
B.1 Diagramas de Requisitos da PMM	119
ANEXO A - COMPLEX SYSTEMS DEVELOPED WITH SYS- TEM CONCURRENT ENGINEERING.	137
ANEXO B - EXPLORAÇÃO DO ESPAÇO DE PROJETO DE SA- TÉLITES PARA UM CENÁRIO REDUZIDO DE ANÁLISE.	139
ANEXO C - LABE3S: A PROTOTYPE OF AN EXPERIMENTAL CONCURRENT DESIGN FACILITY FOR SPACE SYSTEMS EN- GINEERING.. . . .	141

1 INTRODUÇÃO

O desenvolvimento de satélites sempre foi muito desafiador, e a Fase Conceitual do desenvolvimento é extremamente crítica, pois nesta fase são tomadas decisões que terão impacto durante todo o ciclo de vida do sistema. Durante a fase conceitual de desenvolvimento são consideradas diferentes arquiteturas candidatas (elétricas e mecânicas) que satisfaçam a missão espacial considerando as restrições de custo e prazo. Durante a busca por estas arquiteturas a multidisciplinaridade envolvida é bastante ativa, pois engenheiros e projetistas de várias especialidades dividem o importante papel de definir as principais funcionalidades dos subsistemas do satélite. Por exemplo, na definição das arquiteturas mecânica e elétrica, o *layout* dos equipamentos nos painéis estruturais deve satisfazer os requisitos elétricos e mecânicos além das restrições envolvidas (DE SOUSA et al., 2013). O que origina um problema de Otimização de Projeto Multidisciplinar, do inglês *Multidisciplinary design optimization (MDO)*.

O campo de pesquisa da MDO estuda a aplicação de técnicas numéricas de otimização para o desenvolvimento de projetos multidisciplinares (MARTINS; LAMBEY, 2013). Ela é beneficiada pelas novas tecnologias, principalmente pelo aumento da capacidade computacional que torna possível a execução de *loops* de otimização com grande número de iterações. A MDO, como a Engenharia de Sistemas, também é multidisciplinar e considera as interfaces dos subsistemas, envolvendo: diferentes disciplinas, diferentes modelos e ferramentas computacionais trabalhando em conjunto para o desenvolvimento de sistemas.

Um tópico relevante ao desenvolvimento de sistemas complexos, como satélites, é a criação de ambientes integrados de desenvolvimento usando Engenharia Simultânea. Dentro dessa realidade, várias instituições e empresas vêm aumentando o interesse no estudo e implantação de ambientes que permitam o desenvolvimento de sistemas complexos e multidisciplinares. Alguns destes tipos de ambientes em laboratórios de prestígio são: *CDF - Concurrent Design Facility* do (European Space Agency (2012)), *CEF - Concurrent Engineering Facility* do German Aerospace Center (DLR) (2012), *TeamX* do Jet Propulsion Laboratory, *National Aeronautics and Space Administration (NASA)* (2012).

A Engenharia Dirigida a Modelos, do inglês *Model Driven Engineering (MDE)* é uma abordagem para desenvolver sistemas complexos, confiáveis e reusáveis (SCHMIDT, 2006). A Engenharia de Sistemas Baseada em Modelos (*Model Based Systems Engineering - MBSE*) é uma especialização da MDE e usa modelagem de sistemas para

apoiar a análise, especificação, projeto e verificação de sistemas em desenvolvimento.

O uso de modelos já é considerado consolidado nas engenharias específicas, porém na Engenharia de Sistemas ainda não havia um padrão de modelagem que permitisse o do sistema completo, apesar de existirem vários tipos de modelos usados na Engenharia de Sistemas. A Linguagem de Modelagem de Sistemas, do inglês *Systems Modeling Language (SysML)* é uma linguagem padrão para modelagem de sistemas que vem ganhando força entre os engenheiros de sistemas (OBJECT MANAGEMENT GROUP, 2013b). Ela tem a pretensão de tornar-se tão usada como é a Linguagem de Modelagem Unificada, do inglês *Unified Modeling Language (UML)*, na Engenharia de Software atual (OBJECT MANAGEMENT GROUP, 2013a). Ela permite a modelagem completa do sistema e, por ser padronizada, permite que seja compreendida por diferentes especialistas e pelos interessados (*stakeholders*); além disso, modelos padronizados facilitam a simulação e a automatização.

Assim, diante das necessidades apresentadas, esta dissertação de mestrado propõe melhorias para o desenvolvimento de satélites através das abordagens citadas e da criação de um protótipo de infraestrutura de software para apoiar os projetistas de satélites, durante a fase conceitual para o caso específico de otimização de *layout* de equipamentos embarcados em satélites.

1.1 Motivação

A Fase Conceitual do Projeto de Sistemas Espaciais tem papel fundamental, pois nesta fase é feita a análise de viabilidade do sistema, além de tomadas de decisão que deverão influenciarão durante todo o ciclo de vida do sistema.

Com o objetivo de ajudar os projetistas de satélites, na fase conceitual, esta pesquisa busca as soluções mais proeminentes nas áreas de Engenharia de Sistemas, Otimização em Projeto Multidisciplinar e Interoperabilidade entre Softwares que possam ser aplicadas a esta fase proporcionando uma infraestrutura de software que dê apoio e traga melhorias ao desenvolvimento de satélites no INPE.

Com os avanços das tecnologias, os sistemas em geral estão mais complexos e o uso da Engenharia de Sistemas, para o desenvolvimento destes sistemas, vem aumentando e evoluindo para adequar-se a estes novos desafios. Além dos sistemas mais complexos, os avanços tecnológicos também proporcionam meios que alavancam o desenvolvimento destes novos sistemas, sendo ao mesmo tempo desafiadores e promissores.

A Engenharia de Sistemas deve lidar com equipes multidisciplinares, recursos separados fisicamente, grandes quantidade de dados, ferramentas de software, modelos gráficos e matemáticos diferentes, entre muitos outros fatores. Por isso, para o sucesso do emprego da Engenharia de Sistemas, não basta somente metodologias e processos. As principais mudanças a serem feitas para a implantação de um processo de engenharia de sistemas consistem em organizar as ferramentas já usadas, recursos humanos e conhecimento. Portanto, um ambiente de Engenharia Simultânea deve ter um processo, uma equipe (multidisciplinar) um modelo integrado, instalações e uma infraestrutura de software (BANDECCHI et al., 1999). Neste cenário multidisciplinar, uma infraestrutura de software colaborativa e integrada para apoiar este ambiente é desejável, e também um desafio a ser alcançado.

Modelos podem antecipar vários aspectos dos objetos e de sistemas, não somente suas características, mas também como eles interagem uns com os outros. Os modelos podem ser também usados para prover conhecimento que ajude na tomada de decisões, além do seu uso para a compreensão, projeto, construção, implantação, operação, manutenção e modificação do projeto. O uso de modelos padrão proporciona a possibilidade de automação e simulação, o que não pode ser feito com documentação textual.

Atualmente com os avanços na capacidade de processamento dos computadores modernos, os algoritmos de otimização podem processar maior número de dados em menor tempo, tornando viável o processo automático de otimização e favorecendo áreas como a Otimização de Projeto Multidisciplinar (MDO). No desenvolvimento de satélites, a integração necessária entre várias disciplinas distintas compreende um vasto campo para o uso de MDO.

Esta dissertação trata de um caso específico da área de *layout* de equipamentos embarcados usando-se MDO, na fase conceitual do desenvolvimento de satélites, das vantagens do uso de modelos na Engenharia de Sistemas, mostrando a ligação entre os requisitos de alto nível até a especificação dos parâmetros dos subsistemas.

1.2 Descrição do Problema

Este trabalho tem enfoque na fase conceitual do desenvolvimento de projetos de satélites, fase esta muito crítica e importante, pois como apresentado na figura 1.1 é nela que ocorre a caracterização da missão, que é apresentada como um fluxograma com etapas muito complexas. Uma dessas etapas é a de soluções de compromisso (*trade-offs*), no qual ocorre um intenso esforço conjunto dos engenheiros de várias

especialidades com o objetivo de encontrar arquiteturas alternativas e na análise e escolha da melhor delas. Este processo bastante custoso, é tradicionalmente baseado na experiência das pessoas envolvidas e deve levar em conta a intensa troca de informações decorrentes das Inter-relações que ocorrem durante todo o processo de projeto. Decorrente destas inter-relações citadas, o problema do *layout* é influenciado e influencia as decisões tomadas ainda nesta fase inicial, e não somente em fases mais avançadas quando as especificações e parâmetros de projeto já foram estabelecidos.

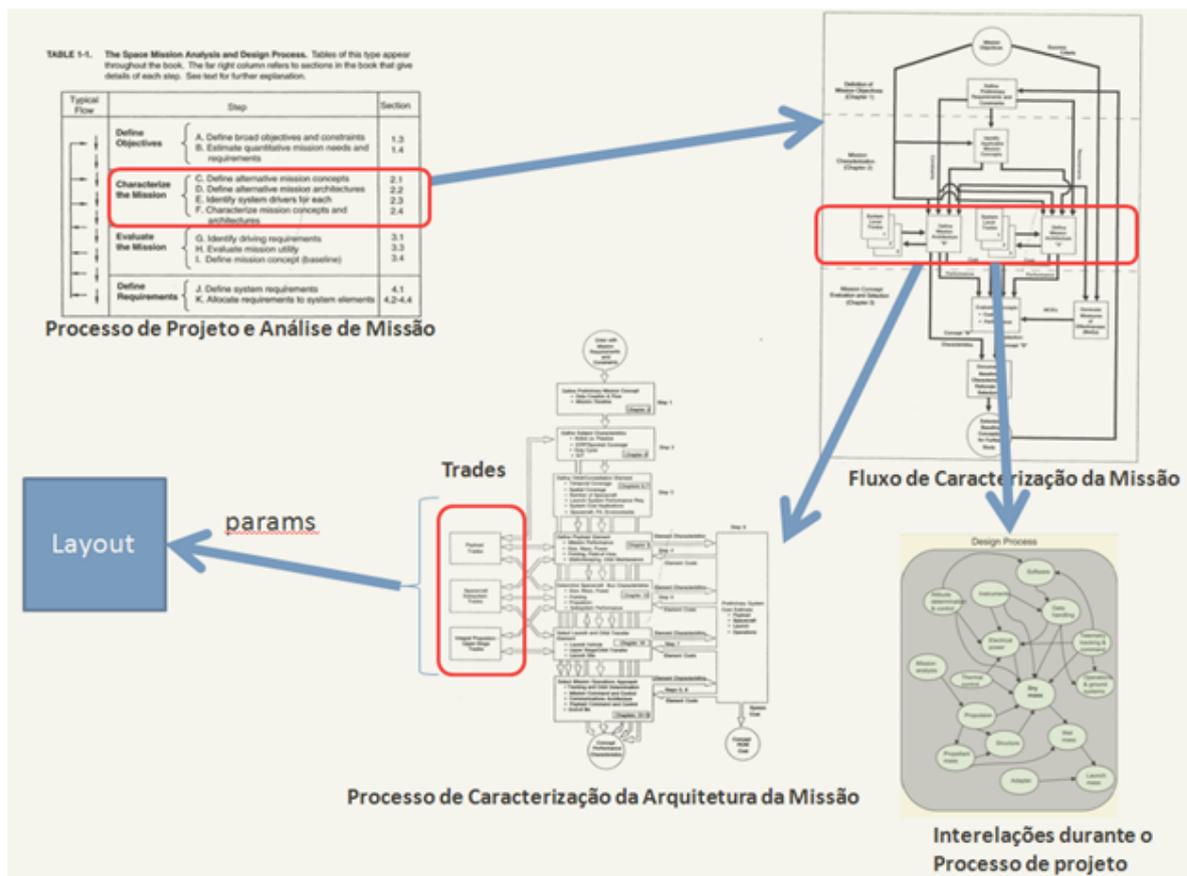


Figura 1.1 - Contexto geral do trabalho

Fonte: Adaptado de Larson e Wertz (1999, p. 20).

1.3 Objetivos

O objetivo principal deste trabalho consiste em lidar com aspectos de interoperabilidade em uma abordagem dirigida a modelos para engenharia de sistemas,

também chamada de Engenharia de Sistemas Baseada em Modelos, do inglês *Model Based Systems Engineering (MBSE)* usada em conjunto com uma abordagem MDO, dentro de uma infraestrutura protótipo de software para um ambiente colaborativo e multidisciplinar.

Este trabalho usa o problema de layout dos equipamentos embarcados em satélites, que ocorre na fase conceitual do desenvolvimento.

A elaboração deste trabalho tem base em algumas abordagens e tecnologias de forma conjunta para dar apoio aos projetistas de software durante a fase conceitual do desenvolvimento de satélites no INPE e a importância da rastreabilidade entre requisitos e parâmetros do sistema.

Para atingir o objetivo geral proposto, foram estabelecidos os seguintes objetivos específicos:

- Analisar bibliografia específica das áreas de Desenvolvimento de Sistemas Espaciais, Engenharia de Sistemas, Interoperabilidade entre Ferramentas de Software, Técnicas e Métodos de Otimização de Projeto Multidisciplinar e Ambientes Integrados de Engenharia Simultânea de Sistemas;
- Desenvolver um protótipo de infraestrutura de software, com base em aplicativos comerciais existentes e com licenças disponíveis no INPE (IBM® Rational® Dynamic Object Oriented Requirements System® - DOORS®¹ (IBM (2014c)), IBM® Rational® Rhapsody®² (IBM (2014a))) e em softwares de pesquisa desenvolvidos pelo INPE (SatBudgets Leonor (2010) e uma ferramenta para concepção de projeto multidisciplinar para *layout* de equipamento de satélites³ (de Sousa et al. (2013))), usando os métodos de interoperabilidade estudados na fundamentação teórica;
- Realizar um estudo de caso simplificado com base nos requisitos de sistema da Plataforma Multimissão (PMM), fazendo uso de modelagem SysML e da infraestrutura de software construída;

¹Neste trabalho esta ferramenta será referenciada somente como: DOORS

²Neste trabalho esta ferramenta será referenciada somente como: Rhapsody

³Neste trabalho esta ferramenta será referenciada como: PJLayout

1.4 Metodologia de Pesquisa

Este trabalho propõe o uso dos conceitos descritos de forma conjunta com o objetivo de dar apoio à tomada de decisões dos projetistas de sistemas na fase conceitual do desenvolvimento de satélites, para o caso específico do problema do posicionamento dos equipamentos embarcados, levando em consideração o uso dos requisitos, modelos, parâmetros e métodos de otimização. Para isto foi criado um protótipo de ambiente de software, que consiste no uso de ferramentas comerciais, softwares de pesquisa, padrões para transferência de dados e codificação/programação de apoio. Este protótipo visa não somente demonstrar a possibilidade dos conceitos serem usados em conjunto (através de ferramentas de software), mas também a importância da rastreabilidade dos requisitos aos parâmetros que serão otimizados, as vantagens do uso de modelos (em especial a automatização), e a possibilidade da criação de um ambiente de Engenharia de Sistemas e MDO integrados (no caso deste trabalho são somente interoperantes) usando ferramentas já existentes, reuso de códigos existentes e programação.

Dentro do Desenvolvimento de Satélites (que usa a Engenharia de Sistemas por ser um sistema extremamente complexo), durante a Fase Conceitual, é necessária a definição de várias soluções de arquiteturas (elétrica e mecânica), dentro de cada uma destas soluções existe o problema da disposição dos equipamentos que serão posicionados nos painéis estruturais do satélite. Este problema do layout é considerado um problema tradicional de otimização e beneficia-se portanto de métodos e técnicas de MDO. A Engenharia de Sistemas vem aderindo ao uso de MBSE e uma adoção cada vez maior da SysML vem sendo estudada e mostrando ser bastante útil à ES. Ambiente para engenharia de sistemas podem ser muito complexos de serem criados, mas permitem melhorias em ambas as soluções.

É importante ressaltar que este trabalho, dentro do contexto de Engenharia de Sistemas, também tem a intenção de enfatizar a rastreabilidade entre os requisitos de missão e os parâmetros dentro dos subsistemas.

A figura 3.1 mostra, em um mapa conceitual, a visão geral das tecnologias, conceitos e abordagens usadas neste trabalho. Durante o desenvolvimento de satélites existe uma fase inicial muito crítica, chamada de fase conceitual, esta fase precisa lidar com o problema do *layout* dos componentes internos do satélite. O problema do *layout* é um problema clássico de otimização e beneficia-se da MDO para a otimização de possíveis soluções. Satélites são sistemas tipicamente complexos, que usam a Engenharia de Sistemas para o seu desenvolvimento. O uso da Engenharia de Sistemas

Baseadas em Modelos já é bastante difundida, porém com a criação da SysML a MBSE vem sendo mais explorada. Os ambientes de engenharia simultânea também estão tornando-se cada vez mais difundidos, apesar de grandes agências espaciais como a ESA e a NASA já possuírem este tipo de ambiente há bastante tempo. Uma das várias dificuldades da criação de ambientes de engenharia simultânea e do uso da MDO é o problema de interoperabilidade entre ferramentas de software, que com sua rápida evolução, estão cada vez mais presentes em todos os tipos de desenvolvimento. O problema da interoperabilidade entre software afeta não somente as ferramentas de engenharia de sistemas, e engenharias em geral, mas é muito comum no desenvolvimento de software, seja qual for sua utilização.

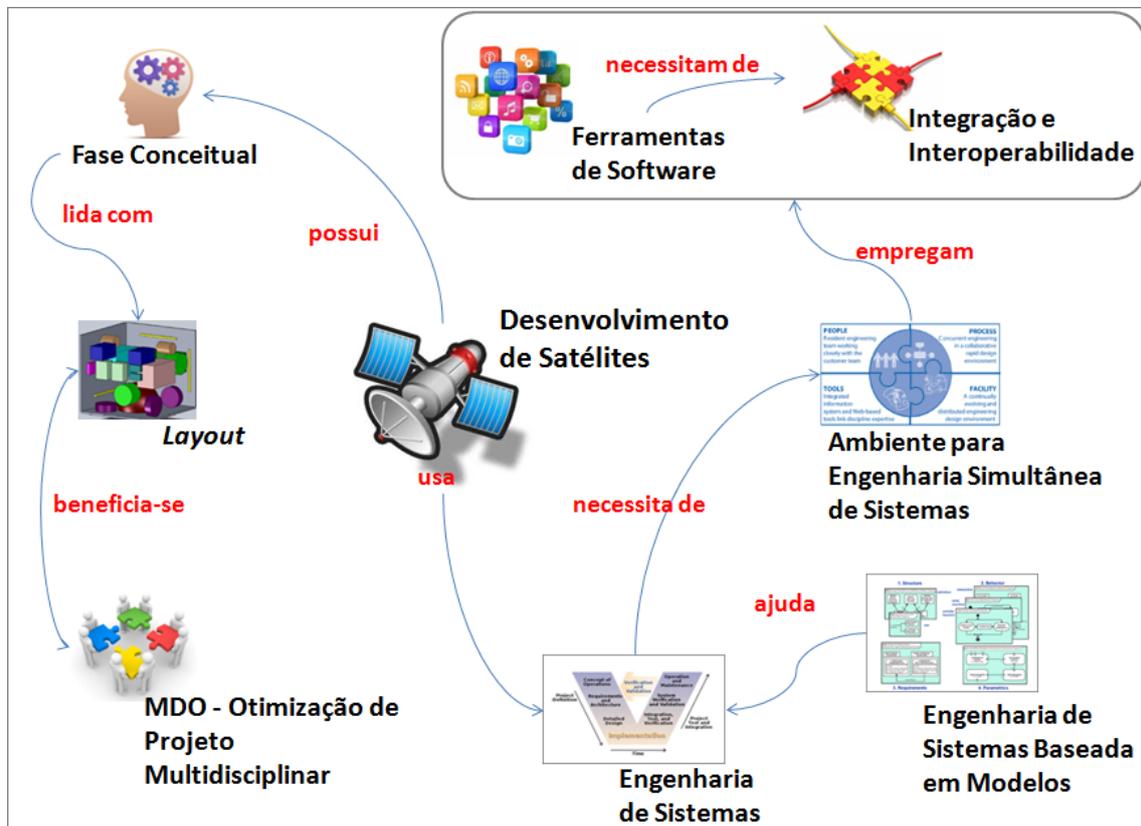


Figura 1.2 - Mapa dos conceitos e tecnologias usadas neste trabalho e aplicadas ao desenvolvimento de satélites.

Para o exercício da proposta deste trabalho, foi feito o rastreamento dos requisitos de missão aos parâmetros que influenciam no *layout* dos equipamentos para a otimização, evidenciando a importância da rastreabilidade e a interdependência entre os requisitos, subsistemas e parâmetros em um projeto espacial. Para a modelagem

foram feitos alguns diagramas usando a linguagem SysML, e foi implementada uma arquitetura MDO para a otimização do *layout* através de uma infraestrutura de software.

Sumariamente, com o intuito de propor melhorias, durante a fase conceitual do desenvolvimento de sistemas espaciais, é proposta uma abordagem orientada a modelos para projetos de engenharia e o uso de otimização em projeto multidisciplinar, para o problema do *layout*, dentro de um processo de Engenharia de Sistemas.

1.5 Organização do texto

Esta dissertação, é organizada da seguinte forma. No capítulo 2, é apresentada a fundamentação teórica desse trabalho com conceitos básicos estudados e também outros trabalhos relacionados com o problema em questão. O capítulo 3 define a metodologia usada e o desenvolvimento do trabalho, no capítulo 4, é mostrada a implementação e configuração das tecnologias e ferramentas usadas, no capítulo 5, é usada, como estudo de caso, a PMM, através de uma modelagem reduzida até a otimização e os resultados obtidos do layout. As considerações finais são apresentadas no capítulo 6 e, por fim, as referências bibliográficas.

2 FUNDAMENTAÇÃO TEÓRICA

Neste trabalho serão apresentados os diversos conceitos, tecnologias e padrões aplicados nas áreas de Engenharia de Sistemas, Engenharia de Sistemas Espaciais, Desenvolvimento de Software e Otimização. Um mapa mental (Figura 2.1) foi criado para ilustrar e guiar o estudo teórico que será apresentado.

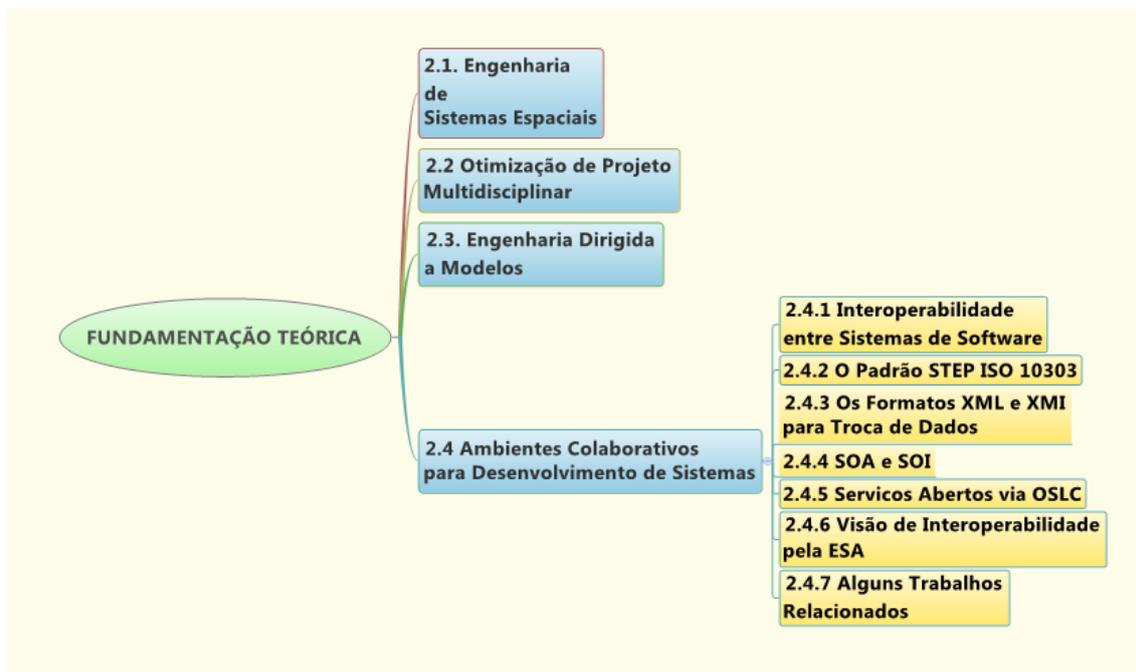


Figura 2.1 - Mapa mental usado para a estruturação do estudo teórico.

A fase conceitual do desenvolvimento de satélites é considerada uma das mais críticas, pois é nessa fase que são tomadas algumas decisões que acompanharão o projeto até o fim do desenvolvimento e influenciarão todo o ciclo de vida do produto. Uma dessas decisões é a disposição dos equipamentos dentro do satélite, esse *layout* deve levar em conta vários fatores que afetam todas as áreas das engenharias específicas envolvidas no desenvolvimento e em todo o ciclo de vida do satélite e da missão.

2.1 Engenharia de Sistemas Espaciais

Segundo a norma [IEEE \(1998\)](#) apud [Loureiro \(1999\)](#), a Engenharia de Sistemas é uma abordagem multidisciplinar colaborativa de engenharia para derivar, desenvolver e verificar uma solução balanceada ao longo do ciclo de vida e que atenda às

expectativas dos interessados (*stakeholders*). Essa abordagem antecipa para as etapas iniciais do desenvolvimento de um produto os requisitos dos processos do seu ciclo de vida, tais como manufatura, montagem e testes, manutenção, logística e descarte do produto.

A Engenharia de Sistemas usa técnicas, métodos e técnicas consagradas para minimizar as restrições naturais de projetos e obter os resultados esperados com alta confiabilidade.

Na área espacial, [Larson e Wertz \(1999\)](#) definem a Engenharia de Sistemas Espaciais como um processo iterativo e disciplinado que emprega práticas aceitas de engenharia para planejar, definir, controlar, projetar, analisar, integrar, verificar e então apoiar a produção, testes e operações de produtos de hardware, software, programas ou sistemas espaciais.

A Engenharia de Sistemas proporciona grandes vantagens ao desenvolvimento de sistemas complexos, como é o caso de satélites e missões espaciais, nos quais a fase de projeto conceitual é bastante crítica. Um exemplo de fluxo de trabalho para o desenvolvimento de missões é mostrado na [Figura 2.2](#)

Satélites e Missões são considerados sistemas complexos e beneficiam-se da Engenharia de Sistemas para o seu desenvolvimento. Vários ambientes de Engenharia de Sistemas têm sido criados e usados por organizações, empresas, instituições e universidades permitindo que especialistas de diversas disciplinas possam pôr em prática a Engenharia Simultânea de Sistemas.

Com o rápido avanço das tecnologias atuais e o aumento da demanda, os produtos e sistemas têm se tornado cada vez mais complexos, envolvendo diversas áreas da engenharia, como Mecânica, Eletrônica, Computação. Devido a esses novos desafios, o uso da Engenharia de Sistemas tornou-se mais popular. No cenário aeroespacial, no qual a confiabilidade é fundamental, a engenharia de sistemas é essencial e é considerada consolidada.

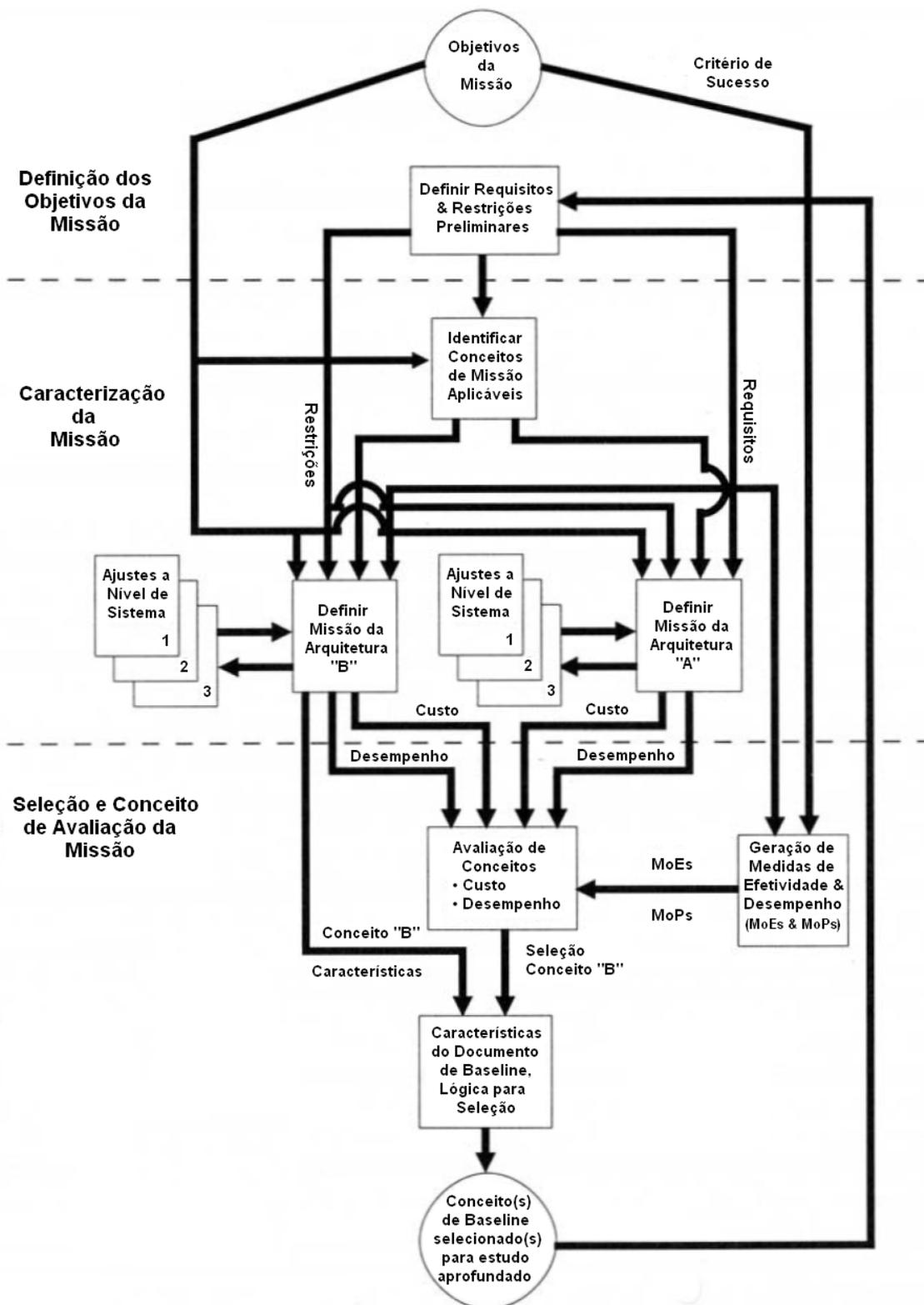


Figura 2.2 - Fluxo de trabalho para exploração de conceitos de missões espaciais. Fonte: Adaptado de Larson e Wertz (1999, p. 20).

2.2 Otimização de Projeto Multidisciplinar (*MDO - Multidisciplinary Design Optimization*)

Existem várias descrições para a MDO, Sobieszczanski-Sobieski (1995) considera a MDO uma nova disciplina que tem o papel de conectar todas as outras disciplinas do campo da engenharia, esta nova disciplina apresenta um esforço no campo de pesquisa e uma ajuda ao projeto de sistemas de engenharia. Em 1997, Sobieszczanski-Sobieski e Haftka (1997) definiu MDO como "uma metodologia para o projeto de sistemas nos quais fortes interações entre as disciplinas motivam projetistas a manipular variáveis das diversas disciplinas simultaneamente."(traduzido de Sobieszczanski-Sobieski e Haftka (1997))

Martins e Lambey (2013) definem a MDO como um campo da engenharia com foco no uso de otimização numérica para o projeto de sistemas multidisciplinares. A MDO, por ter caráter multidisciplinar enfrenta desafios muito maiores dos encontrados em uma única disciplina.

Uma das principais motivações da MDO é que o desempenho de um sistema multidisciplinar não pode ser considerado no desempenho de cada uma das disciplinas isoladamente, e sim no comportamento do todo, levando em conta as suas interações (MARTINS; LAMBEY, 2013). Essa visão sistêmica é um dos principais conceitos da Engenharia de Sistemas. Portanto, dentro do contexto apresentado, uma abordagem de Engenharia de Sistemas e MDO devem estar interligadas, não só no processo, mas também integrados no ambiente de desenvolvimento através da interoperabilidade entre as ferramentas envolvidas.

De acordo com Agte et al. (2010), a MDO tem raízes na otimização estrutural. Um dos primeiros usos do formalismo da Programação Não-linear (NP) para otimização estrutural foi feita por Schmit 1971 em uma simples treliça de três barras. Desde então o seu uso tem se espalhado da engenharia estrutural para outras disciplinas, inicialmente em sistemas aeroespaciais nos quais a massa é crucial (AGTE et al., 2010).

Na Engenharia de sistemas, o uso da MDO na fase inicial do processo e ferramentas de análise, os projetistas podem diminuir o custo e tempo do ciclo de vida do projeto (MARTINS; LAMBEY, 2013).

Devido aos grandes avanços tecnológicos, principalmente na área de processamento computacional, a otimização tem ganhado grande impulso para a sua aplicação em diversas áreas do conhecimento, usando não mais somente um código de programa-

ção e sim diversas ferramentas computacionais acopladas. Com o aumento no uso de ferramentas de software pelas engenharias, e também o aumento da variedade de ferramentas de software disponíveis para uso, é cada vez mais difícil escolher apenas uma ferramenta para ser usada, ou somente um algoritmo de otimização implementado. Muitas ferramentas, apesar de servirem para propósitos diferentes, podem precisar da entrada das mesmas informações, ou uma pode usar a saída da outra, criando retrabalho na entrada destas informações. Neste contexto a criação de infraestruturas de software de MDO, pode ajudar no reaproveitamento de dados, evitando retrabalho e com isso diminuindo a probabilidade de entrada de informações erradas por erro humano. Do ponto de vista de software, a criação de uma infraestrutura de MDO, apresenta os mesmos problemas enfrentados na criação de uma infraestrutura de um ambiente de engenharia simultânea, o problema de interoperabilidade entre ferramentas de software.

Para o uso da MDO é necessário que seja montada uma infraestrutura específica para o problema em questão, para isso é necessário que seja feita uma análise dos modelos e ferramentas específicos de cada disciplina, como os dados se comunicam (interfaces) e uso de algoritmos de otimização (atualmente existem vários). Na Figura 2.3 é mostrado um exemplo de um framework que expõe de maneira didática um ambiente de MDO.

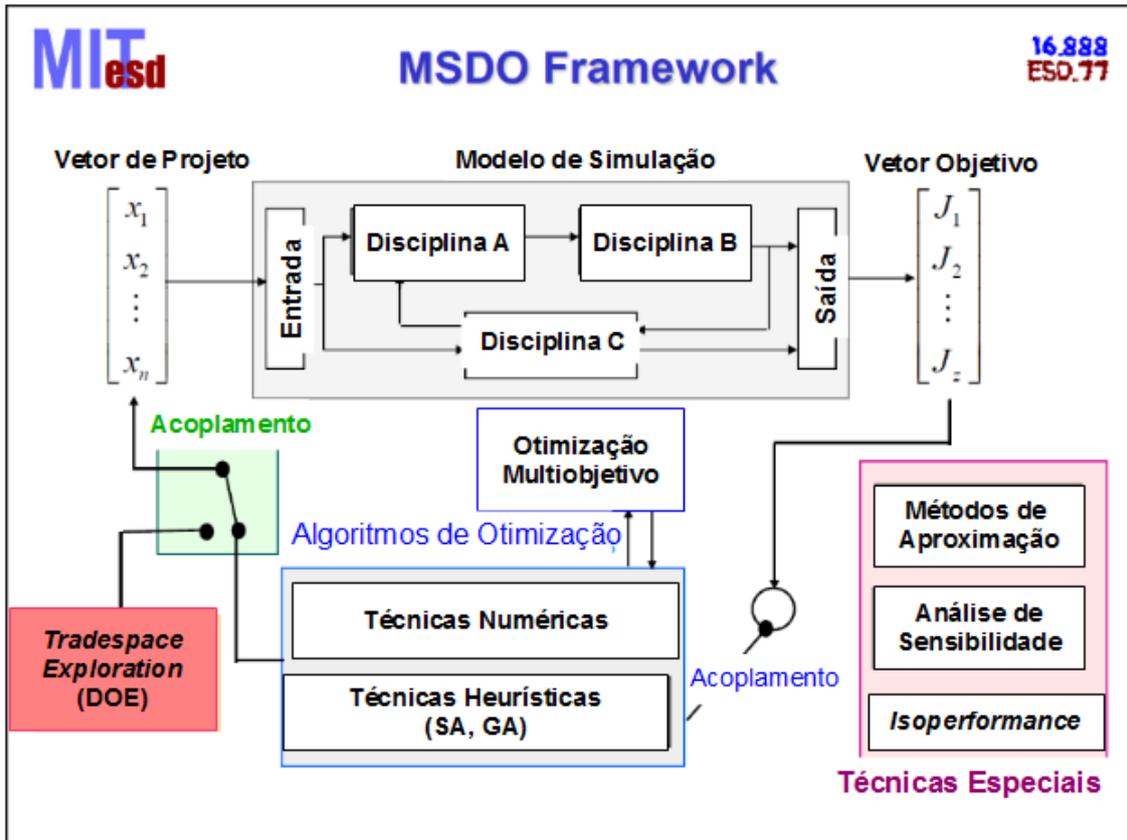


Figura 2.3 - *Framework* usada para Otimização de Projeto de Sistema Multidisciplinar (*Multidisciplinary System Design Optimization - MSDO*).
 Fonte: Weck e Willcox (, p. 31).

Durante o projeto conceitual é feita a exploração de conceitos alternativos e estudos de compensação (*trade studies*): Para este estudo são usadas ferramentas que envolvem diferentes disciplinas. Após a escolha de um conceito, este evolui para componentes e subsistemas, os projetos dos subsistemas passam a ser mais detalhados, resultando em uma interação mais limitada entre as disciplinas (KORTE et al., 1997).

A metodologia MDO permite que seus processos sejam aplicados a qualquer fase do projeto de um sistema complexo de engenharia. Korte et al. (1997) demonstra o uso da metodologia MDO durante o Projeto Preliminar, e mostra algumas vantagens, como melhorias nos objetivos e facilidade com que as variáveis multidisciplinares de projeto foram incluídas no processo de projeto. Também mostra como um sistema computacional distribuído pode ser usado em um problema de MDO (KORTE et al., 1997).

2.3 Engenharia Dirigida a Modelos (*MDE - Model Driven Engineering*) e a Linguagem de Modelagem SysML)

Tradicionalmente, projetos de grande porte usam a abordagem de engenharia de sistemas centrada em documentos. Segundo [Friedenthal \(2009\)](#), essa abordagem possui algumas limitações que podem causar deficiências. Nessa abordagem a informação está espalhada em diversos documentos, dificultando o entendimento de aspectos particulares do sistema, da rastreabilidade e da avaliação do impacto de mudanças, causando deficiência na sincronização entre requisitos de sistema e os projetos de hardware e software. Outra desvantagem é a dificuldade de manutenibilidade e reuso de requisitos e informações do projeto. Estas limitações podem resultar em ineficiências ou problemas de qualidade que podem aparecer durante a integração e testes ou após o sistema ser entregue ao cliente.

Devido a essas limitações da abordagem centrada em documentos, existe uma tendência crescente no uso da abordagem centrada em modelos para desenvolvimentos de sistemas. A Engenharia Dirigida a Modelos (MDE) é uma abordagem para desenvolver sistemas complexos, confiáveis e reutilizáveis.

Na engenharia de sistemas complexos existe uma grande tendência na mudança da abordagem centrada a documentos para a abordagem centrada em modelos, pois traz muitos benefícios como os citados por [Friedenthal \(2009\)](#): (i) melhorias nas comunicações, (ii) auxílio no gerenciamento, (iii) melhoria na qualidade do projeto, (iv) apoio aos ciclos de vida e (v) melhora na captura de conhecimento. Além disso, o mais importante para [Bezivin e Gerbe \(2001\)](#) é que os modelos não são mais usados somente para documentação, mas para guiar o desenvolvimento de sistemas.

A explicação de [Friedenthal \(2009\)](#) sobre a MBSE também converge para as vantagens citadas anteriormente. A Engenharia de Sistemas Baseada em Modelos (*MBSE - Model Based Systems Engineering*) usa modelagem de sistemas para apoiar a análise, especificação, projeto e verificação de sistemas em desenvolvimento. Esta abordagem melhora as comunicações entre o time de desenvolvimento, a qualidade de especificação e de projeto e permite o reuso de especificações e artefatos ([FRIEDENTHAL, 2009](#)).

Segundo [Wall \(2000\)](#), combinações de modelos, métodos de engenharia simultânea e projeto dirigido a cenários têm sido usados nas fases iniciais de projetos com ótimos resultados. Modelos de requisitos capturam requisitos da mesma forma que requisitos baseados em texto; porém, a grande vantagem é a possibilidade de serem

executados, permitindo que sejam simulados.

De acordo com a comunidade científica, existem várias definições de modelos. Para [Bezivin e Gerbe \(2001\)](#), um modelo é uma simplificação de um sistema construído com um objetivo, e ele deve responder às questões, dentro do domínio do objetivo geral, como o próprio sistema responderia. [Kleppe et al. \(2003\)](#) considera um modelo como uma descrição de um sistema (ou parte dele) em uma linguagem bem definida. [Seidewitz \(2003\)](#) define um modelo como um conjunto de afirmações sobre o sistema em estudo, ou seja, para ele, modelos são como especificações. [Friedenthal \(2009\)](#), em seu livro, define modelo como uma representação de conceitos que podem ser materializados no mundo real, considerando os modelos como abstrações, representações e protótipos físicos.

Na área de Engenharia de Software, [Sommerville \(2011\)](#) define modelo como "representação abstrata que permite descrever e/ou prever comportamentos específicos de um sistema através do estudo de suas características relevantes".

Simplificando, um modelo pode ser definido como uma representação ou uma descrição que ilustra um objeto, um sistema ou um conceito. Modelos podem antecipar vários aspectos dos objetos e do sistema, não somente suas características, mas também como interagem uns com os outros, bem como podem prover conhecimento que ajude na tomada de decisões sobre o projeto e sua construção antes mesmo da sua construção de fato.

Em 2000, a OMG criou a Model Driven Architecture (MDA), um padrão que segue a abordagem MDE para obter reusabilidade, portabilidade e Interoperabilidade. A MDA é implementada em linguagens de modelagem, como a Modelica, a AutoSar, a UML e a SysML. [Favre \(2004\)](#) ressalta que MDE não é MDA, o padrão MDA é uma variação particular, concretizada a partir da abordagem MDE.

Os modelos de linguagens padronizadas e robustas são necessários para a MDE. A SysML é uma linguagem de modelagem gráfica de sistemas da Object Management Group (OMG-SysML), usada para especificação, análise, projeto e verificação de sistemas complexos. Ela nasceu da iniciativa do International Council on Systems Engineering (INCOSE) com a OMG para estender a Unified Modeling Language (UML) para a Engenharia de Sistemas com a intenção de unificar a modelagem de Engenharia de Sistemas, assim como a UML hoje está consolidada para modelagem na Engenharia de Software (OMG). A figura 2.4 mostra visualmente a criação da SysML.

Na Engenharia de Sistemas são utilizadas várias ferramentas de modelagem, sendo que com o uso da SysML é possível modelar requisitos, estrutura e comportamento, suprimindo as necessidades de modelagem da Engenharia de Sistemas. A adoção de SysML no desenvolvimento de sistemas em diversas áreas de atuação é ilustrada pelos trabalhos científicos publicados sobre este tema. Esses trabalhos motivam e fundamentam a escolha da SysML, como linguagem de modelagem para sistemas aeroespaciais

Segundo Wolkl e Shea (2009), engenheiros tem trabalhado com ferramentas de software que lidam com um único estágio da fase conceitual de projeto, em geral, as ferramentas de gerenciamento de requisitos, sendo a integração com outras ferramentas de software nos outros estágios geralmente pobre. Wolkl e Shea (2009) propõe um modelo de produto integrado construído usando SysML e apresenta a aplicabilidade da SysML como uma linguagem formal de modelagem apropriada para a fase conceitual de projeto mecânico. Algumas das vantagens apontadas por ele são a codificação do conhecimento de recursos do projeto conceitual antes em papel em uma forma que pode ser lida pelo computador; a facilidade de modelagem para os projetistas, que fornece um padrão de modelagem no qual a biblioteca pode ser estendida e customizada; e aumenta a potência de reusabilidade dos modelos.

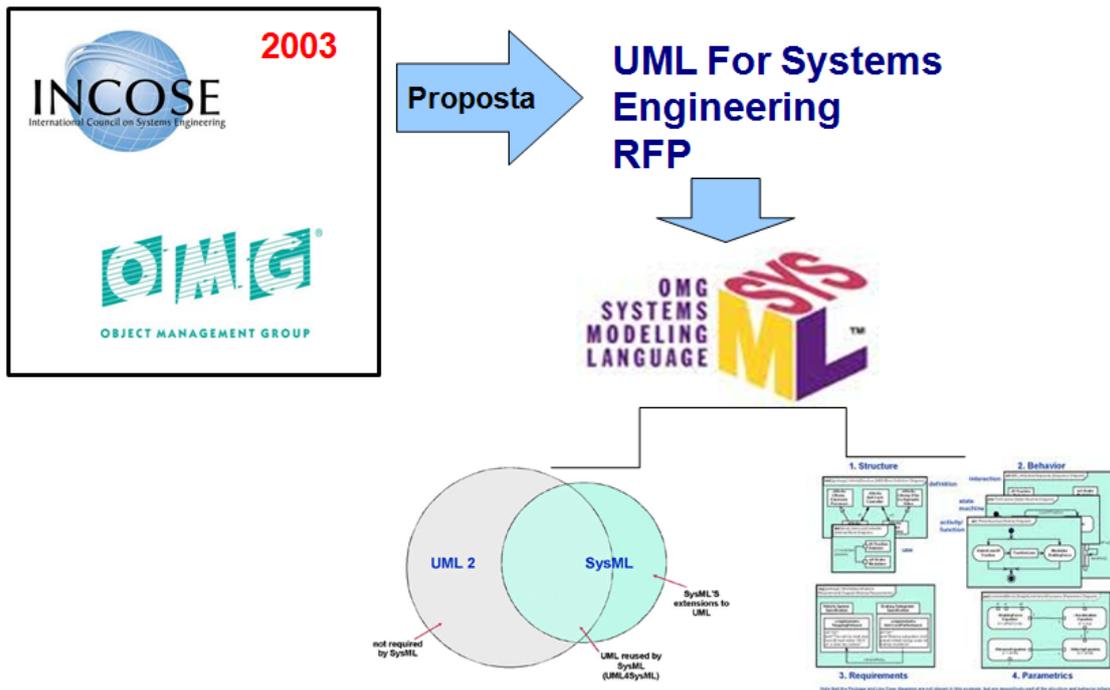


Figura 2.4 - Sysml.

Fonte: Adaptado de Object Management Group (2013b)

2.4 Ambientes Colaborativos para Desenvolvimento de Sistemas

Com a evolução dos sistemas de software e comunicação de dados via redes, os ambientes colaborativos vêm sendo muito utilizados, principalmente para o desenvolvimento de software.

Ambientes de engenharia simultânea podem ser beneficiados por ambientes colaborativos, adaptando-os aos processos envolvidos pelas empresas envolvidas.

Na área espacial, a Engenharia de Sistemas vem sendo utilizada em grandes agências espaciais através ambientes de Engenharia Simultânea de Sistemas. Essas agências perceberam a necessidade da criação de ambientes de Engenharia Simultânea Sistemas para melhorar a comunicação entre seus desenvolvedores, bem como de infraestruturas de software que possibilitassem a troca de dados de forma consistente e eficiente.

Diferentes processos e equipes separadas geograficamente são alguns dos desafios en-

contrados na criação de um ambiente de desenvolvimento colaborativo, mas do ponto de vista da infraestrutura de software, o desafio mais crítico é a troca de dados, ou seja, a comunicação/interoperabilidade entre ferramentas de software. Atualmente é indiscutível que o apoio computacional é indispensável ao desenvolvimento de sistemas. Inúmeras ferramentas são desenvolvidas com o propósito de apoiar o desenvolvimento de sistemas, sejam ferramentas da Engenharia Específica ou de Engenharia de Sistemas.

Essas ferramentas, chamadas ferramentas CASE (*Computer Aided Systems Engineering*), frequentemente precisam trocar dados entre si e devido à falta atual de padrões para interfaces, este problema muitas vezes é resolvido manualmente (o engenheiro precisa inserir os dados manualmente), ou através de dispendiosos esforços de programação.

2.4.1 Interoperabilidade entre Sistemas de Software

O problema da falta de interfaces para comunicação entre diferentes ferramentas, apesar de ser um problema antigo, ainda não possui uma solução consolidada. As diferentes ferramentas são produzidas por diferentes fabricantes, usando modelos de dados e formas de armazenamento diferentes, sendo que a falta de padrões é hoje um grande desafio para os arquitetos de software.

A Engenharia de Software vem crescendo mais rapidamente que a Engenharia de Sistemas, e várias tecnologias e conceitos do desenvolvimento de software estão sendo incorporados ou estendidos para o desenvolvimento de sistemas, como é o caso da UML e SysML já mencionado. Logo, o problema da interoperabilidade existe também no desenvolvimento de software, e algumas soluções são descritas a seguir.

2.4.2 O Padrão STEP ISO 10303

Com o uso de várias ferramentas CASE nas engenharias, o padrão STEP ISO 10303 foi proposto como uma solução para a troca de dados entre as ferramentas. O uso desse padrão possibilita a habilidade de mover dados de uma ferramenta para outra através da criação de interfaces.

A ISO 10303 constitui uma série de documentos que descreve como deve ser feita a troca e o compartilhamento das informações usadas durante todo o ciclo de vida do produto. Para cada disciplina de engenharia existe um protocolo de aplicação próprio que representa o modelo de informação específico para o domínio.

Apesar do sucesso do padrão STEP, muitos questionam a sua velocidade de desenvolvimento e implantação, apontando a rápida evolução da XML (eXtensible Markup Language). A XML nasceu das necessidades do comércio eletrônico, no qual as informações normalmente são formulários, já os métodos tradicionais de troca de informações de modelos de produto são desenhos, e estes possuem relacionamentos complexos, o que torna o trabalho do padrão STEP mais difícil.¹

2.4.3 Os Formatos XML e XMI para Troca de Dados

Diversas ferramentas de software adotam soluções de interoperabilidade baseadas nos formatos XML/XMI, explicitados a seguir.

Segundo a [World Wide Web Consortium \(W3C\) \(2012\)](#), a XML (*Extensible Markup Language*) é uma linguagem de marcação representada em um formato de texto simples para representar informação estruturada, sendo atualmente uma das linguagens mais usadas para compartilhamento de informação estruturada ([WORLD WIDE WEB CONSORTIUM \(W3C\), 2012](#)). A XML foi criada para a transferência de documentos estruturados através da Web, pois as linguagens existentes (HTML² e SGML³) não atendiam de forma apropriada a essa necessidade. Ela é uma linguagem de marcação, baseada em *tags*, que especifica como os documentos devem ser estruturados. A XML, ao contrário de HTML, não especifica nenhum conjunto pré-definido de *tags*; portanto não especifica nenhuma semântica que deve ser definida através do software que irá usá-lo. Os blocos de construção básicos da XML são chamados de elementos, que são a forma mais comum de marcação de texto. Esses elementos possuem atributos (pares de nomes e valores) e também as referências de entidade para que a marcação seja incluída em um documento ([WALSH, 2012](#)). Um exemplo de documento escrito em XML é apresentado a seguir:

¹As estruturas necessárias para que o padrão STEP tenha um documento próprio em formato XML está em desenvolvimento.

²Hypertext Markup Language

³Standard Generalized Markup Language (ISO 8879:1986 SGML) é uma norma ISO para descrever linguagens de marcação para documentos.

```

<part number="1976">
  <name>Windscreen Wiper</name>
  <description>The Windscreen wiper
    automatically removes rain
    from your windscreen, if it
    should happen to splash there.
    It has a rubber <ref part="1977">blade</ref>
    which can be ordered separately
    if you need to replace it.
  </description>
</part>

```

Figura 2.5 - Exemplo de XML.

Fonte: World Wide Web Consortium (W3C) (2012).

A manipulação das informações de um documento XML é feita através de um analisador (*parser*) de XML, que tem como finalidade extrair os elementos XML do documento e disponibilizá-los para a aplicação. Esses objetos podem então ser usados dentro da aplicação, a figura 2.6 mostra o fluxo de informação entre o documento XML e os objetos de aplicação. Existem várias APIs ⁴ para a manipulação de XML, como por exemplo: SAX (Simple API for XML), DOM (Document Object Model), JDOM ⁵, entre outros.

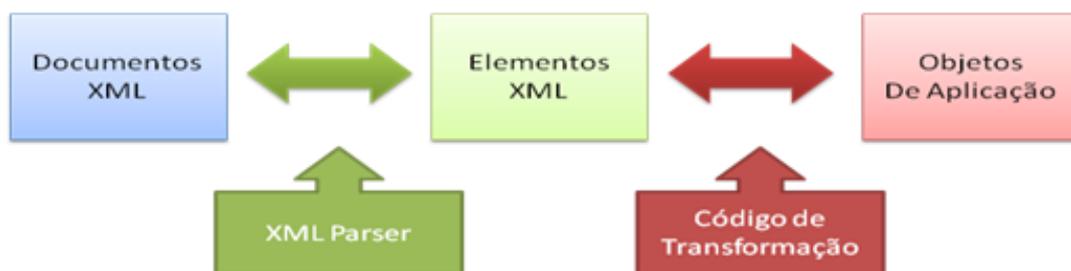


Figura 2.6 - Manipulação de XML.

O XMI (*XML Metadata Interchange*) é um padrão criado pela OMG para possibili-

⁴Application Programming Interface

⁵ API Java que possui integração com DOM e SAX

tar troca de modelos baseada em MetaObject Facility (MOF). Ele representa objetos como elementos e atributos XML, possui um mecanismo padrão para ligação de objetos dentro de arquivos, inclui identidade para objetos (permite a referência entre objetos através de um identificador (ID⁶ e UUID⁷) e possui validação de documentos XML através de esquemas XML (*XML Schemas*). O padrão XMI é mais usado atualmente para a transferência de metadados entre ferramentas de modelagem que usam UML. Algumas desvantagens no uso de XMI, é a falta de especificação que inclua a representação de informações gráficas e a grande variedade de implementações das linguagens UML e XML disponíveis, que precisam ter implementações e versões iguais para funcionarem corretamente.

2.4.4 SOA e SOI

A interoperabilidade pode ser também tratada do ponto de vista arquitetural de sistemas via Arquitetura Orientada a Serviços (*Service-oriented architecture - SOA*) e Integração Orientada a Serviços (*Service Oriented Integration - SOI*) detalhados a seguir. No desenvolvimento de software, o uso de Arquitetura Orientada a Serviços se mostra bastante adequado ao problema de interoperabilidade entre softwares. Uma breve descrição sobre SOA será apresentada a seguir.

A Arquitetura de Software é uma descrição de um sistema de software que modela a estrutura e o comportamento do sistema. Para [Pressman \(2011\)](#): “A Arquitetura de software deve modelar a estrutura de um sistema, bem como a maneira por meio do qual dados e componentes procedurais colaboram entre si”.

Um estilo de arquitetura é uma combinação de características e funcionalidades (princípios de arquitetura) nas quais a arquitetura é descrita. Estilos de arquitetura representam famílias de arquiteturas que possuem princípios comuns.

A SOA, é na verdade um estilo de arquitetura, ou famílias orientadas a serviços comuns.

Um conceito importante para o entendimento de SOA é o de serviço, que é o bloco de construção principal em SOA e representa uma funcionalidade codificada em forma de um componente de software. A Figura 2.7 mostra como os níveis de abstração são incorporados em SOA, os objetos são encapsulados em componentes e depois em serviços

⁶Identifier

⁷Universally Unique Identifier

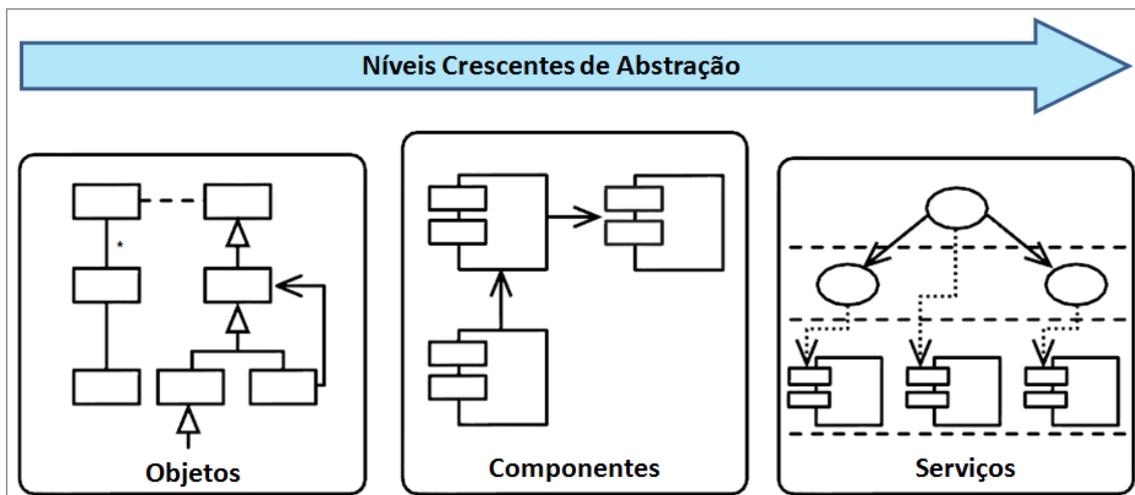


Figura 2.7 - Níveis de Abstração.

Fonte: Adaptado de [Holley e Arsanjani \(2010\)](#).

Um serviço em SOA é uma funcionalidade autônoma com as seguintes características: sem estado (*stateless*), detectável (*discoverable*), auto-descritivo (*self-describing*), combinável (*composable*), fraco acoplamento (*loose coupling*), gerenciado por uma política, independente de local, linguagem e protocolo, granularidade grossa e assíncrono ([ROSEN et al., 2008](#)).

O principal objetivo de SOA é o acoplamento fraco (*loose coupling*) entre os serviços, ou seja, sua interface é independente de sua implementação. O acoplamento fraco permite que os desenvolvedores de software possam criar aplicações através da composição de serviços sem se preocupar em como esses serviços são implementados, favorecendo a reutilização dos serviços ([ROSEN et al., 2008](#)).

A SOA é mais conhecida por ser um estilo de arquitetura, mas [Holley e Arsanjani \(2010\)](#) atentam para a questão de que classificando SOA somente como um estilo de arquitetura, como mostrado na Figura 2.8, não é possível determinar os benefícios estratégicos que SOA oferece, restringindo SOA somente à visão de TI (Tecnologia da Informação). Neste estudo a arquitetura SOA será descrita somente do ponto de vista de TI, mas deve-se ter em mente que ela também abrange outras visões, como a visão estratégica já citada, acrescentando mais benefícios ao seu conceito. A Figura 2.8 mostra a arquitetura SOA usando o diagrama de classes da UML.

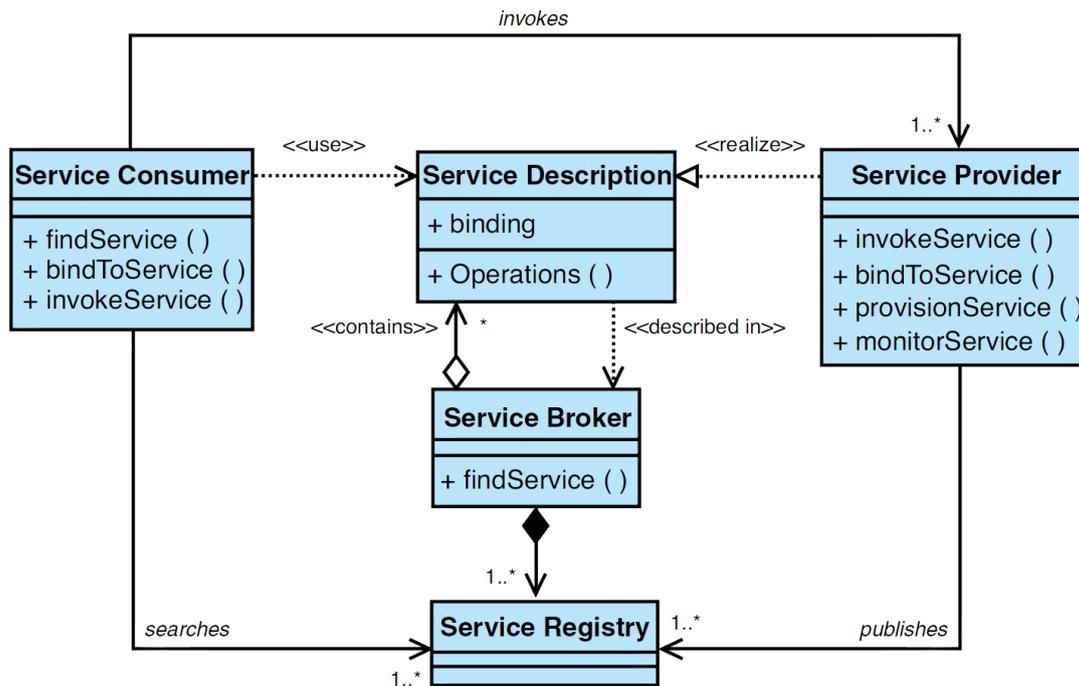


Figura 2.8 - SOA como um estilo de arquitetura.
 Fonte: Adaptado de Rosen et al. (2008).

Explicando brevemente a Figura 2.8, um consumidor (*Service Consumer*) usa uma funcionalidade, encapsulada em um serviço. O provedor (*Service Provider*) define um contrato através de uma descrição de serviço (*Service Description*). Um intermediário (*Service Broker*) usa um registro (*Service Registry*) para encontrar os serviços publicados.

Agora que a estrutura da arquitetura foi apresentada, podemos complementar a definição de serviço em SOA para: Um serviço representa uma funcionalidade de negócio que é disponibilizada através de um contrato de serviço (*Service Description*) (ROSEN et al., 2008). SOA possui um conceito enraizado de encapsular lógica de aplicação dentro de serviços que interagem, usando um protocolo de comunicações comum e tem como uma das formas para conseguir um acoplamento fraco a disponibilização de somente algumas interfaces, e as semânticas específicas de aplicação são expressas em mensagens.

Segundo Rosen et al. (2008), o uso de serviços com SOA foi primeiramente definido por Liu e Gorton (2012) como uma abordagem baseada em serviços para a integração de sistemas legados, heterogêneos e inflexíveis, que possibilitam organizações a

usarem funcionalidades fechadas dentro de aplicações existentes como serviços reutilizáveis. Essa abordagem foi chamada SOI.

As características mais importantes de SOI são: interfaces padronizadas e bem definidas; opacidade (as funcionalidades estão escondidas atrás das interfaces) e flexibilidade (os provedores e consumidores de serviço podem mudar, mas a descrição do serviço não).

2.4.5 Serviços via OSLC

Apesar de muito ser discutido sobre interoperabilidade entre ferramentas de software, pouco foi feito até agora para facilitar essa tarefa. Uma solução para esse problema seria encontrar um conjunto de especificações que fosse seguido por todos os fabricantes/desenvolvedores de software no desenvolvimento de seus produtos, mas isso é praticamente inviável, já que é impossível que todos cheguem a um acordo comum. Além disso, essa solução iria comprometer a evolução de soluções para as ferramentas que nascem dessa diversidade de fabricantes e desenvolvedores de software (OSLC WORKGROUPS, 2012).

Outra alternativa seria encontrar uma ferramenta única que atendesse todas as necessidades do ciclo de vida de um produto, o que também é inviável, pois cada produto é diferente e é gerenciado e desenvolvido de uma forma diferente. Além disso, usando essa solução, o mais provável é que adaptemos as nossas necessidades à ferramenta e não ao contrário como deveria ser.

Existe a necessidade de encontrar uma solução que permita que os clientes possam escolher ferramentas com base nas suas necessidades sem se preocupar em como elas irão interoperar, quais serão os fabricantes, se serão ferramentas de código fechado ou aberto, ou se serão usadas ferramentas desenvolvidas pela própria empresa (*in-house*⁸). Esta necessidade consiste na essência para a criação da (OSLC WORKGROUPS, 2012).

A OSLC (Serviços Abertos para Comunicações no Ciclo de Vida), do inglês (*Open Services for Lifecycle Communications*) é uma comunidade que está desenvolvendo um conjunto de especificações (com o mesmo nome da comunidade: OSLC) para que ferramentas de software possam fornecer e compartilhar suas informações com outras ferramentas, permitindo integração entre elas (OSLC WORKGROUPS, 2012).

⁸ferramenta desenvolvida pela própria empresa.

Durante o desenvolvimento de sistemas de engenharia simultânea são usadas várias ferramentas de software para auxiliar os engenheiros e gerentes de projetos. Essas ferramentas são chamadas de ferramentas de ciclo de vida por cobrirem todo o ciclo de vida do produto. Uma representação visual da ligação dos dados via OSLC é apresentada na figura 2.9.

Para a criação de uma plataforma de software para engenharia simultânea de sistemas, é necessário o uso de várias ferramentas de ciclo de vida (ALM⁹) para o desenvolvimento de um projeto, e um dos objetivos dessa infraestrutura é proporcionar que essas ferramentas interoperem, fornecendo um ambiente colaborativo, evitando replicação, garantindo a integridade dos dados e permitindo a rastreabilidade das informações que circulam por essas ferramentas. Esse é exatamente o objetivo da OSLC, permitir e facilitar o compartilhamento de informações entre ferramentas do ciclo de vida.

⁹*Application Lifecycle Management* é um processo contínuo de gerenciamento da vida de uma aplicação.

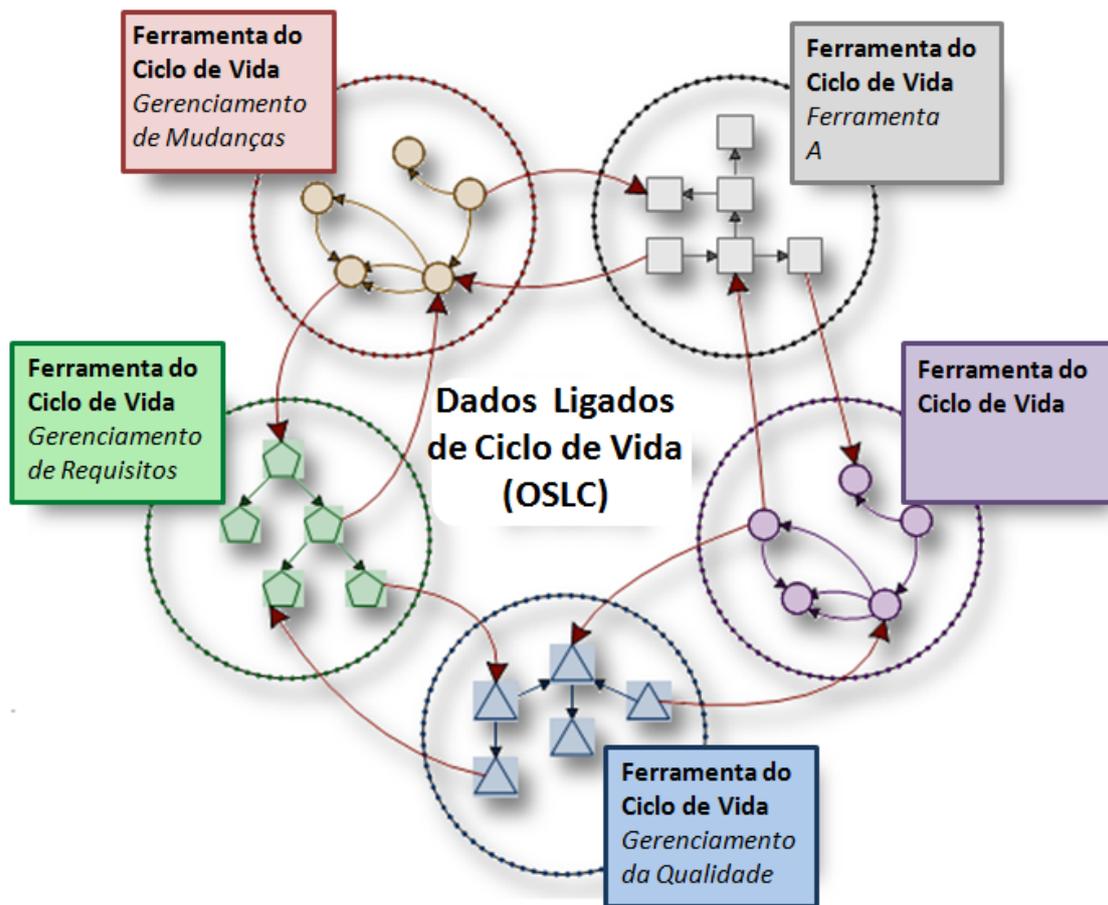


Figura 2.9 - Serviços Abertos para Comunicações do Ciclo de Vida.
 Fonte: Adaptado de OSLC Workgroups (2012).

Essa comunidade é organizada em grupos de trabalho, cada um desses grupos é responsável pelo desenvolvimento de especificações OSLC de domínio (*Domain*) para um cenário específico do ciclo de vida. O grupo Núcleo (*Core*) é responsável por desenvolver as especificações-base para a integração de ferramentas do ciclo de vida, as quais são chamadas especificações OSLC Core e as outras especificações de domínio devem obedecê-las.

A OSLC ainda é uma iniciativa bastante nova, entretanto possui o apoio de empresas significativas no mercado de ferramentas de software, como IBM, Oracle e Siemens. Já existem algumas ferramentas desenvolvidas usando OSLC (<http://open-services.net/software/>), dentre elas algumas ferramentas da família Rational da IBM, como a DOORS e a Quality Manager, a Atlassian JIRA (ferramenta para gerencia-

mento de erros (*bugs*) bastante usada no desenvolvimento de software, entre outras. Existe uma comunidade aberta que está desenvolvendo um projeto chamado Lyo, através da Fundação Eclipse, que tem como objetivo desenvolver um Kit de desenvolvimento de software (SDK¹⁰) que permita a adoção das especificações OSLC.

Assim, o uso das especificações OSLC é de grande valia para a criação de um ambiente de Engenharia de Sistemas. Por essa ser uma iniciativa bastante nova e ainda ser um projeto em andamento, falta documentação mais completa e detalhada, ambientes para o desenvolvimento que facilitem a adoção de OSLC (o projeto Lyo, em fase de incubação) e, o mais importante, uma maior variedade de ferramentas que implementem as especificações.

Resumindo, a iniciativa OSLC é uma ideia promissora, porém depende de uma maior adesão por parte dos fabricantes e desenvolvedores de software para que seja consolidada. Portanto, as especificações OSLC, apesar de terem sido consideradas, não serão implementadas diretamente no software codificado neste trabalho, mas os softwares da IBM usados neste trabalho, já apresentam interfaces usadas internamente que usam estas especificações.

2.4.6 Visão de Interoperabilidade pela ESA

Na área espacial se destacam os memorandos técnicos, iniciativa da ESA para a criação de modelos de dados adequados para a troca de informações e de um repositório de dados espaciais. A figura 2.10 mostra a arquitetura implementada por [Bandecchi et al. \(1999\)](#) com foco na troca de informações entre diversas bases de dados envolvidas no desenvolvimento de uma missão espacial.

¹⁰Software Development Kit.

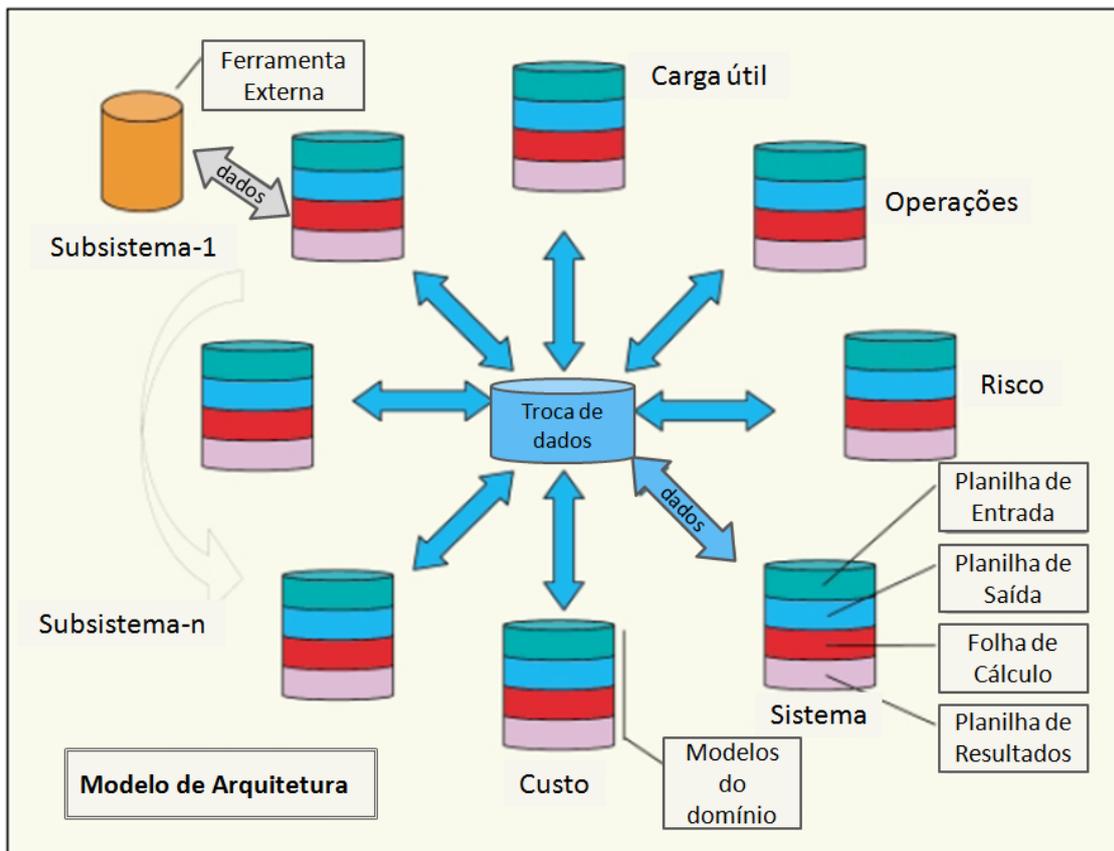


Figura 2.10 - Modelo de Arquitetura

Fonte: Adaptado de [Bandecchi et al. \(1999\)](#)[p.39].

ECSS-E-TM-10-25A: Modelo para Troca de Dados de Projeto de Engenharia (*Engineering design model data exchange*) Este Memorando Técnico define um conjunto de recomendações para dados de troca baseados em modelos para as fases iniciais de projeto de engenharia (fases 0 e A) quais sejam:

- Criar de instalações (facilities) para projeto simultâneo usando um modelo de dados comum/compatível.
- Habilitar transferência de dados de forma efetiva entre modelos pertencentes a organizações/acomodações diferentes, mas compatíveis.
- Habilitar colaboração em tempo real e atividades conjuntas entre múltiplas organizações/acomodações ([EUROPEAN COOPERATION FOR SPACE STANDARDIZATION, 2010](#)).

Para dar suporte à padronização de processos, o ambiente de engenharia simultânea da ESA (textitConcurrent Design Facility - CDF) desenvolveu o IDM (*Integrated Design Model*) que define processos e itens de dados para todas as fases do projeto, porém o memorando somente descreve as fases 0 e A.

Este memorando define um modelo de dados neutro e um conjunto de definições pré-definidas de dados baseadas no IDM para ser possível a troca e colaboração de dados entre diferentes ambientes de desenvolvimento para os ciclos de vida de todas as fases do projeto. O protocolo para a troca de dados possui duas partes o *Core Data Model*, define os tipos principais de objetos e os relacionamentos entre eles, ou seja, a estrutura dos dados que serão trocados e a *Reference data library*.

ECSS-E-TM-10-23A: Repositório de Dados de Sistema Espacial (*Space system data repository*)

Este memorando segue uma abordagem centrada em modelos, na qual várias aplicações de software são utilizadas dentro dos processos do ciclo de vida do sistema espacial. Considerando que cada aplicação seja desenvolvida por fabricantes diferentes, com arquiteturas diferentes, com linguagens e bancos de dados diferentes, cada uma delas possui seu próprio banco de dados e não é só impossível, mas também indesejável que o banco de dados dessas aplicações seja padronizado. Dessa forma, é necessária que a semântica dos dados trocados seja a mesma para todos os envolvidos. Se isso for alcançado, é possível o conceito de um repositório de dados único, mas não centralizado e sim distribuído entre várias aplicações que podem trocar dados de maneira eficiente e confiável.

Para conseguir interoperabilidade semântica entre aplicações de software, o memorando propõe que o modelo de dados conceitual de cada aplicação seja um subconjunto do modelo de dados conceitual global. O modelo de dados conceitual global é especificado nesse memorando e abrange o ciclo de vida completo de um sistema espacial. Além do modelo de dados conceitual global também irão existir bibliotecas de referência com instâncias pré-definidas. Esse Memorando Técnico também descreve, em alto nível (ver figura 2.11), como criar um repositório de dados para um projeto de sistema espacial, usando as suas especificações. (EUROPEAN COOPERATION FOR SPACE STANDARDIZATION, 2011)

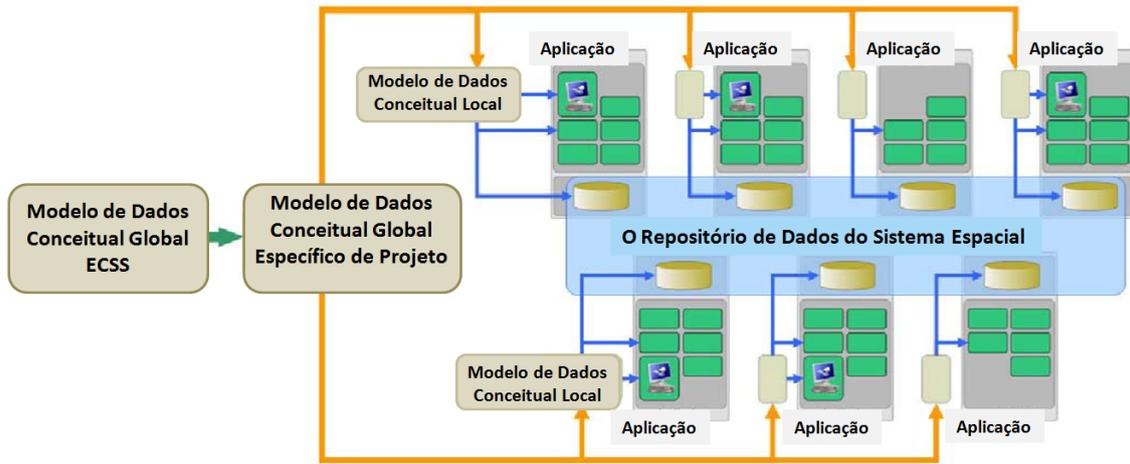


Figura 2.11 - Arquitetura do Modelo de Dados Conceitual Global da ECSS para um projeto.

Fonte: Adaptado de [European Cooperation for Space Standardization \(2011, p.20\)](#).

Os dois memorandos apresentados são complementares e visam a criação de um CDF com base em um OCDS (*Open Concurrent Design Server*) para sistemas espaciais. Cada projeto deve ter o seu próprio repositório de dados, sendo que o processo de criação/configuração de cada repositório deve ser feito individualmente para cada um deles, seguindo as orientações mostradas no memorando.

A ESA está desenvolvendo uma implementação usando essas especificações, o OCDS, o que facilitaria a implementação customizada de repositórios de dados segundo essas especificações; porém, esse ainda é um trabalho em desenvolvimento, e desenvolver independentemente essas implementações não é viável no caso deste trabalho.

Existem várias abordagens sobre arquiteturas de repositórios de dados, a abordagem da ESA mostra uma solução com foco no desenvolvimento de projetos orientado a modelos, em oposição ao orientado a documentos, e mostra uma visão do uso de diversas aplicações de software que foram desenvolvidas independentemente, com arquiteturas diferentes, que trabalham de forma diferente uma das outras, e que juntas definem e armazenam as informações necessárias ao projeto.

O processo para a criação de um banco de dados para um sistema é bastante trabalhoso e deve ser feito com muita cautela, pois as definições feitas irão causar impacto em todo o sistema. O modelo proposto no memorando necessita de um tra-

balho extenso de levantamento e a criação de um modelo conceitual específico para cada projeto, da definição das aplicações de software que serão usadas, da criação de subconjuntos de modelos de dados conceituais para cada aplicação e do mapeamento deste com o modelo conceitual do projeto. Esse trabalho não é de forma alguma simples, além disso, deve-se lembrar de que essas especificações ainda não foram compiladas como normas, ainda são memorandos, e a implementação desse modelo de dados ainda está em fase de implementação juntamente com o *OCDS* (*Open Concurrent Design Server*).

Apesar desses memorandos não serem usados diretamente na proposta deste estudo, será usada a ideia do repositório de dados de um projeto ser composto pelos repositórios das várias ferramentas usadas em um projeto.

2.4.7 Alguns Trabalhos Relacionados

No INPE, alguns trabalhos de mestrado abordaram problemas semelhantes. [Cuco \(2011\)](#), em sua dissertação de mestrado, propõe uma metodologia para a otimização do *layout* dos equipamentos de satélites artificiais. Para isso executa cálculos em ferramentas CASE de engenharia específica como SolidWorks e MatLab, e a ferramenta ModeFrontier para simular e integrar essas ferramentas. A intenção do seu trabalho é desenvolver uma metodologia aplicável ao dia-a-dia dos projetistas de satélites ([CUCO, 2011](#)).

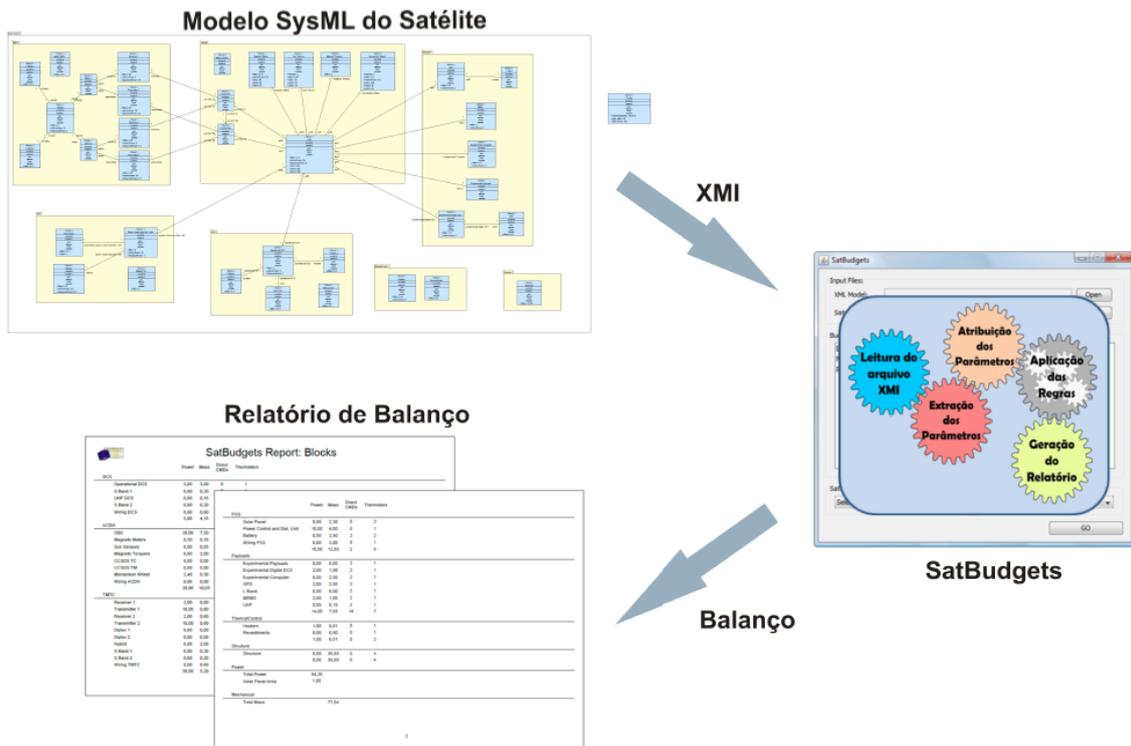
O trabalho de [Cuco \(2011\)](#) se diferencia da proposta desta dissertação, pois o foco principal é o desenvolvimento de uma metodologia com cálculos computacionais, e as ferramentas usadas somente dão apoio a esta metodologia. Nesta dissertação, as ferramentas são o foco principal, mais especificamente como elas podem interagir umas com as outras para que possam ser facilmente acopladas a diferentes ambientes de Engenharia de Sistemas. O problema do *layout* somente é usado como estudo de caso para exercitar o uso da MDO através de um caso prático.

No trabalho de [Cuco \(2011\)](#), é usado um ambiente composto por algumas ferramentas CASE conectadas a um integrador, o ModeFrontier.

O SatBudgets é um aplicativo baseado em Java e em software livre que auxilia, na fase conceitual de um projeto de satélites, gerando automaticamente balanços mecânicos, elétricos entre outros, a partir de modelo SysML e regras inseridas em seu código ([LEONOR, 2010](#)).

O fluxo de trabalho da ferramenta SatBudgets inicia com a carga do modelo SysML

do satélite via um arquivo XMI. Em seguida, as etapas de processamento são realizadas com geração de relatório e gráficos. Este fluxo pode ser visualizado na figura 4.1. Para cada modelo SysML de satélite fornecido, seus balanços podem ser selecionados na interface e processados localmente (LEONOR, 2010). O SatBudgets serviu como base para esse trabalho, sendo seu código adaptado para os novos requisitos.



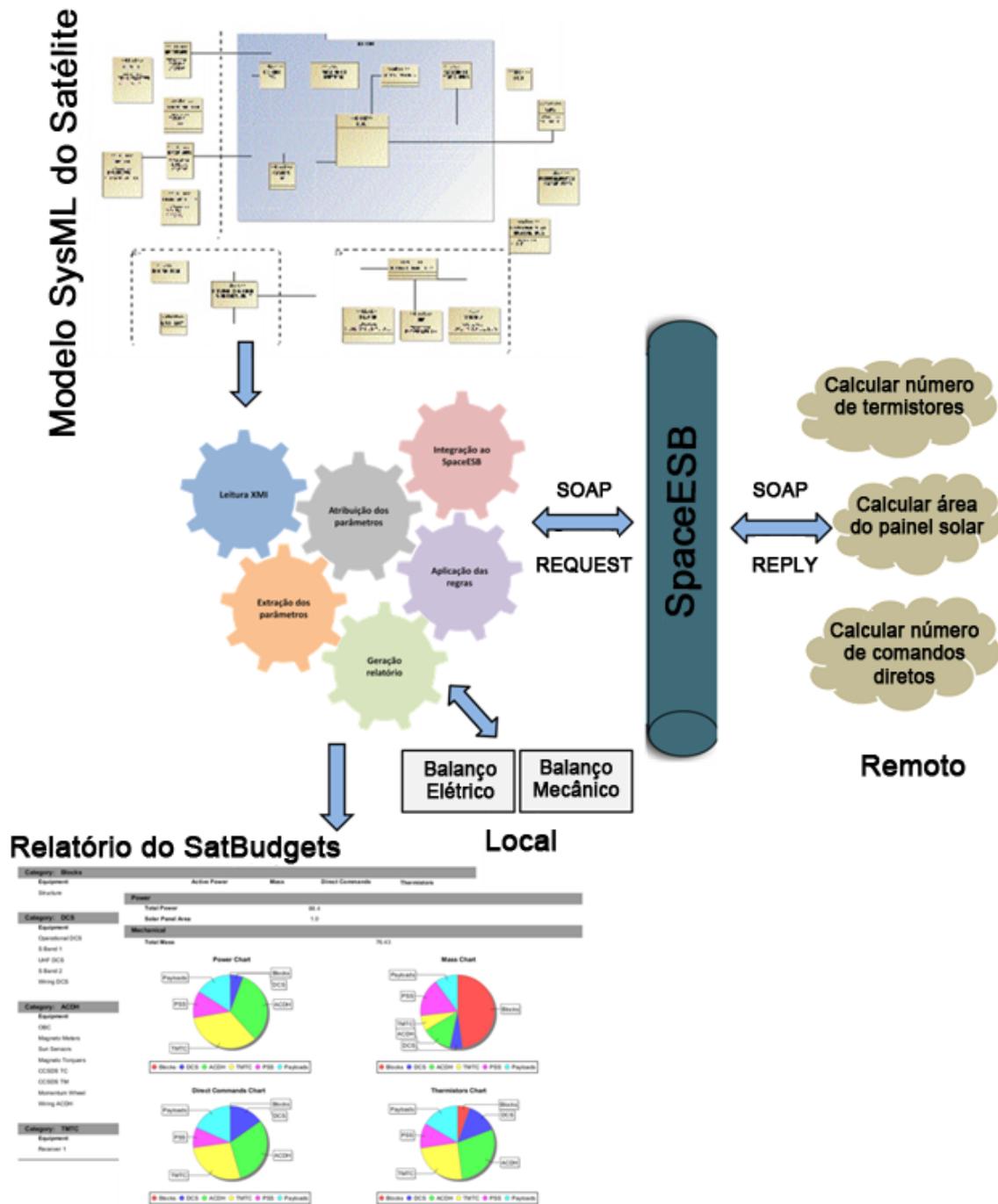


Figura 2.13 - Integração e execução entre SatBudgets e SpaceESB.
 Fonte: Souza (2011a).

Neste trabalho, Baier H.; Puehlhofer (2003) destacam a importância de integração de ferramentas e códigos de software para o desenvolvimento de satélites. Eles ressaltam

ainda que em um projeto multidisciplinar é necessário a integração de ferramentas de software em um ambiente de análise e otimização. A solução proposta tem foco na configuração de projeto dentro do ciclo de desenvolvimento de produto, na qual é preciso lidar com diferentes ferramentas e seus diferentes modelos. Foi criado um Gerente de Aplicação, codificado em MATLAB, com a responsabilidade de gerenciar dados, administrar e executar trabalhos que têm a função de um integrador, no qual podem ser integradas ferramentas de CAD, software de elementos finitos, ferramentas *in house*, entre outras. Dentro dessa aplicação, foi codificado um algoritmo evolucionário multiobjetivo para otimização do problema. Essa solução foi testada para solucionar alguns problemas de disposição de equipamentos em satélites e se mostrou promissora.

No próximo capítulo são apresentados a metodologia e o desenvolvimento desta dissertação.

3 METODOLOGIA E DESENVOLVIMENTO

Este capítulo apresenta a metodologia usada em uma abordagem qualitativa baseada no estudo dos conceitos básicos da bibliografia existente, trabalhos anteriormente relacionados, tecnologias atuais e opiniões de especialistas. E será indutiva a partir das observações feitas através do estudo de caso apresentado no capítulo 5.1. Com base na fundamentação teórica estudada, é proposto o uso de abordagens e tecnologias que possam ser usadas em conjunto para fornecer apoio aos projetistas e desenvolvedores de satélites do INPE. Para isto foi criado um protótipo de infraestrutura de software que agrega a importância da Fase Conceitual para o Desenvolvimento de Satélites, o uso de Engenharia de Sistemas e de ambientes para sua execução, o uso de modelos e padrões, a importância da Otimização de Projeto Multidisciplinar e de seus ambientes/software de execução.

Para guiar o leitor, é apresentado na figura 3.1 um diagrama com os tópicos que serão apresentados neste capítulo.



Figura 3.1 - Visão geral com estrutura dos tópicos deste capítulo.

3.1 Visão Geral

A partir do estudo feito, partiu-se do Desenvolvimento de Missões Espaciais, com foco no desenvolvimento de satélites. Dentro deste contexto, a fase conceitual do desenvolvimento é de extrema importância e o uso de Engenharia de Sistemas é fun-

damental. Atualmente algumas tendências da Engenharia de Sistemas são a adoção de Engenharia de Sistemas Baseada em Modelos e a criação de ambientes que proporcionem o uso de ferramentas e técnicas de Engenharia Simultânea de Sistemas. Para a realização prática das atividades dos conceitos citados, foi feita uma seleção de ferramentas de software já existentes no INPE que pudessem ser utilizadas neste trabalho.

Inicialmente foi adotada uma abordagem bottom-up para mapear quais os requisitos influenciam na otimização do *layout*, isso foi feito pois a quantidade de requisitos é muito grande o que tornaria uma abordagem *top-down* extremamente exaustiva para ser feita neste trabalho para toda uma missão espacial. Em seguida uma abordagem *top-down* foi adotada a partir do método tradicional de desenvolvimento de sistemas espaciais com o objetivo de indicar as fases do desenvolvimento que influenciam a definição do *layout* (AGTE et al., 2010).

3.2 Processo de desenvolvimento de projetos espaciais

3.2.1 Fase Conceitual

Durante a fase conceitual do desenvolvimento de sistemas espaciais, ocorre a Caracterização da Missão, esta atividade ocorre de forma iterativa, com refinamento de requisitos, com interação de vários fatores. Dentro deste processo iterativo, ocorrem várias soluções de compromisso (*trade-offs*) entre vários elementos e subsistemas, como é mostrado na figura 1.1 ¹. Segundo Fortescue e Stark (2001), a troca balanceada é baseada em requisitos chamados drivers de projeto, que são os requisitos que tem impacto na missão como um todo. Esta troca é feita dentro da Análise da Missão.

3.2.2 Missões Espaciais

Com o foco nos fatores que influenciam o *layout* dos equipamentos embarcados em satélites, foi feito um levantamento com o mapeamento das interdependências mais relevantes que afetam o *layout*: 1) requisitos de Missão, 2) desenvolvimento de satélites, 3) estrutura mecânica e 4) configuração mecânica (onde é definido o *layout*).

Esta dependência entre estes fatores é de extrema importância, e ressaltada por inúmeros autores da área, como Fortescue e Stark (2001), que atentam que quase

¹A figura 1.1 não tem a intenção de ser legível, só de mostrar uma visão geral, os elementos da figura são figuras já apresentadas no capítulo 2.1 (legíveis)

quaisquer mudanças nos parâmetros da Missão causam grandes efeitos nas especificações dos subsistemas relacionados ao satélite, e que a escolha da órbita da Missão tem grande impacto no projeto do satélite.

Baseados nos cinco elementos fundamentais de missões espaciais: o tópico da missão, a órbita, o sistema de solo, a arquitetura das comunicações e os requisitos, é possível definir uma "arquitetura" preliminar da Missão Espacial e usar o método de decomposição para continuar com o processo de refinamento.

Através da decomposição em elementos da arquitetura da missão espacial, mostrada na figura 3.2, pode ser notada uma estrutura em forma de árvore.

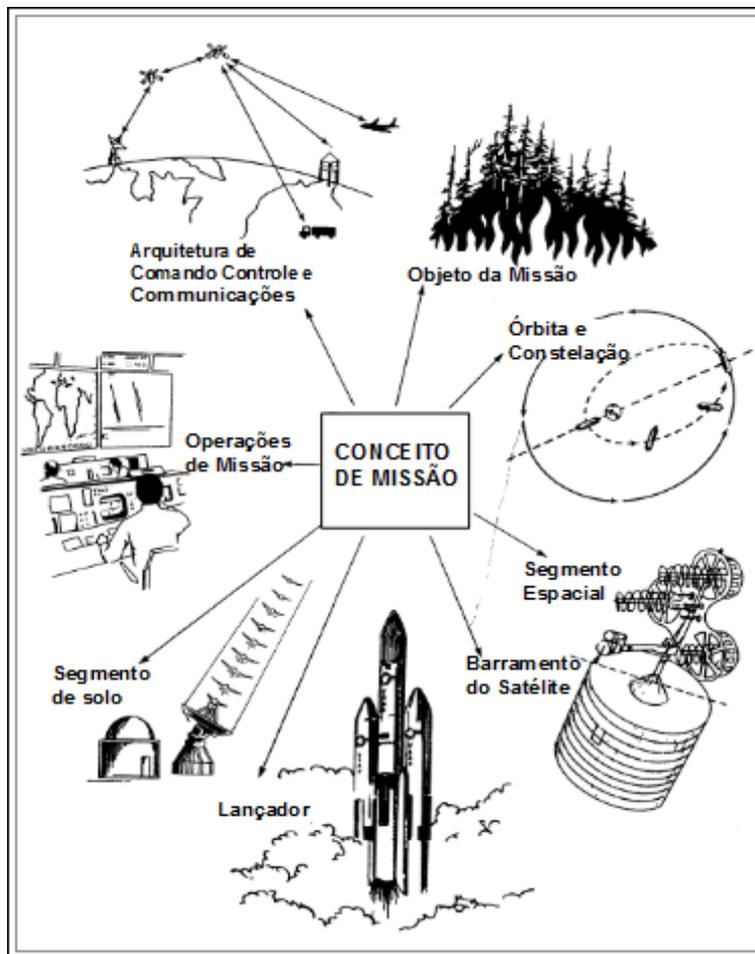
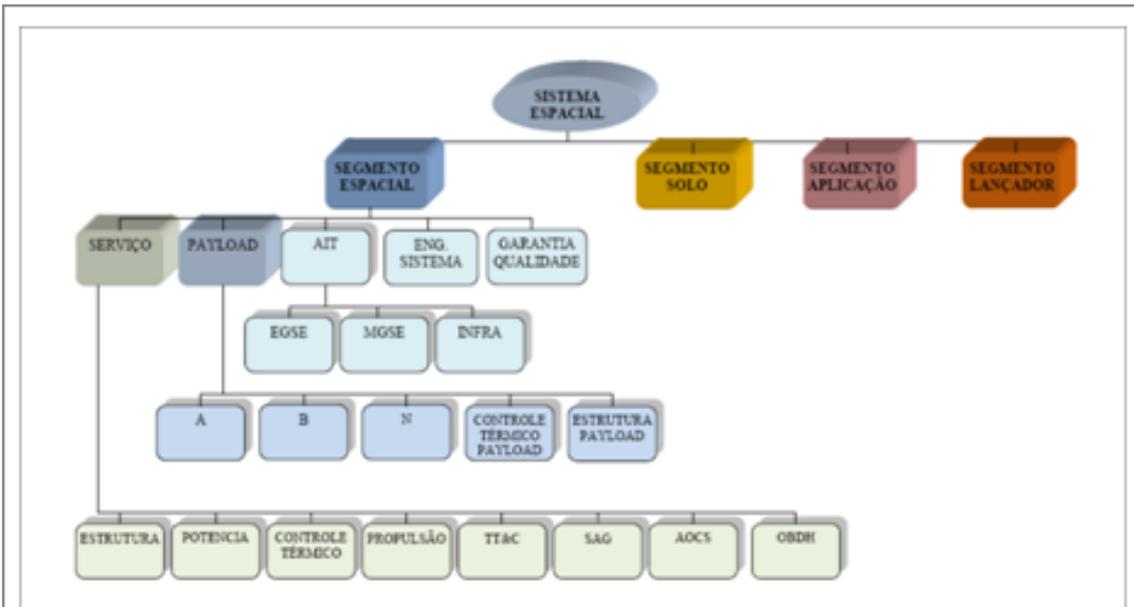


Figura 3.2 - Elementos da Missão

Fonte: Adaptado de Larson e Wertz (1999)

Na estrutura em árvore, mostrada na figura 3.3, fica clara a dependência entre a raiz (Missão) e todos seus nós derivados. Em uma decomposição mais detalhada, o número de nós tende a crescer rapidamente e começam a aparecer as dependências diretas entre nós irmãos, o que será mostrado mais adiante.



Nível Hierárquico	Conteúdo
Missão	Necessidades + Objetivos
Sistema	Sistema espacial completo = Segmento Espacial + Segmento Lançador + Segmento Solo.
Segmento	Segmento espacial (Satélite , sondas.), Segmento lançador (foguetes+estação de controle), Segmento solo (estação terrena de controle + estação de processamento de dados.)
Elemento	Satélite = Estrutura + Potência + AOCS + ...
Subsistema	Potência = Painel Solar + Bateria + ...
Componente	Painel Solar = células solares + estrutura de fixação + ...
Parte	Célula solar

Figura 3.3 - Decomposição do Sistema

Fonte: Adaptado de SILVA JÚNIOR (2011, p.100 e 105)

3.2.3 Desenvolvimento de Satélites

Na decomposição do segmento espacial ao satélite, é verificado que as decisões, requisitos e restrições referentes a definição da Missão, influenciam diretamente a especificação dos satélites envolvidos.

A especificação de satélites artificiais é um processo complexo e árduo, na qual vários fatores influenciam diretamente, segundo Souza (2011b), estes fatores são: Operações, Cobertura, Dimensão e Massa, Potência Requerida, Sistema de Comunicação, Apontamento, Número de Satélites, Altitude e Taxa de Transferência de Dados. A tabela 3.1 sumariza estes fatores.

Tabela 3.1 - Associações e restrições de fatores.

Fatores	Associado à	Limitado por
Operações	missão	disponibilidade de comunicação, pessoal e custo
Cobertura	missão	própria órbita e campo de visada de carga útil
Dimensão e Massa	carga útil, necessidades de combustível e potência e ao lançador	
Potência	carga útil	peso e dimensão
Sistemas de Comunicação	carga útil e à missão	custo e massa
Apontamento	carga útil e à missão	custo e massa
Número de satélites	missão	custo
Altitude	carga útil e à missão	lançador e carga útil
Taxa de transmissão de dados	missão	capacidade de processamento, armazenamento e transmissão de dados a bordo
Cronograma das passagens	missão	

A próxima figura 3.4 mostra como é estabelecida a hierarquia dos requisitos, as setas bidirecionais indicam a natureza iterativa da Engenharia de Sistemas, assim como também representa os *feedbacks*.



Figura 3.4 - Objetivos e Requisitos de uma missão de satélite.
 Fonte: Adaptado de Fortescue e Stark (2001, p. 6)

Um satélite é dividido em duas partes: Carga Útil e Plataforma. A Carga útil é responsável pelo cumprimento dos objetivos da missão espacial, e a plataforma deve manter a carga útil, de modo que ela possa executar os seus objetivos. A plataforma é tradicionalmente decomposta em subsistemas. Estes subsistemas são Controle de Atitude, Suprimento e Energia, Telecomunicações, Gestão de Bordo, Estruturas e Mecanismos, Controle Térmico e Propulsão. O subsistema responsável pela disposição dos equipamentos que serão embarcados no satélite é o subsistema de Estruturas.

3.2.4 Subsistema de Estruturas - Configuração e Acondicionamento

Uma das responsabilidades do Subsistema de Estruturas é a Configuração e Acondicionamento dos Equipamentos à bordo do satélite. Devido a vários fatores, este trabalho torna-se extremamente complexo e envolve múltiplas disciplinas pela dependência com os outros subsistemas.

A tabela 3.5, mostra as origens dos requisitos de estrutura durante as fases da Missão. Além do problema do *layout* citado anteriormente, ainda existem outros fatores que influenciam na configuração e acondicionamento dos equipamentos, como a necessidade de acesso aos equipamentos durante os testes. De acordo com Fortescue e Stark (2001), uma parte inerente ao projeto de estrutura é a verificação do projeto final.

Mission Phase	Source of Requirements
Manufacture and assembly	<ul style="list-style-type: none"> • Handling fixture or container reactions • Stresses induced by manufacturing processes (welding)
Transport and handling	<ul style="list-style-type: none"> • Crane or dolly reactions • Land, sea, or air transport environments
Testing	<ul style="list-style-type: none"> • Environments from vibration or acoustic tests • Test fixture reaction loads
Prelaunch	<ul style="list-style-type: none"> • Handling during stacking sequence and pre-flight checks
Launch and ascent	<ul style="list-style-type: none"> • Steady-state booster accelerations • Vibro-acoustic noise during launch & transonic phase • Propulsion system engine vibrations • Transient loads during booster ignition & burn-out, stage separations, vehicle maneuvers, propellant slosh, and payload fairing separation • Pyrotechnic shock from separation events
Mission operations	<ul style="list-style-type: none"> • Steady-state thruster accelerations • Transient loads during pointing maneuvers and attitude control burns or docking events • Pyrotechnic shock from separation events, deployments • Thermal environments
Reentry & landing (if applicable)	<ul style="list-style-type: none"> • Aerodynamic heating • Transient wind and landing loads

Figura 3.5 - Origem de Requisitos para o subsistema de Estrutura.
 Fonte: Adaptado de Larson e Wertz (1999, p. 432)

A alocação dos requisitos inicialmente é feita através de uma abordagem *top-down*, na qual estimam-se as dimensões e a estrutura do satélite, que serão refinadas nas iterações posteriores. Por outro lado, uma abordagem *bottom-up*, permite que se façam soluções de compromisso (*trade-offs*) entre as alternativas e a realocação dos requisitos para viabilizar um o projeto. Portanto ambas as abordagens devem ser usadas para o desenvolvimento de satélites (LARSON; WERTZ, 1999). Neste trabalho foram usadas as duas abordagens para a alocação dos requisitos.

3.2.5 O problema do *Layout* de Equipamentos Embarcados em Satélites

Os equipamentos que fazem parte do *layout* desta dissertação são equipamentos pré-selecionados, portanto não há variação no número ou nas especificação dos equipamentos usados na otimização do *layout*.

Durante a concepção da arquitetura mecânica de satélites, a disposição dos equipamentos interfere diretamente na estrutura, desempenho, vida útil, custo de montagem e integração de todo o sistema, influenciando diretamente no desempenho global do satélite. Para isso, alguns requisitos (figura 3.5) devem ser atendidos, como resistência estrutural, balanço de massa, dissipação térmica dos equipamentos eletrônicos, interferências eletromagnéticas, entre outros (CUCO, 2011).

A otimização do posicionamento dos equipamentos é chamada de Problema de Projeto de Layout de Módulo de Satélites ou SMLDP, sigla em inglês para *Satellite Module Layout Design Problem* (WANG et al., 2009a). Várias abordagens para a resolução desse problema foram propostas, além de, pesquisas utilizando como apoio algoritmos computacionais. Tradicionalmente, a disposição dos equipamentos é feita de forma “manual”, e outros requisitos com base na experiência dos engenheiros envolvidos na definição da arquitetura mecânica e elétrica.

Como o problema envolve várias engenharias específicas, sua resolução pode ser proposta usando diferentes ferramentas computacionais, tais como ferramentas CASE ou produzidas *in-house*, as quais devem comunicar entre si.

A figura 3.6 mostra um exemplo de posicionamento de equipamentos.

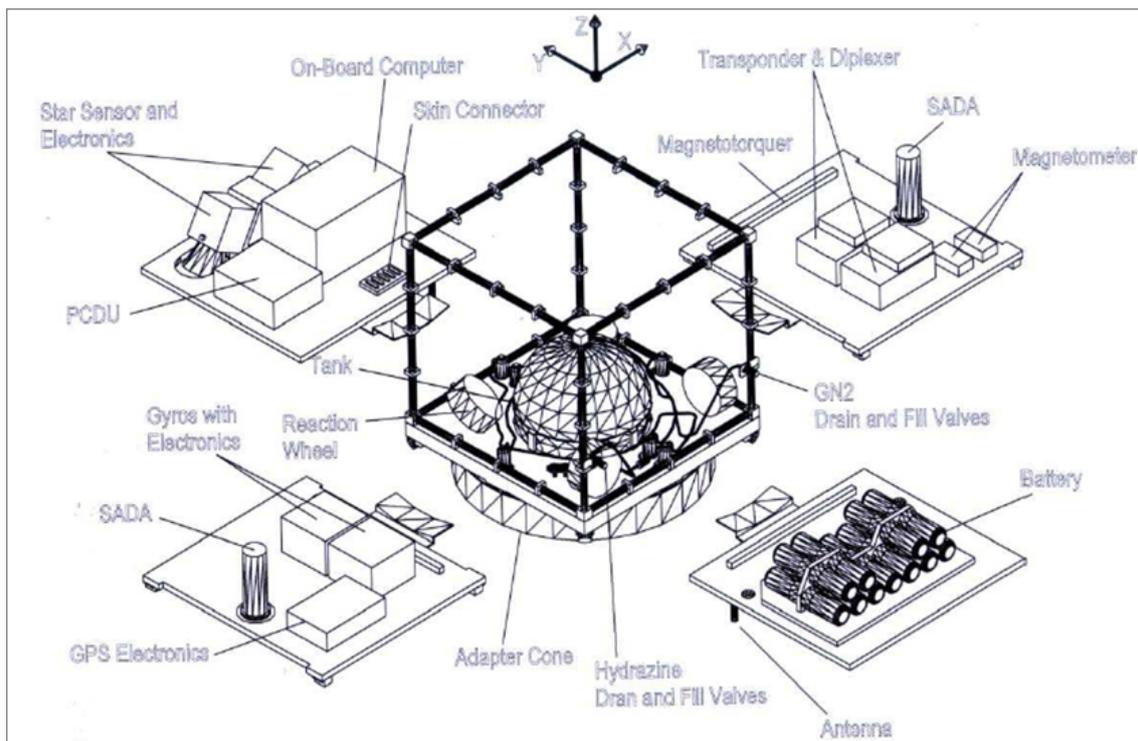


Figura 3.6 - *Layout* dos equipamentos da PMM.

Fonte: Modificado de Instituto Nacional de Pesquisas Espaciais (INPE) (2001a, p. 137).

De acordo com Wang et al. (2009b), a otimização de *layout* é uma das técnicas usadas para se melhorar o desempenho do satélite como um todo. Alguns fatores são fundamentais na decisão dos projetistas sobre o posicionamento dos equipamentos a bordo de um satélite. Para atendê-los da melhor maneira possível são feitas as soluções de compromisso (*trade-offs*) com os outros subsistemas do satélite e elementos da Missão.

Um dos fatores importantes é garantir o alinhamento dos eixos principais de inércia do sistema corrente em relação aos eixos de referência. A estabilidade estática também deve ser garantida minimizando a diferença entre o centro de gravidade do sistema corrente e o centro de gravidade definido como requisito pela equipe de projetistas.

A uniformidade de dissipação térmica dos equipamentos sobre um módulo evita regiões com alta densidade de fluxo térmico, que pode danificar equipamentos sensíveis ou requerer um controle térmico ativo. Isto adiciona massa ao satélite e consome po-

tência, um recurso crítico em muitas missões (CUCO, 2011). Cuco (2011) considera a dissipação térmica como uma das funções objetivo para o *layout* de equipamentos.

Adicionalmente, não deve haver sobreposição de equipamentos ou mesmo entre qualquer equipamento e o módulo do satélite, bem como todos os componentes devem estar contidos no módulo. E também é desejável que alguns equipamentos, não necessariamente pertencentes ao mesmo subsistema, estejam próximos uns dos outros.

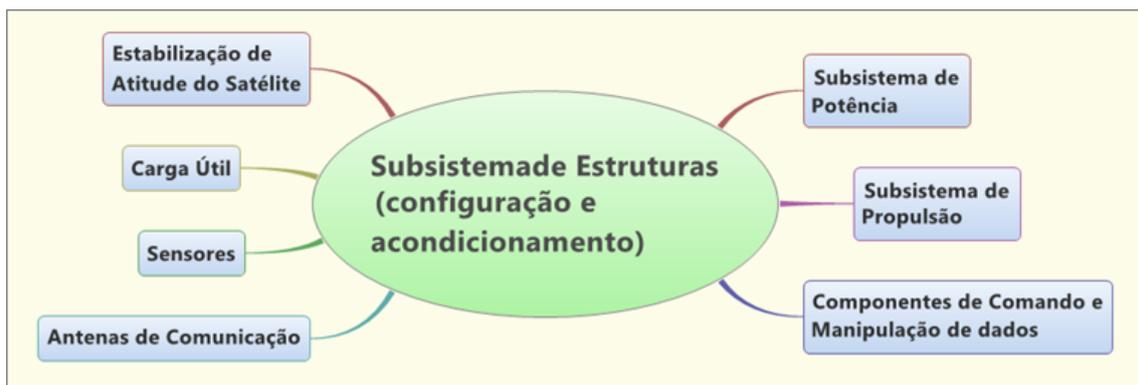


Figura 3.7 - Alguns fatores e elementos que influenciam do projeto de estruturas e do *layout* de equipamentos em satélites

3.2.6 Alguns Fatores que Influenciam o *Layout*.

São mostrados na figura 3.7 os elementos de alto nível e os fatores que influenciam o subsistema de estruturas, durante a configuração e o acondicionamento dos equipamentos. A carga útil e o tipo de estabilização são basicamente, os *drivers* da configuração e o acondicionamento.

Em seguida é feito um detalhamento dos requisitos referentes a cada um dos fatores e o rastreamento até os parâmetros (massa, dimensões e posição) que influenciam no posicionamento dos equipamentos que serão embarcados no satélite. O rastreamento aqui detalhado é somente ilustrativo, pois estes fatores são influenciados entre si e influenciam outros subsistemas interdependentes com o subsistema de estruturas.

O sistema de lançamento é formado pelo veículo lançador, pelas cargas que serão lançadas, pelo estágio superior do satélite, coifa e equipamentos, bem como instalações de apoio em solo. O veículo lançador restringe o peso, a geometria, a rigidez e a resistência do satélite. Estas restrições são propagadas para a configuração e

acondicionamento dentro do subsistema de estruturas, afetando a massa, o tamanho e o posicionamento dos equipamentos.

A rigidez exigida pelo lançador influencia a deflexão máxima que por sua vez, influencia diretamente o subsistema de estruturas, através dos materiais que serão escolhidos para manufatura. Além disso, as frequências naturais do lançador influenciam no volume e na massa do satélite. O veículo lançador, o estágio superior e a órbita estabelecem o peso máximo permitido para o satélite. A Figura 3.8 mostra as restrições impostas pelo sistema de lançamento na configuração e acondicionamento do subsistema de estruturas.

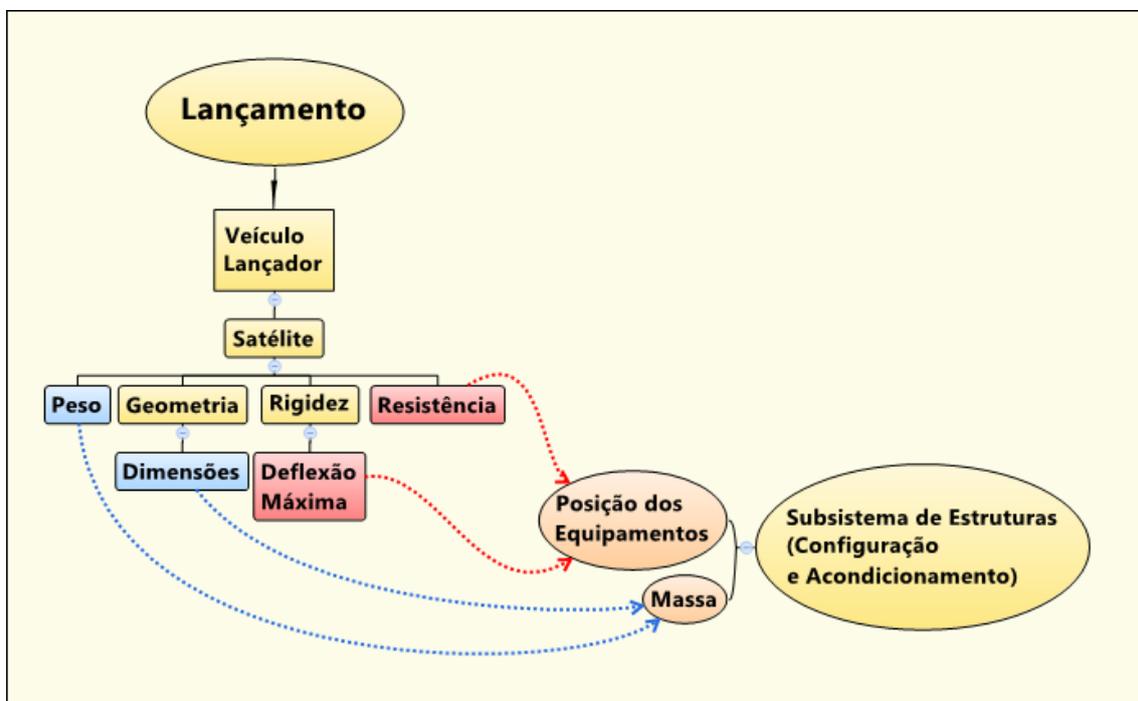


Figura 3.8 - Influência do Sistema de Lançamento no *layout*.

A partir da definição da missão é projetado o subsistema de controle de atitude que terá grande influência no *layout* da estrutura e no *layout* dos equipamentos. Por exemplo, na estabilização por spin, o momento de inércia do eixo de giro deve ser maior que o dos outros eixos para manter a estabilidade, exigindo mais combustível para a reorientação e, conseqüentemente, aumento de massa. Já na estabilização por três eixos deve haver separação do magnetômetro com as rodas de reação, para evitar interferência magnética, causando restrição de posicionamento. Estes fatores

são sumarizados na Figura 3.9

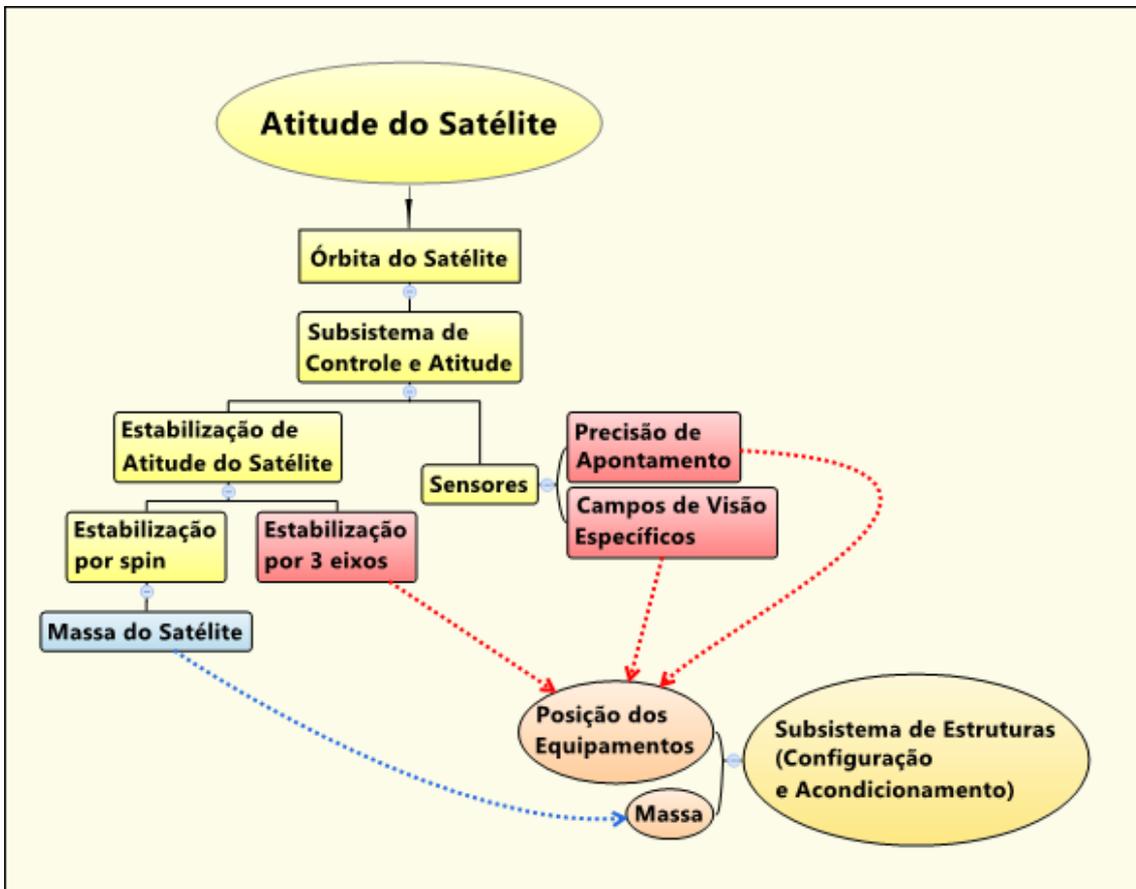


Figura 3.9 - Influência da escolha de órbita no *layout*.

O ambiente espacial é a grande fonte de restrições para o desenvolvimento de satélites. A Figura 3.10 ilustra os fatores do ambiente espacial que afetam o *layout*. Por exemplo, ele afeta o subsistema de estruturas devido à exposição dos equipamentos mais sensíveis à radiação que demandam proteções físicas alterando a massa e o posicionamento.

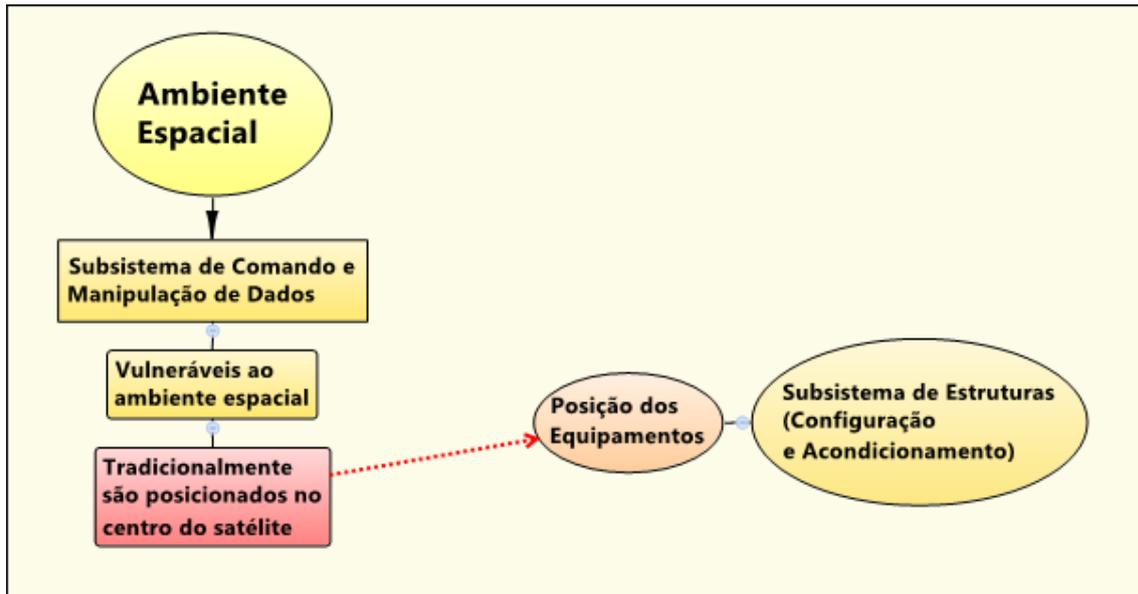


Figura 3.10 - Influência da radiação espacial no *layout* de equipamentos .

A carga útil de um satélite é um fator de extrema importância, pois está fortemente conectada à missão. Dela derivam várias restrições que devem ser atendidas e usadas durante as soluções de compromisso (*trade-offs*) que acontecem na fase de caracterização da arquitetura de Missão conforme mostrado na Figura 3.11.

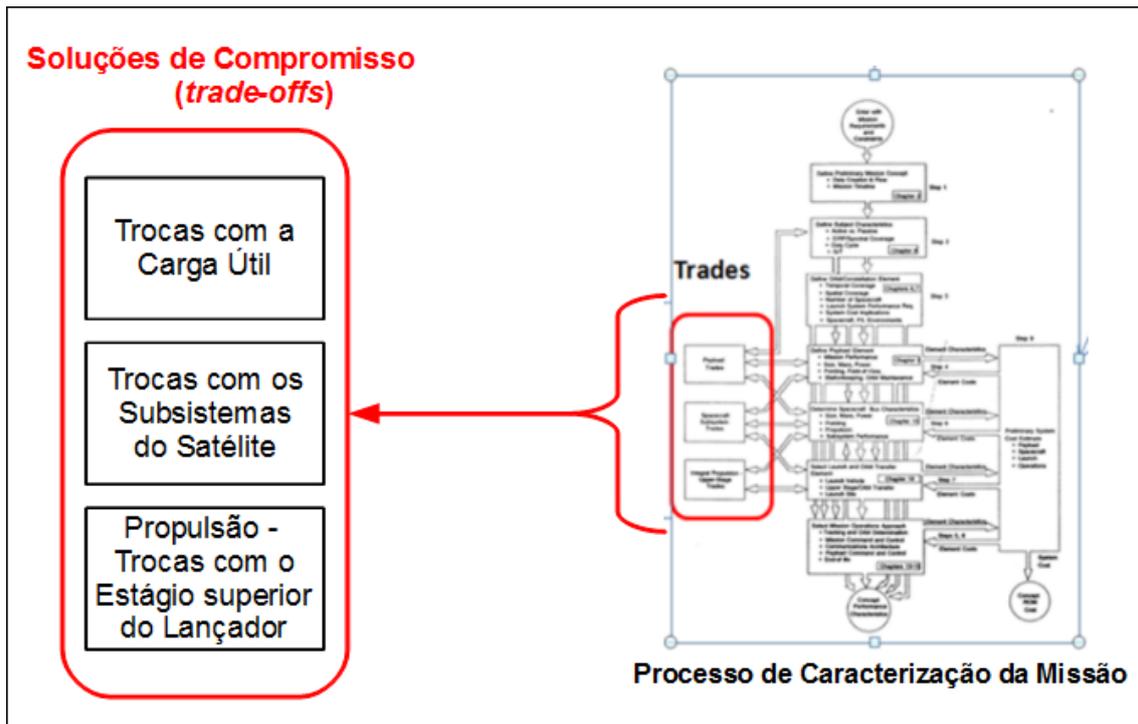


Figura 3.11 - As soluções de compromisso (*trade-offs*) durante a caracterização da missão

Todos os subsistemas do satélite influenciam uns aos outros, aumentando ainda mais a complexidade da definição de um *layout* adequado e evidenciando o fluxo de informações envolvidas nas soluções de compromisso (*trade-offs*). A Figura 3.12 mostra o impacto desses outros subsistemas no subsistema de estruturas, com foco na configuração/acondicionamento.



Figura 3.12 - Exemplo de acoplamento do *layout* do subsistema de estruturas com os outros subsistemas.

A dependência do subsistema de estruturas com outros subsistemas é ressaltada pois todos os subsistemas são altamente dependentes uns dos outros.

Um exemplo disto são:

- As antenas de comunicação necessitam de um campo de visão, não podem ser obstruídos por outros equipamentos, além das restrições de rigidez e estabilidade térmica, restringindo o posicionamento e a massa. Dependendo da precisão do apontamento das antenas se a abertura de visada da antena aumenta, o seu ganho também aumenta diminuindo a potência, diminuindo a massa.
- Os componentes de comando e manipulação de dados são geralmente sensíveis ao ambiente espacial, e são tradicionalmente posicionados na posição central para protegê-los da radiação.
- Os componentes dos subsistemas de propulsão são usualmente localiza-

dos na periferia, e podem contaminar outros equipamentos com gases do propelente. Os componentes do Subsistema de Propulsão também influenciam no posicionamento dos equipamentos, pois podem contaminar outros equipamentos com os gases resultantes do propelente. E é claro também influencia na massa e dimensões dependendo do tipo de propelente e tamanho do tanque.

3.3 O uso da linguagem SysML

A SysML é uma linguagem padronizada e criada especificamente para a Engenharia de Sistemas, e vários estudos estão sendo feitos sobre a sua utilização. Neste trabalho a escolha da SysML foi feita pelo fato de ser uma linguagem padronizada, e pela disponibilidade da ferramenta Rhapsody. O uso da SysML possibilita a modelagem do sistema através de diagramas e permite expressar estrutura e comportamento do sistemas e seus subsistemas, apesar de não haver especificações na SysML sobre simulação ela fornece um padrão para modelos, e modelos podem ser simulados. A Ferramenta Rhapsody permite a simulação dos diagramas dinâmicos, ou seja, que expressam comportamento. Neste trabalho não foi modelado o comportamento do sistema, pelo esforço que este trabalho traria e também porque não é diretamente necessário para este protótipo, mas é importante ressaltar esta possibilidade da SysML e da ferramenta Rhapsody para possíveis versões futuras do protótipo.

A proposta é a modelagem da estrutura através de diagramas de blocos e diagrama de blocos internos, para melhor entendimento do sistema e facilidade de comunicação entre os envolvidos, além de fornecer mecanismo de iterações que refinem os requisitos. Também é proposto que sejam feitos os diagramas de requisitos, usando os elementos SysML do tipo "Requisito"(objeto "bloco"com esteriótipo do tipo "requisito") importados do documento de requisito inicial, desta forma é possível ver de forma gráfica os requisitos e ligá-los com outros elementos SysML, como neste caso os parâmetros que serão usados na otimização.

Nesta dissertação usamos a linguagem SysML para a construção de modelos para uso na Engenharia de Sistemas e o foco é a interoperabilidade de ferramentas de software. O problema de interoperabilidade entre modelos não foi tratado aqui.

3.4 O uso proposto da MDO

Na fase conceitual de desenvolvimento de satélites são tomadas decisões baseadas não somente nos requisitos da mais alta hierarquia, mas também de aspectos que

envolvem parâmetros específicos dos subsistemas, como o problema do *layout* dos equipamentos embarcados. A partir dos requisitos e restrições do projeto, os projetistas analisam e escolhem o *layout* mais adequado, baseados em cálculos específicos e em suas experiências profissionais. O uso da MDO proposto neste trabalho tem a finalidade de dar apoio aos projetistas na fase conceitual, fornecendo várias opções soluções otimizadas de *layout* que possam ser úteis durante as tomadas de decisão envolvidas nesta fase.

A solução de MDO usada nesta dissertação é composta de vários softwares interoperantes, mas é possível agregar vários problemas de MDO em uma estrutura de software única, dependendo da solução do processo adotado.

4 IMPLEMENTAÇÃO DA SOLUÇÃO MBSE - MDO

Neste capítulo é apresentada uma infraestrutura de software desenvolvida com o objetivo de proporcionar um ambiente multidisciplinar para exercitar a MBSE e a MDO no caso específico da otimização de *layout*. Para a criação foi necessário considerar as ferramentas envolvidas, estudar suas configurações, seus usos básicos e códigos-fonte, quando disponíveis, além de soluções para a integração e interoperabilidade de dados entre estas ferramentas.

Atualmente as aplicações de software vão além de oferecer funcionalidades de computação e armazenamento de dados, elas devem fornecer meios para serem estendidas, integradas e alteradas facilmente. Dentro dessa nova realidade, as arquiteturas de software têm se tornado cada vez mais complexas e novas tecnologias surgem a cada dia.

Nota-se que uma ferramenta única jamais terá todas as funcionalidades desejadas e que o reuso de ferramentas e códigos é a forma mais adequada para qualquer infraestrutura de software, porém existem vários problemas que surgem, como questão de mudanças de versões e de sistemas operacionais diferentes que devem ser levados em questão.

Outra questão que foi levantada para a escolha da implementação foi a das interfaces ponto a ponto (melhores, mas inflexíveis) verso as interfaces genéricas/padronizadas (piores mas flexíveis). Nesta dissertação, após o estudo feito na introdução teórica, optou-se pelo uso das duas, a ponto a ponto para comunicação entre o SatBudgets e o PJLayout e as genéricas através dos padrões XML e da integração das ferramentas da IBM Rational.

4.1 Seleção de Ferramentas Comerciais para Engenharia de Sistemas

A solução proposta envolve aplicativos distintos, desenvolvidos de formas distintas, com linguagens diferentes e funcionalidades complementares, bem como ferramentas CASE de fabricantes diferentes. Para isto, foram usadas abordagens diferentes para resolver o problema de interoperabilidade e fluir as informações dentro deste ambiente criado.

4.1.1 A ferramenta IBM® Rational® DOORS®

A ferramenta DOORS ¹ (IBM (2014c)) é uma ferramenta CASE usada para o gerenciamento de requisitos. Ela possui capacidade para gerenciar grandes volumes de requisitos. Uma de suas funcionalidades é a capacidade de organização de requisitos, por projetos, pastas e módulos formais. Além disso esta ferramenta possui capacidade de importação de requisitos de documentos ".doc" e geração automática de relatórios em diversas extensões. Porém a funcionalidade mais importante é a rastreabilidade dos requisitos, que podem estar conectados em hierarquias ou por outros tipos de ligação. Esta rastreabilidade pode ser visualizada através da geração automática de tabelas de rastreabilidade.

4.1.2 A ferramenta IBM® Rational® Rhapsody®

A ferramenta Rhapsody ² (IBM (2014a)) é uma ferramenta de modelagem UML e SysML, ela possui boa usabilidade para a construção de diagramas SysML, por apresentar uma biblioteca dos objetos e símbolos básicos da SysML e também possibilitar a criação de esteriótipos customizados. Esta ferramenta também possui verificação dos objetos, símbolos e ligações de cada diagrama, proibindo que objetos pertencentes a diagramas não compatíveis sejam adicionados a um determinado diagrama. Neste trabalho esta ferramenta será referenciada somente como: Rhapsody. Além de diagramação, apresenta outras funcionalidades, como a possibilidade de simulação, através dos diagramas de comportamento, geração de relatórios e conexão com a ferramenta MathWorks MATLAB e Simulink.

4.1.3 A ferramenta IBM® Rational® Rhapsody® Gateway®

A ferramenta Gateway ³ é classificada como um complemento (*plugin*) da ferramenta Rhapsody, apesar de possuir sua própria interface ela somente pode ser usada e executada a partir do Rhapsody. Esta ferramenta é usada para a ligação dos objetos da ferramenta Rhapsody com outras ferramentas (IBM, 2014b).

Neste trabalho somente foi explorado sua integração com o DOORS. Esta integração permite que objetos das duas ferramentas sejam exportados entre elas, evitando retrabalho e, o mais importante, garante a rastreabilidade dos modelos SysML aos requisitos. Ela permite que sejam visualizados quais requisitos estão "cobertos", ou seja, estão ligados com objetos SysML e indica quando algum deles é modificado,

¹Neste trabalho esta ferramenta será referenciada somente como: DOORS.

²Neste trabalho esta ferramenta será referenciada somente como: Rhapsody.

³Neste trabalho esta ferramenta será referenciada somente como: Gateway.

tanto na ferramenta DOORS, como na Gateway. Desta forma a rastreabilidade é sempre atualizada na interface do Gateway.

4.2 As Ferramentas Desenvolvidas no INPE

Com base nos conceitos de reusabilidade, foram usadas duas ferramentas já desenvolvidas internamente pelo INPE e com acesso ao código: o SatBudgets (Leonor (2010)) e o PJLayout (de Sousa et al. (2013)), que serão apresentadas a seguir. Estas duas ferramentas possuem, cada uma delas, uma solução de interoperabilidade com ferramentas CASE.

4.2.1 O Aplicativo Satbudgets

O SatBudgets (citeonlineleonor/10) é uma ferramenta protótipo, desenvolvida para cálculos de balanços mecânico, elétrico e de potência para uso durante a fase conceitual em projetos de satélites. Seu funcionamento consiste na leitura de um modelo SysML (diagrama de blocos), no qual cada bloco representa um equipamento da arquitetura do satélite, usando a ferramenta TopCased.

É possível selecionar as opções de quais balanços serão computados e qual o formato do relatório gerado (PDF, Visualizar na tela) na sua interface. A aplicação calcula os balanços para o modelo selecionado e os exibe em forma de relatório (LEONOR, 2010). O fluxo completo do Satbudgets é mostrado na Figura 4.1.

O SatBudgets foi desenvolvido usando a linguagem orientada a objetos Java, do tipo *standalone* (instalada no computador do usuário) usando a IDE Eclipse, as bibliotecas Java: JasperReports (geração de relatório), JFreeChart (geração de gráficos), DROOLS (processamento de regras) e JDOM (manipulação de XML)(LEONOR, 2010).

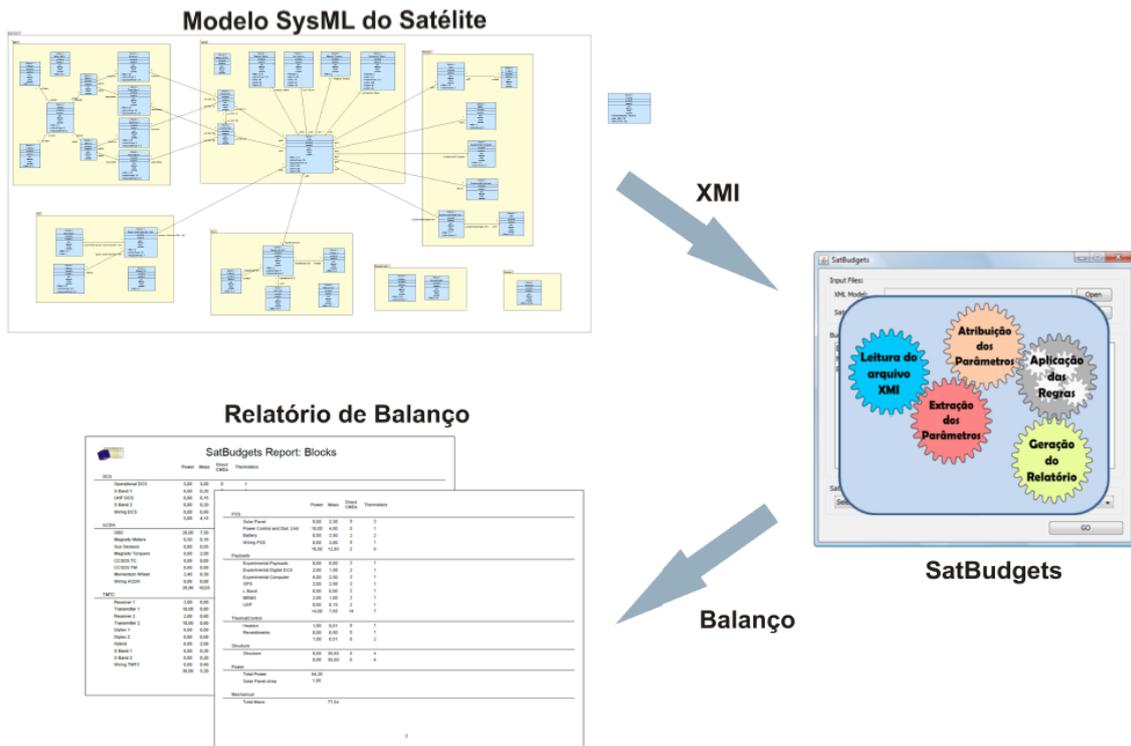


Figura 4.1 - Fluxo de Trabalho do SatBudgets.
 Fonte: Adaptado de Leonor (2010, p. 40)

4.2.2 O Aplicativo PJLayout e o Algoritmo MGEO

O aplicativo PJLayout (de Sousa et al. (2013)) foi desenvolvido com o objetivo de implementar o algoritmo de otimização evolutivo MGEO (Galski (2006)) para a otimização de *layout* de equipamentos embarcados em satélites. O algoritmo MGEO é codificado usando a linguagem de programação Visual Basic e executado a partir da interface da ferramenta Excel, no qual também são apresentados os resultados. Ele também usa a API do Solidworks (DS (2014)) para cálculos de sobreposição de equipamentos, cálculos de centro de massa, momentos principais de inércia e para visualização gráfica dos resultados. O fluxo de trabalho do PJLayout é apresentado na Figura 4.2.

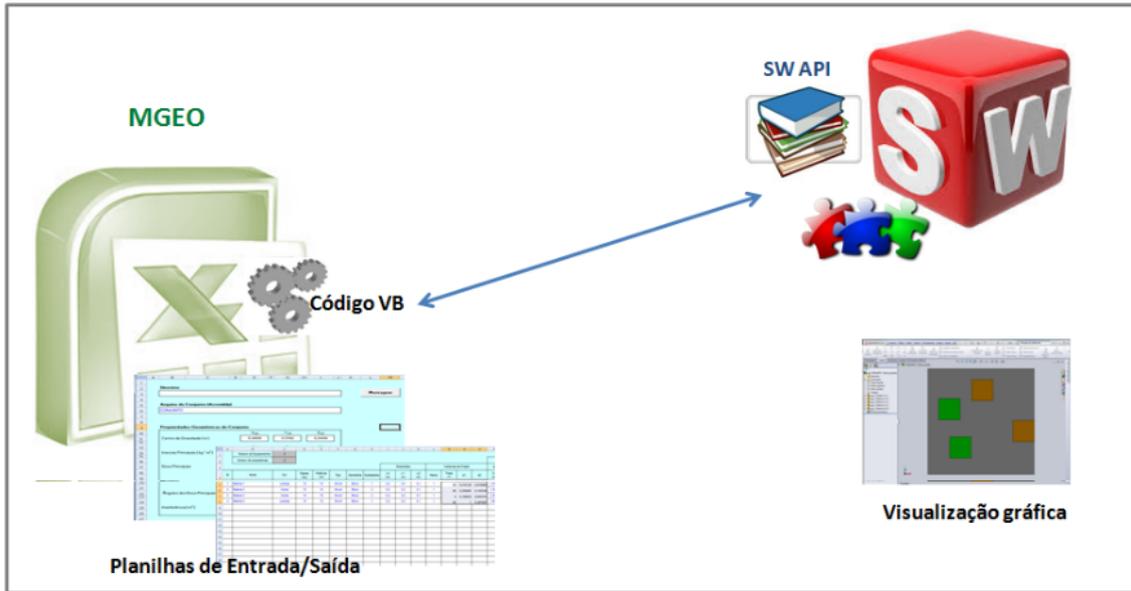


Figura 4.2 - Fluxo de Trabalho do PJLayout.
 Fonte: Adaptado de de Sousa et al. (2013)

O aplicativo PJLayout usa três funções objetivo apresentadas a seguir.

$$f_1 = \sqrt{\sum_{i=1}^3 (x_{i_CM_sis} - x_{i_CM_alvo})^2} \quad (4.1)$$

f_1 representa o objetivo de que o centro de massa do sistema esteja o mais próximo possível da posição alvo desejada. $x_{i_CM_sis}$ e $x_{i_CM_alvo}$ são, respectivamente, as coordenadas Cartesianas do centro de massa do sistema e alvo.

$$f_2 = \frac{\sum_{k=1}^{N_c} \left(\sum_{i=1}^{N_{equip}} \frac{P_i}{r_{i,k}^2} - \frac{\sum_{j=1}^{N_c} \sum_{i=1}^{N_{equip}} \frac{P_i}{r_{i,j}^2}}{N_c} \right)^2}{N_c} \quad (4.2)$$

f_2 representa o objetivo de que o calor dissipado pelos equipamentos seja distribuído de maneira uniforme sobre o painel. O painel é dividido em N_c células com largura igual a metade da menor dimensão lateral encontrada nos N_{equip} equipamentos. P_i representa o calor dissipado pelo equipamento i . $r_{i,j}^2$ ou $r_{i,k}^2$ é a distância Euclidiana, no plano do painel, entre o centro geométrico de um equipamento i e a célula k .

Na Figura 4.3 é apresentado um exemplo destas distâncias para um equipamento posicionado sobre um painel com 16 células.

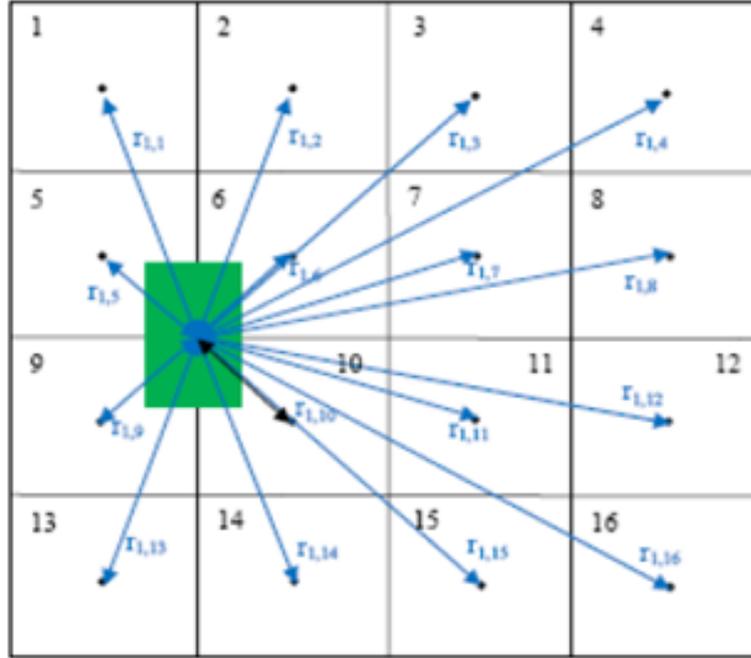


Figura 4.3 - Exemplo das distâncias para um equipamento posicionado sobre o painel.
Fonte: de Sousa et al. (2013)

$$f_3 = \sum_{k=1}^{N_{sub}} \sum_{i=1}^{N_{equip,k}-1} \sum_{j=i+1}^{N_{equip,k}} d_{i,j,k} \quad (4.3)$$

f_3 representa o objetivo de que equipamentos de um mesmo subconjunto fiquem próximos entre si. $d_{i,j,k}$ representa a distância Euclidiana entre os centros geométricos de dois equipamentos i, j do subsistema k .

Estas funções estão sujeitas a:

$$0 \leq s_{i,1}, s_{i,2} \leq 1$$

e

$$\theta_{min} \leq \theta_i \leq \theta_{max}$$

$s_{i,1}$, $s_{i,2}$, e θ_i são as variáveis de projeto. Respectivamente, as duas primeiras definem a posição de um equipamento i sobre o painel, parametrizada pelas dimensões do mesmo e do equipamento. A terceira define a rotação do equipamento sobre o painel, conforme definido na Figura 4.4.

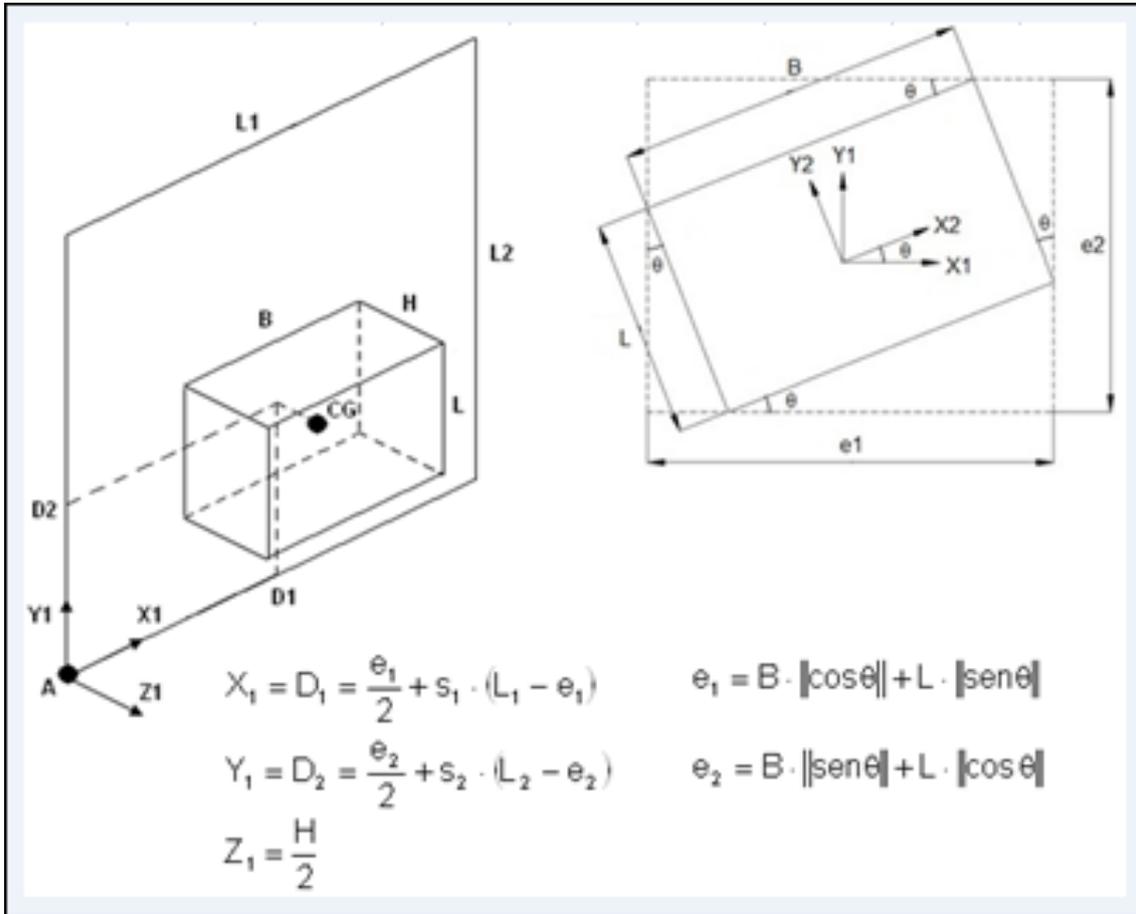


Figura 4.4 - Exemplo de rotação do equipamento sobre o painel.
 Fonte: de Sousa et al. (2013)

4.3 Infraestrutura Prototipada de Software e Tecnologias Usadas

A ferramenta SatBudgets foi escolhida com o propósito da reusabilidade de funcionalidades mediante a sua extensão e para exemplificar a interoperabilidade entre diferentes ferramentas usando o padrão XML. Esta ferramenta também faz uso de modelos através da linguagem SysML e foi modificada de uma versão *standalone*

para uma Web⁴ implantada dentro de um servidor Web.

Em paralelo o aplicativo PJLayout foi escolhido por já ter implementado o algoritmo de otimização, as funções-objetivo já modeladas matematicamente e interoperar com o SolidWorks que não possui API para Java atualmente.

As interfaces necessárias para aplicação e o código necessário para a interoperabilidade com o código do PJLayout foram codificados dentro do Satbudgets, além da leitura de modelos criados no Rhapsody e a configuração do Gateway para a integração DOORS/Rhapsody.

Uma representação simplificada do processo, através das interfaces das ferramentas de software utilizadas é mostrada na figura 4.5.

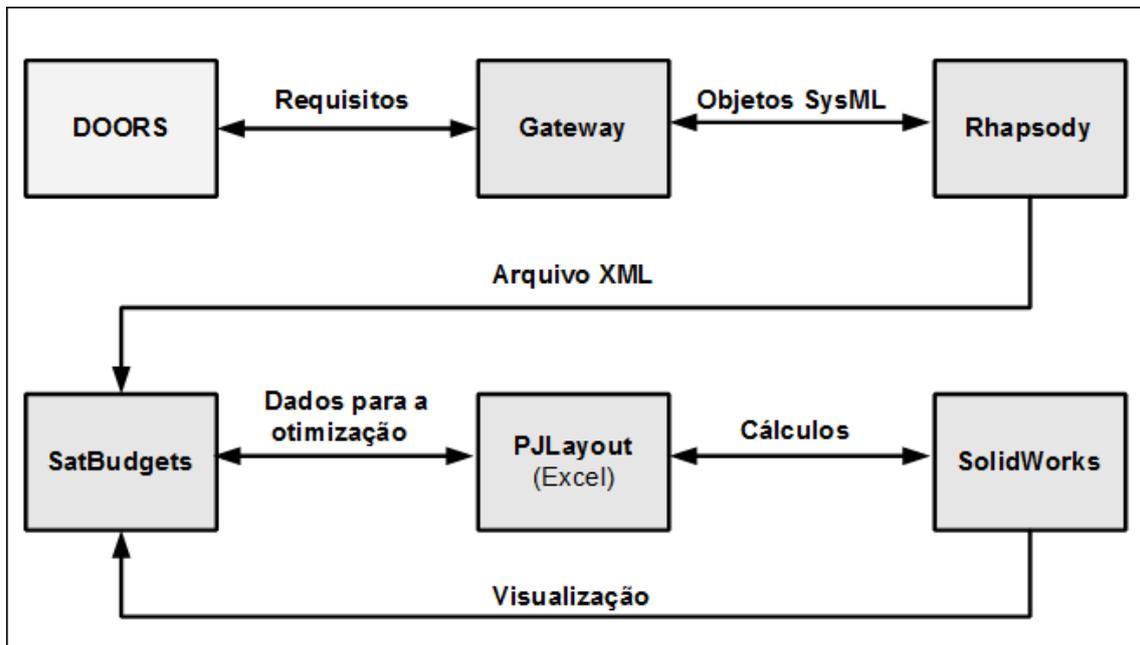


Figura 4.5 - Processo simplificado usando as interfaces das ferramentas de software.

Esta estrutura criada foi denominada de Optima e o seu fluxo de trabalho dentro do ambiente de MBSE-MDO é mostrado na Figura 4.6 e detalhado no Apêndice A.

⁴aplicativo que é instalado no servidor e pode ser acessado a partir de outras máquinas através de um navegador.

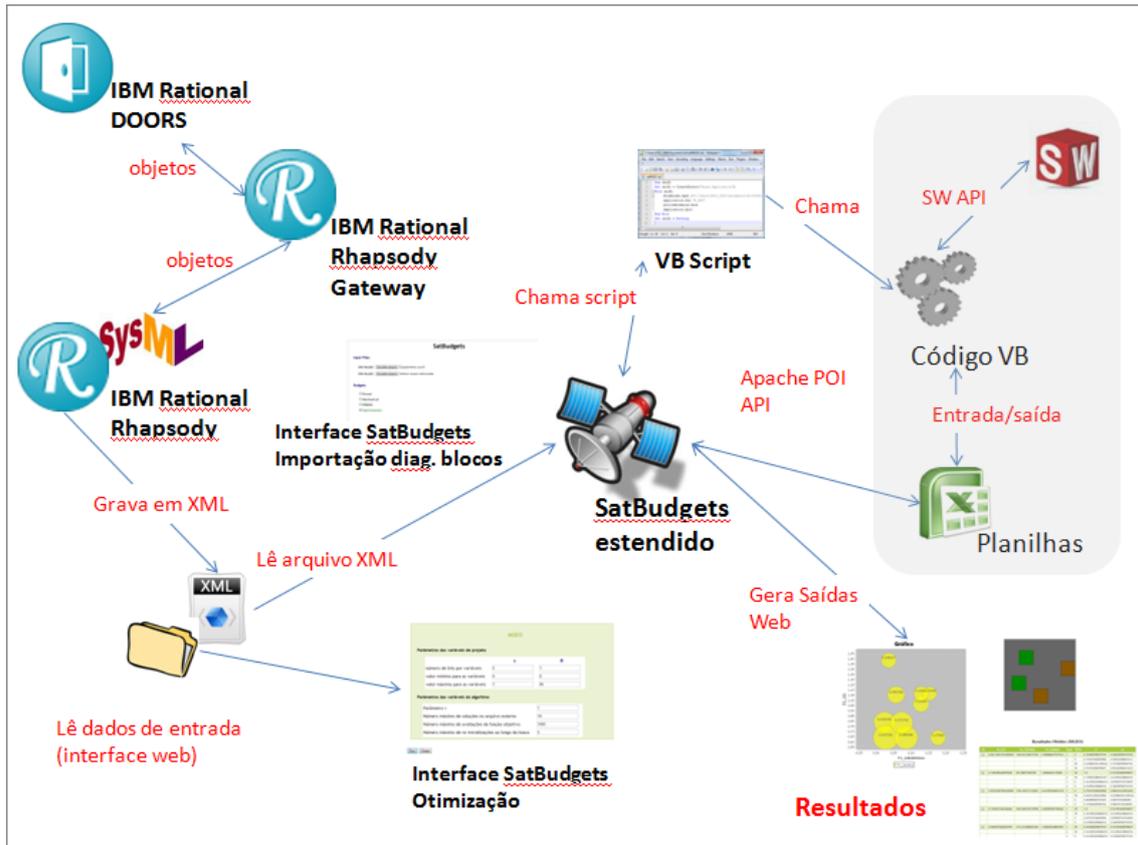


Figura 4.6 - Fluxo de Trabalho do Optima.

O software DOORS é responsável pelo gerenciamento dos requisitos do sistema e o software Rhapsody permite a diagramação da linguagem SysML. Estas duas ferramentas são integradas através do plugin Gateway, que permite, entre outras funcionalidades, a troca de dados entre as duas ferramentas. Além de garantir a rastreabilidade entre requisitos, e requisitos e elementos SysML proporciona que o usuário destas ferramentas não tenha o retrabalho de inserir as mesmas informações em duas aplicações diferentes, ganhando tempo e diminuindo a possibilidade de duplicidade de dados.

A partir dos requisitos cadastrados no DOORS, e da modelagem SysML é possível usar a ferramenta SatBudgets para executar o aplicativo de otimização. Para isto o usuário, através da interface Web do SatBudgets, precisa importar o arquivo xml gerado pelo Rhapsody que contenha as informações do diagrama de equipamentos e o painel que serão otimizados. Cada equipamento deve estar preenchido, em seus parâmetros, os valores necessários de entrada para o aplicativo PjLayout. Após serem preenchidos os parâmetros de algoritmo e de projeto, o usuário consegue

executar o aplicativo PJLayout e visualizar os resultados na interface do SatBudgets e pode selecionar qualquer uma das arquiteturas otimizadas na tela do SolidWorks.

Apesar dos passos do fluxo de trabalho o usuário final somente visualiza as interfaces gráficas das ferramentas para a entrada de dados. A seguir a implementação da arquitetura será apresentada mais detalhadamente.

A extensão do Satbudgets foi feita usando a plataforma Java, dentro do ambiente Eclipse (Eclipse Foundation (2013)) e do servidor Web Tomcat (The Apache Software Foundation (2013)). Nesta tarefa foram também utilizadas as bibliotecas Java (API's): Apache POI (The Apache POI Project (2013)) e JFreeChart (JFree.org (2013)). A modelagem SysML foi feita usando a ferramenta Rhapsody, ferramenta gráfica que permite a modelagem SysML através da sua biblioteca interna de componentes onde é possível sua diagramação.

Para o desenvolvimento da arquitetura de MDO, foi também usada a IDE Eclipse, o servidor Web Apache Tomcat, é um servidor Web, de código aberto, que implementa as tecnologias: Java Servlet (Oracle (2013b)) e JavaServer Pages (Oracle (2013a)).

É importante ressaltar que além da API básica do Java, foram usadas as API's Apache POI e JFreeChart, que respectivamente, fornece bibliotecas Java para leitura e escrita em arquivos em formatos do Microsoft Office (exemplo Word, Power Point e Excel) e permite a criação de diagramas, como PieChart, SpiderChart, entre outros.

VBScript é uma linguagem de script desenvolvida pela Microsoft e tem como base o Visual Basic. Ela tem por objetivo ser uma linguagem de script para a Web, como por exemplo o JavaScript, porém é muito usada para escrever arquivos batch⁵ do sistema operacional Windows através do *Windows Scripting Host(WSH)*, desta forma o Windows trata o VBScript como um executável.

Para lidar com a interoperabilidade entre o aplicativo Rhapsody com o PJLayout, a aplicação Satbudgets, extrai dados do arquivo XMI gerado a partir dos modelos SysML e os transforma em objetos Java para serem manipulados e encaminhados para o PJLayout. Para a interoperabilidade com o PJLayout o Satbudgets usa um arquivo VBScript para executar as macros do PJLayout.

A extensão de uma aplicação *standalone* para uma versão Web, permite que o aplicativo seja usado em diversas máquinas através da rede, útil em equipes fisicamente

⁵Um batch é um arquivo que contém comandos que podem ser entendidos pelo interpretador, da mesma forma que na linha de comando. São usados para automatizar tarefas.

separadas. Como o foco aqui é uma infraestrutura de software MBSE-MDO, na qual são necessárias várias ferramentas diferentes, não foi possível que esta modificação do SatBugets para ser usado em qualquer máquina pelos motivos seguintes:

- A ferramenta Solidworks só roda em ambientes Windows, e precisa estar instalada na máquina que será usada.
- As ferramentas Rhapsody e DOORS, apesar de possuírem licenças flutuantes, também precisam estar instaladas na máquina do usuário.
- O complemento (*plugin*) Gateway deve ser instalado no aplicativo Rhapsody onde possibilita a integração entre a ferramenta Rhapsody com a ferramenta DOORS e possui funcionalidades para a interoperabilidade com outras ferramentas, como ex. MATLAB.
- Para que seja feita a integração é necessário que sejam feitas algumas configurações na interface do Gateway através da modelagem dos tipos de documentos/projetos envolvidos. Desta forma é possível manter sincronizados os documentos envolvidos e permitir a rastreabilidade entre seus elementos.

A fim de verificação do funcionamento da estrutura criada foi feito um estudo de caso, apresentado no próximo capítulo.

5 MODELAGEM DO ESTUDO DE CASO E RESULTADOS

Este capítulo aborda a modelagem e resultados do estudo de caso simplificado de um satélite real, a Plataforma Multi-Missão - PMM ([Instituto Nacional de Pesquisas Espaciais \(INPE\) \(2001b\)](#)), para exercitar o protótipo da infraestrutura de software criada desde os requisitos, os modelos até os resultados obtidos pela otimização.

Adicionalmente procura-se mostrar como os requisitos iniciais de um projeto são convertidos em parâmetros que podem ser usados na otimização do posicionamento dos equipamentos embarcados em satélites a importância de se manter a rastreabilidade destes elementos.

Para isso foram seguidos os seguintes passos:

- a) Identificação dos requisitos através do documento [Instituto Nacional de Pesquisas Espaciais \(INPE\) \(2001b\)](#);
- b) Uso da modelagem SysML;
- c) Ligação entre os elementos: requisitos e objetos SysML;
- d) Uso do modelo SysML criado, no software SatBudgets (que faz acoplamento com o PJLayout);
- e) Visualização dos resultados da otimização.

Também foi usada a estrutura de tópicos da Figura 5.1.

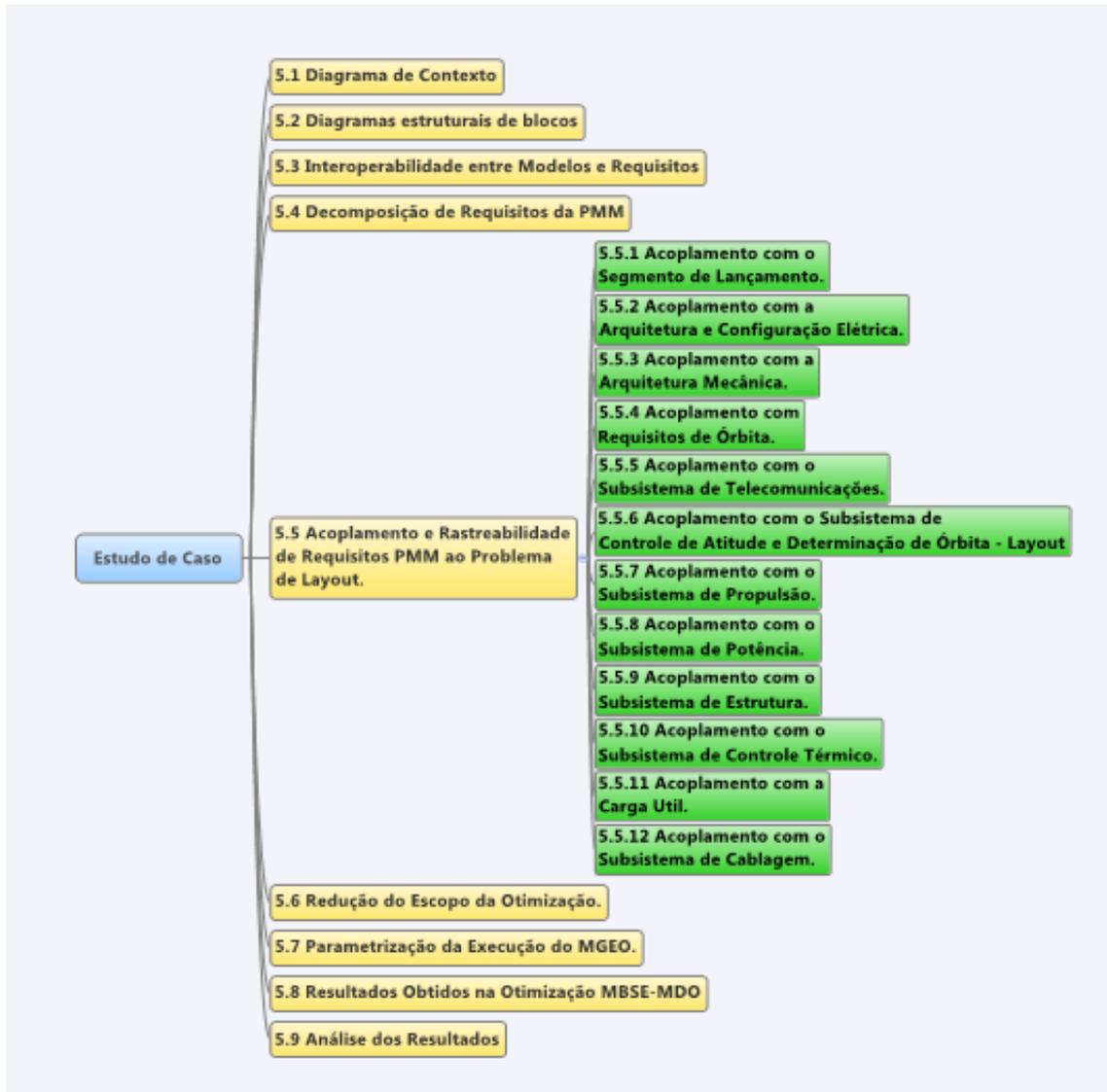


Figura 5.1 - Visão geral com estrutura dos tópicos deste capítulo.

Apresentam-se a seguir os diagramas desenvolvidos para a PMM.

5.1 Diagrama de Contexto

O diagrama de contexto é derivado inicialmente da arquitetura de missão da Figura 3.2 no capítulo 3 onde se identifica os principais aspectos de uma missão quais sejam (1) Segmento Lançador, (2) Segmento Espacial, (3) Órbita e Constelação, (4) Carga Útil, (5) Objeto da Missão, (6) Operações de Missão, (7) Segmento de Solo.

Para a PMM, o diagrama SysML de Contexto derivado é apresentado na Figura

5.2. Note que como não há “constelações” um dos blocos se resume a apenas órbita.

A linguagem de modelagem SysML não define um diagrama de contexto, nem define quais objetos usar, portanto na literatura podem ser encontradas diversas formas de representar o diagrama de contexto usando os elementos da SysML. Neste estudo de caso foi usado o diagrama de blocos com relacionamentos do tipo associação, porém uma alternativa seria usar o diagrama de Casos de Uso. A ferramenta Rhapsody usada para a diagramação não permite que sejam adicionados relacionamentos sem que sejam acrescentados os Casos de uso, dentro do sistema, influenciando a escolha do diagrama de blocos.

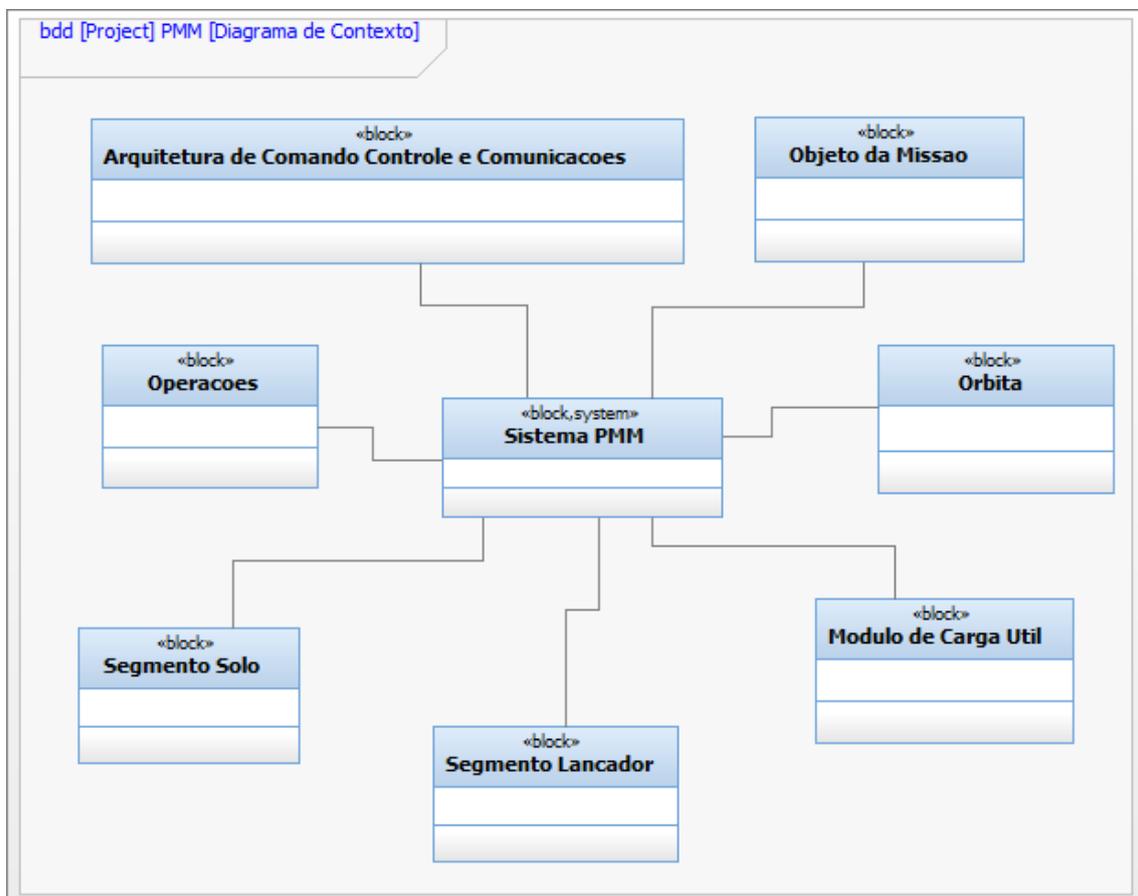


Figura 5.2 - Diagrama de Contexto da PMM

5.2 Diagramas estruturais de blocos

Após o diagrama de contexto foram elaborados os diagramas que mostram a estrutura de blocos internos à PMM com seus diversos subsistemas. O diagrama da figura 5.3 mostra o diagrama de blocos básico da PMM:

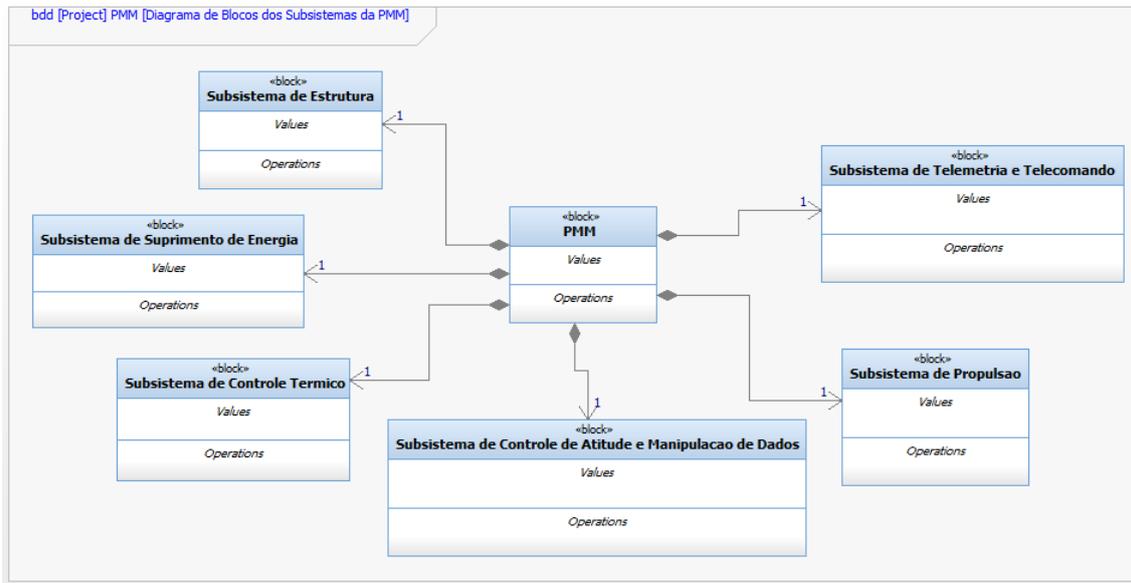


Figura 5.3 - Diagrama estrutural de blocos da PMM

O sistema PMM é representado através de um bloco, assim como todos os seus subsistemas (Subsistema de Estrutura; Subsistema de Suprimento de Energia; Subsistema de Controle Térmico; Subsistema de Controle de Atitude e Manipulação de Dados e Subsistema de Propulsão).

O próximo diagrama apresentado na figura 5.4 também modela estrutura, e mostra a hierarquia de cada subsistema da PMM. O subsistema é modelado como um bloco e representa o modelo genérico dos subsistemas da PMM com alguns parâmetros mas os valores são diferentes, como ex. massa de cada subsistema. que são comuns a todos os subsistemas.

Portanto todos os subsistemas herdam os parâmetros deste bloco, mas também possuem outros parâmetros que são distintos, ou exclusivos do subsistema em questão. Também no diagrama existe uma ligação do tipo composição, mostrando que cada subsistema é composto por um ou vários equipamentos, sendo que estes equipamen-

tos só possuem valor quando fazem parte do subsistema.

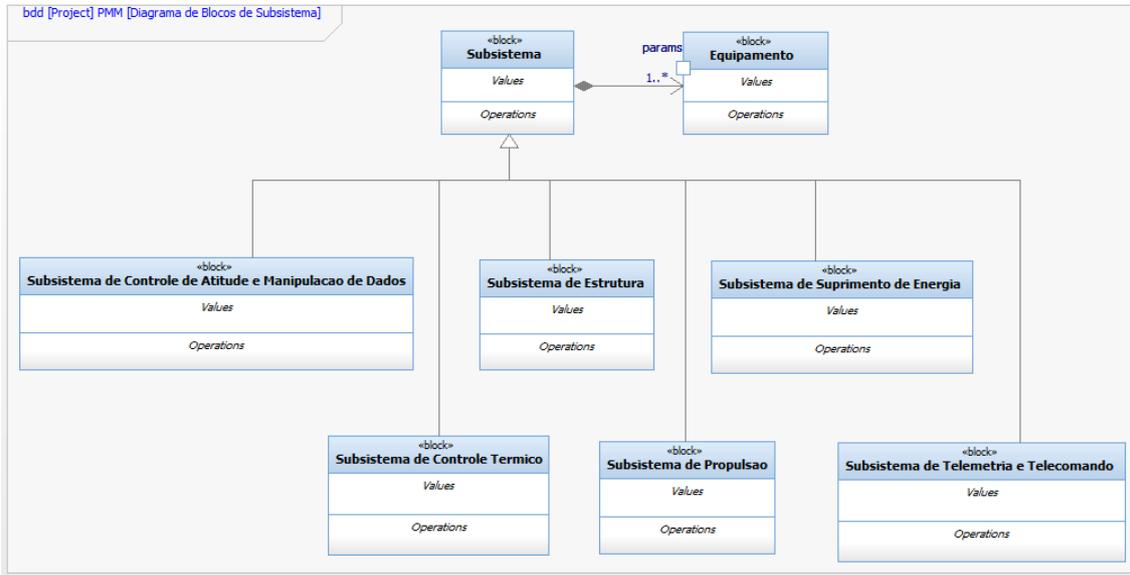


Figura 5.4 - Diagrama de Hierarquia de Subsistemas da PMM

Em seguida foi feito o diagrama de estrutura de Blocos de Restrição (*Constraint Block*) para ilustrar a arquitetura criada e o uso do algoritmo MGEO. Neste diagrama, representado na fig 5.5 é mostrado como cada equipamento é considerado pelo algoritmo e usado para a obtenção do *layout*.

O algoritmo MGEO é representado como um bloco de restrição e composto por três funções objetivo: (1) Distância do Centro de Massa, (2) Distribuição Térmica e (3) Distância Euclidiana dos Equipamentos do subconjunto. Cada uma destas funções é modelada também como bloco de restrição e compõem o algoritmo MGEO. Estas funções estão sendo consideradas como fundamentais para o funcionamento do algoritmo MGEO, mas poderiam ser usadas para outros algoritmos de otimização e, portanto, serem abstraídas usando a ligação de agregação ao invés de composição.

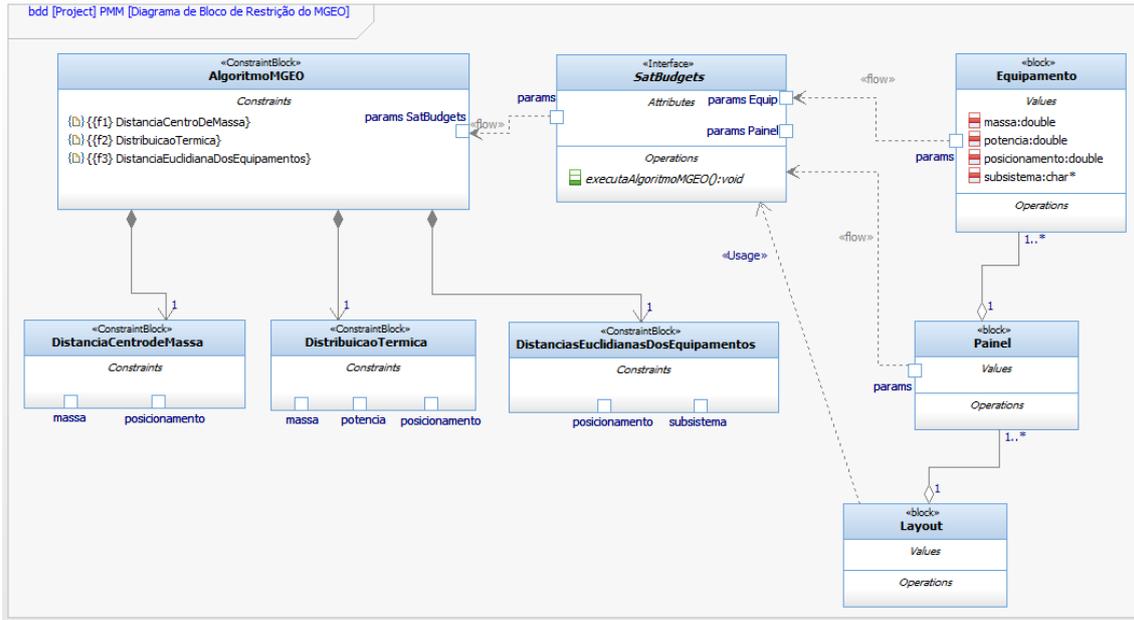


Figura 5.5 - Diagrama Estrutural de Blocos de Restrição com inclusão do bloco de otimização MGEO

5.3 Interoperabilidade entre Modelos e Requisitos

Após a modelagem dos diagramas relacionados à estrutura da PMM, foram criados os requisitos na ferramenta DOORS. Usando como referência o documento Instituto Nacional de Pesquisas Espaciais (INPE) (2001b) da PMM,

Para adicionar os requisitos no DOORS é necessário que estes estejam relacionados a um módulo formal do projeto. Portanto, dentro deste projeto foi criado o Módulo Formal Requisitos de Sistemas conforme mostrado na tela inicial do DOORS na Figura 5.6. Neste módulo, os requisitos foram adicionados dentro módulo, como mostrado na Figura 5.7.

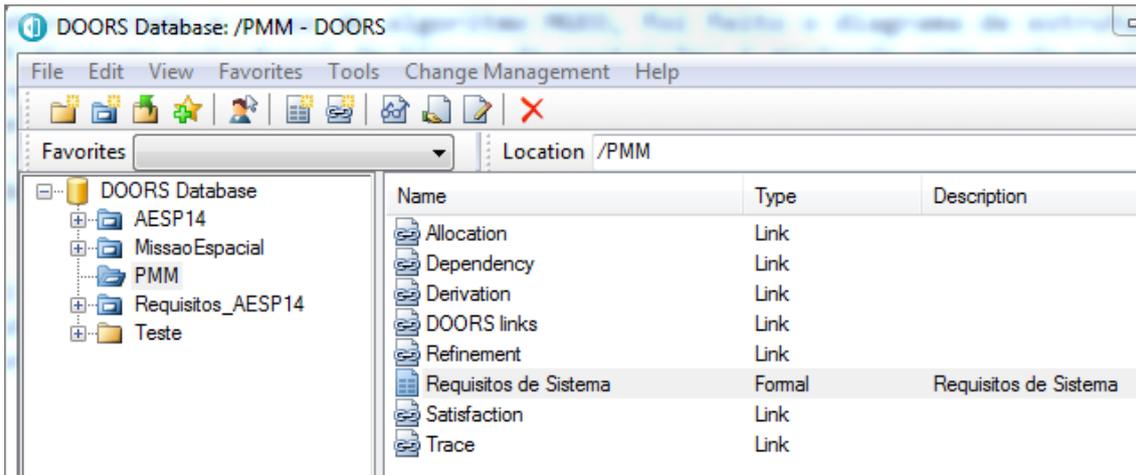


Figura 5.6 - Tela da estrutura inicial do DOORS com módulo “Requisitos de Sistema”

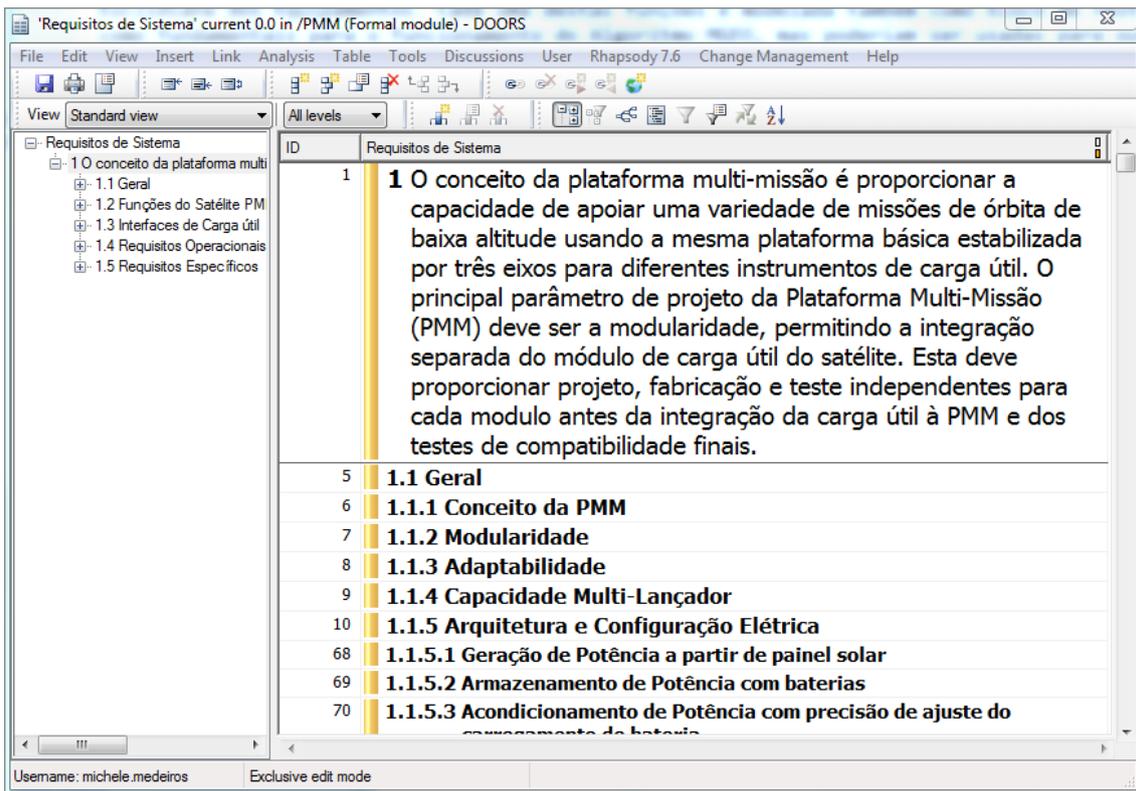


Figura 5.7 - Requisitos da PMM inseridos no DOORS

O DOORS trabalha com objetos, portanto para indicar se um objeto é do tipo requi-

sito, é necessário criar uma coluna adicional chamada "Requirement". Esta coluna é populada com "true" se o objeto é um requisito e "false" caso contrário (Figura 5.7). Esta indicação de objeto do tipo requisito é importante para configurar a geração de relatórios no próprio DOORS e também no Rhapsody Gateway, responsável pela ligação entre as ferramentas DOORS e Rhapsody.

Ao todo foram criados aproximadamente 100 objetos, sendo em sua maioria requisitos. Após inserir os dados no DOORS, foi usado o software Rhapsody Gateway e criado um projeto para ligar o projeto do Rhapsody com o módulo formal do DOORS.

No Rhapsody Gateway foram criados dois documentos, um representando o Modelo SysML do Rhapsody apontando para o projeto SysML e um documento representando o módulo formal do DOORS conforme indicado na Figura 5.8. Após foi criado uma ligação do tipo "*cover link*" indicando que o modelo SysML engloba os requisitos do DOORS.

Finalmente após a criação do modelo, foram importados os requisitos do DOORS para dentro do projeto PMM do Rhapsody. Desta forma é possível fazer a modelagem usando os requisitos já cadastrados no DOORS e evitar retrabalhos. Isto também possibilita a criação de ligações para rastreabilidade entre requisitos e elementos SysML.

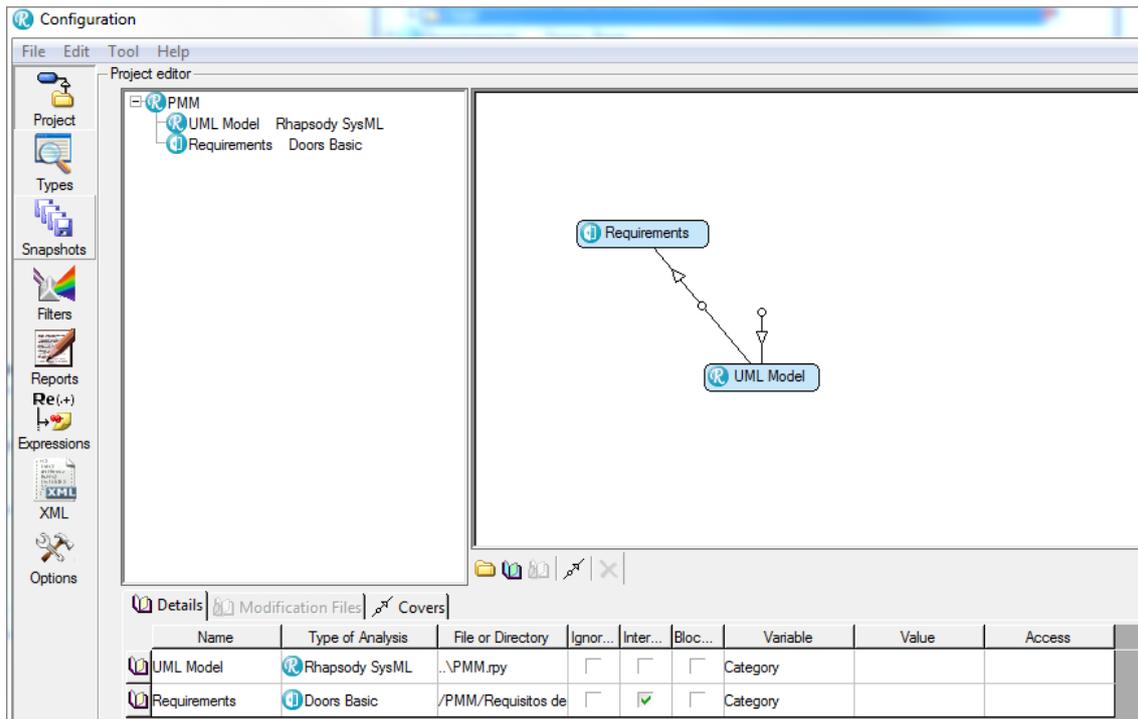


Figura 5.8 - Conexão de Modelos a Requisitos usando o Rhapsody Gateway

5.4 Decomposição de Requisitos da PMM

Diversos diagramas de requisitos foram criados para ilustrar a decomposição dos requisitos da PMM. Como estes diagramas possuem tipos de objetos e relacionamentos similares, somente será discutido um deles aqui e os outros dez diagramas elaborados podem ser vistos no apêndice B.

No diagrama de decomposição dos requisitos, foram usados elementos SysML do tipo Requisitos que foram importados do DOORS, para isso basta arrastar o elemento requisito da árvore do projeto para o diagrama. Estes requisitos já foram relacionados no DOORS através da hierarquia de objetos, portanto o relacionamento no diagrama é automaticamente adicionado no momento da inclusão do Requisito no diagrama. O relacionamento exibido na Figura 5.9, representa um requisito raiz e é do tipo aninhamento da SysML. Isto indica o relacionamento de contenção dos requisitos, ou seja, que requisitos estão aninhados a outros.

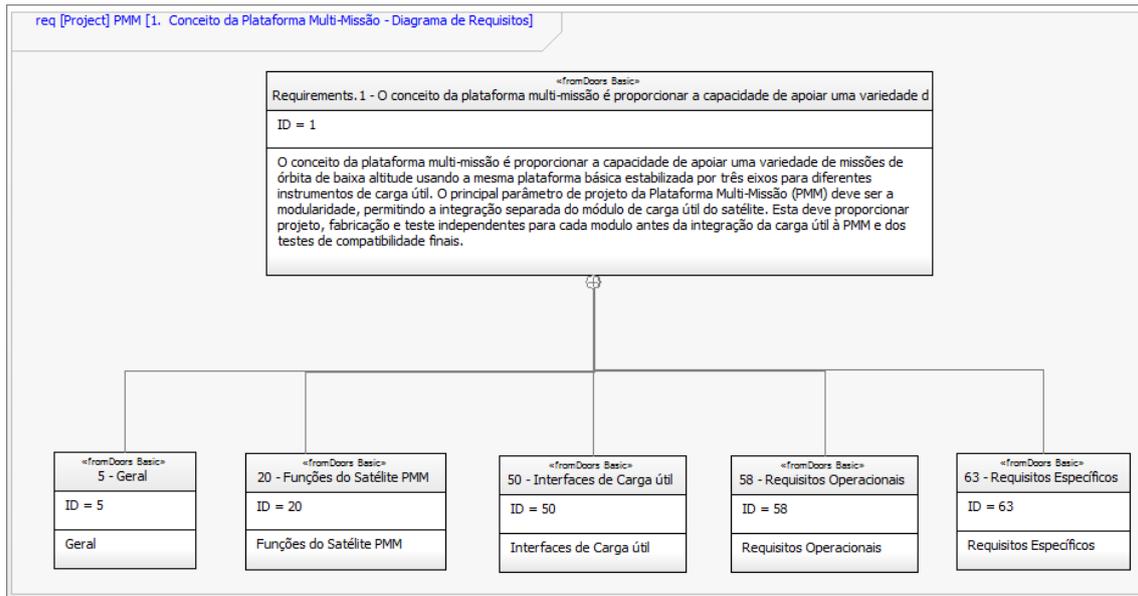


Figura 5.9 - Diagrama de Blocos com Decomposição de Requisitos da Missão da PMM

Visando uma varredura dos requisitos apresentados no documento [Instituto Nacional de Pesquisas Espaciais \(INPE\) \(2001b\)](#) da PMM foi feita, a partir do Requisito de Missão, verificar quais requisitos interferem no layout do satélite. Os resultados da análise destes requisitos serão apresentados a seguir, usando a linguagem SysML.

5.5 Acoplamento e Rastreabilidade de Requisitos PMM ao Problema de *Layout*.

Como mencionado anteriormente, o problema de *Layout* de equipamentos apresenta acoplamento com diversos aspectos da arquitetura de missão da PMM que nesta seção serão apresentadas. Novamente foi usado o software Rhapsody para a modelagem, com a intenção de evitar retrabalho, pois os requisitos já estavam cadastrados no DOORS e importados no Rhapsody. Além de evitar o retrabalho, os modelos também permitem visualizar as ligações feitas no DOORS e no Rhapsody.

5.5.1 Acoplamento com o Segmento de Lançamento.

O lançamento e o veículo lançador restringem de forma direta o projeto de satélite. Durante o lançamento, equipamwntos e seu *layout* devem tipicamente suportar as cargas geradas neste ambiente, e as frequências naturais geradas pelo veículo lançador.

A Figura 5.10 mostra como o acoplamento de requisitos ocorre gradualmente.

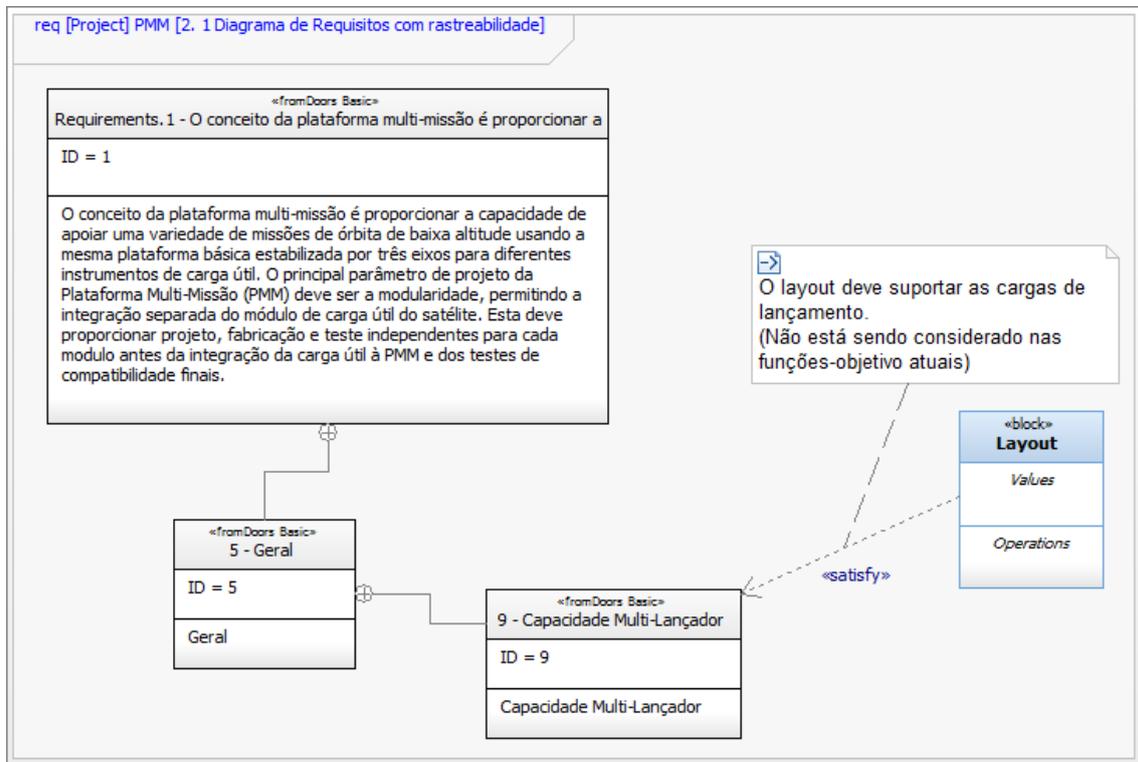


Figura 5.10 - Acoplamento de Lançamento - *Layout*.

5.5.2 Acoplamento com a Arquitetura e Configuração Elétrica.

A decomposição de requisitos para a arquitetura elétrica até o acoplamento ao *layout* é ilustrado na Figura 5.11

Durante a montagem do layout deve ser levado em consideração os requisitos de Integração Elétrica. Em específico a cablagem, afeta vários fatores como alteração da massa, a manuseabilidade durante a integração e teste, interferência eletromagnética. Portanto é vantajoso sua otimização e minimização. Mediante funções-objetivo usadas no algoritmo MGEO.

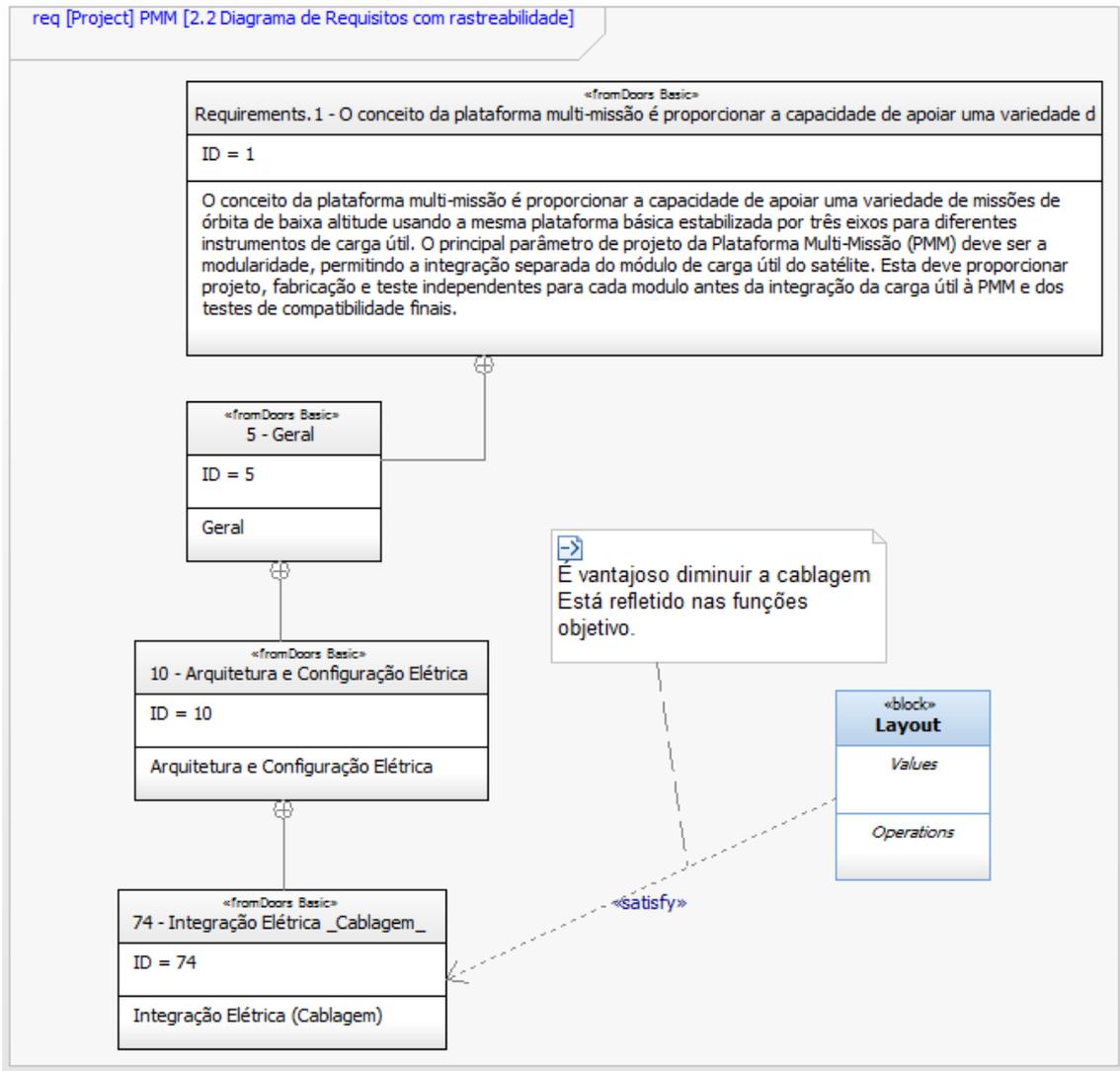


Figura 5.11 - Acoplamento de Arquitetura e Configuração Elétrica - *Layout*.

5.5.3 Acoplamento com a Arquitetura Mecânica.

A Arquitetura mecânica é a responsável pela configuração mecânica, incluindo o *layout*, *per se* portanto seus requisitos influenciam diretamente o projeto. Um exemplo é o requisito que especifica as dimensões do satélite que definem a área disponível para os equipamentos serem acondicionados.

O requisito 86 da Figura 5.13, define as áreas que estão disponíveis para o posicionamento, pois especifica que as antenas e sensores devem ser posicionados no exterior do satélite e que alguns blocos serão móveis.

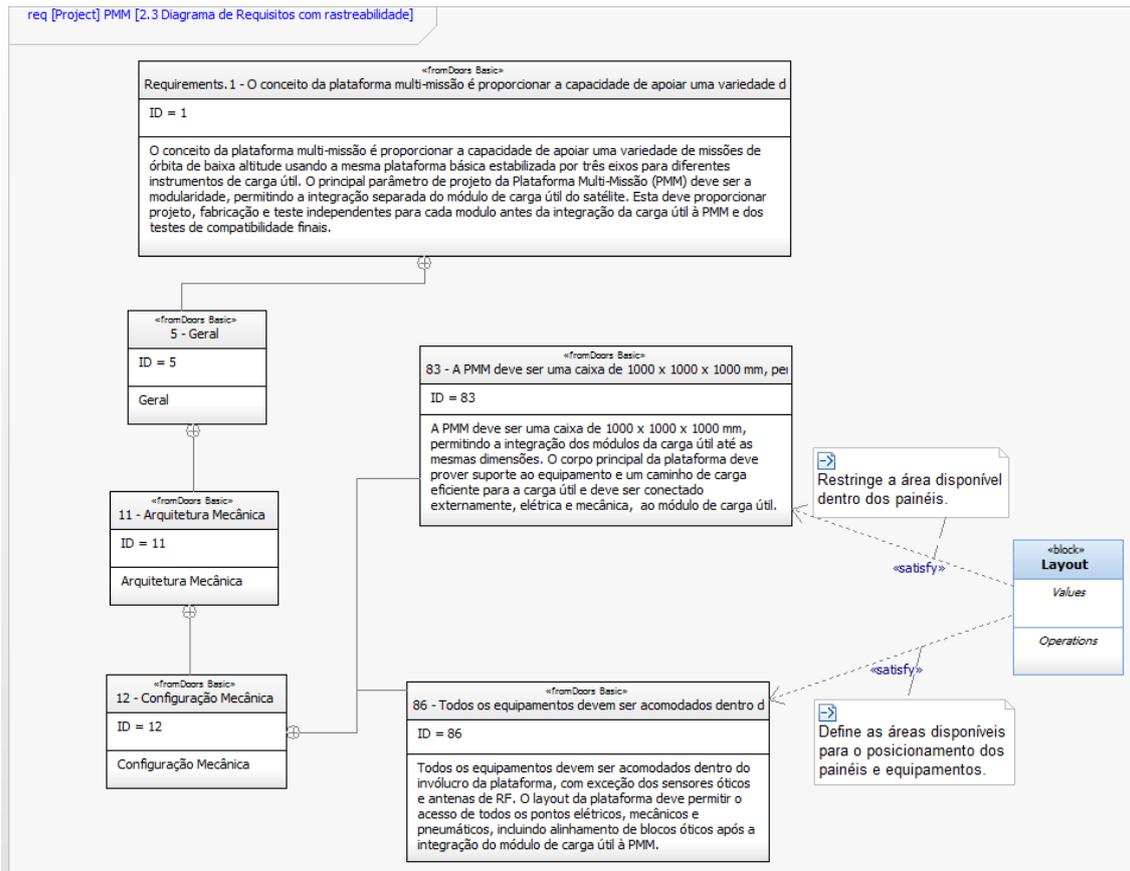


Figura 5.12 - Acoplamento com a Arquitetura Mecânica - *Layout*: área de posicionamento

Adicionalmente o posicionamento dos equipamentos afeta diretamente o centro de massa do satélite, que influencia o controle de atitude, órbita do mesmo e o projeto das estruturas para atender as cargas do lançamento. Este acoplamento é explicitado na Figura 5.13

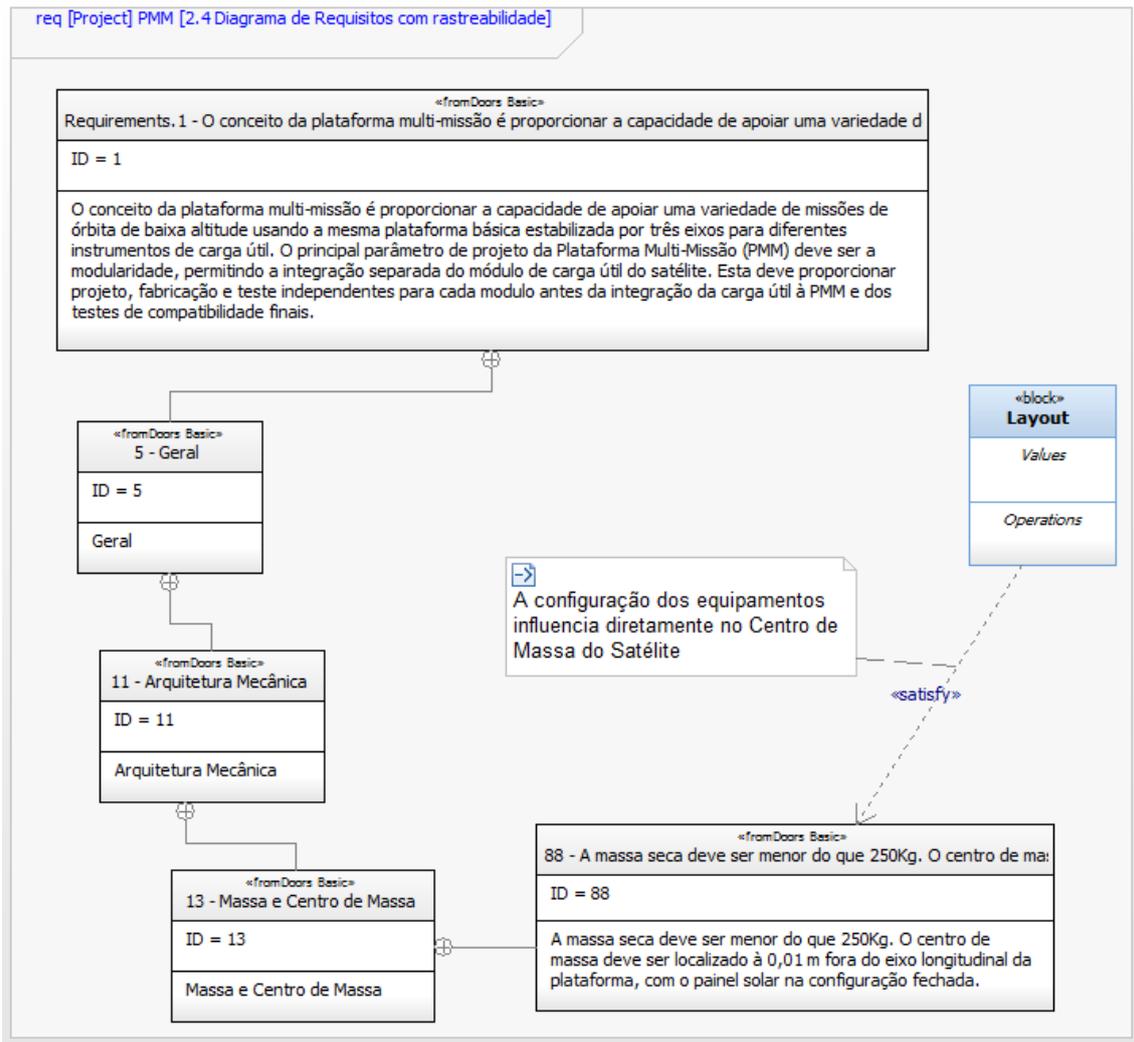


Figura 5.13 - Acoplamento com Arquitetura Mecânica - *Layout*: centro de massa com rastreabilidade.

5.5.4 Acoplamento com Requisitos de Órbita.

Como visto anteriormente, a órbita escolhida afeta em vários fatores no *layout*. Aqui é ressaltado que o posicionamento de alguns equipamentos podem sofrer restrições, pois estes não podem ser posicionados apontados para o Sol, por exemplo. Este acoplamento entre requisitos na hierarquia de derivação dos mesmos é ilustrada na Figura 5.14 via ligação ao requisito 93.

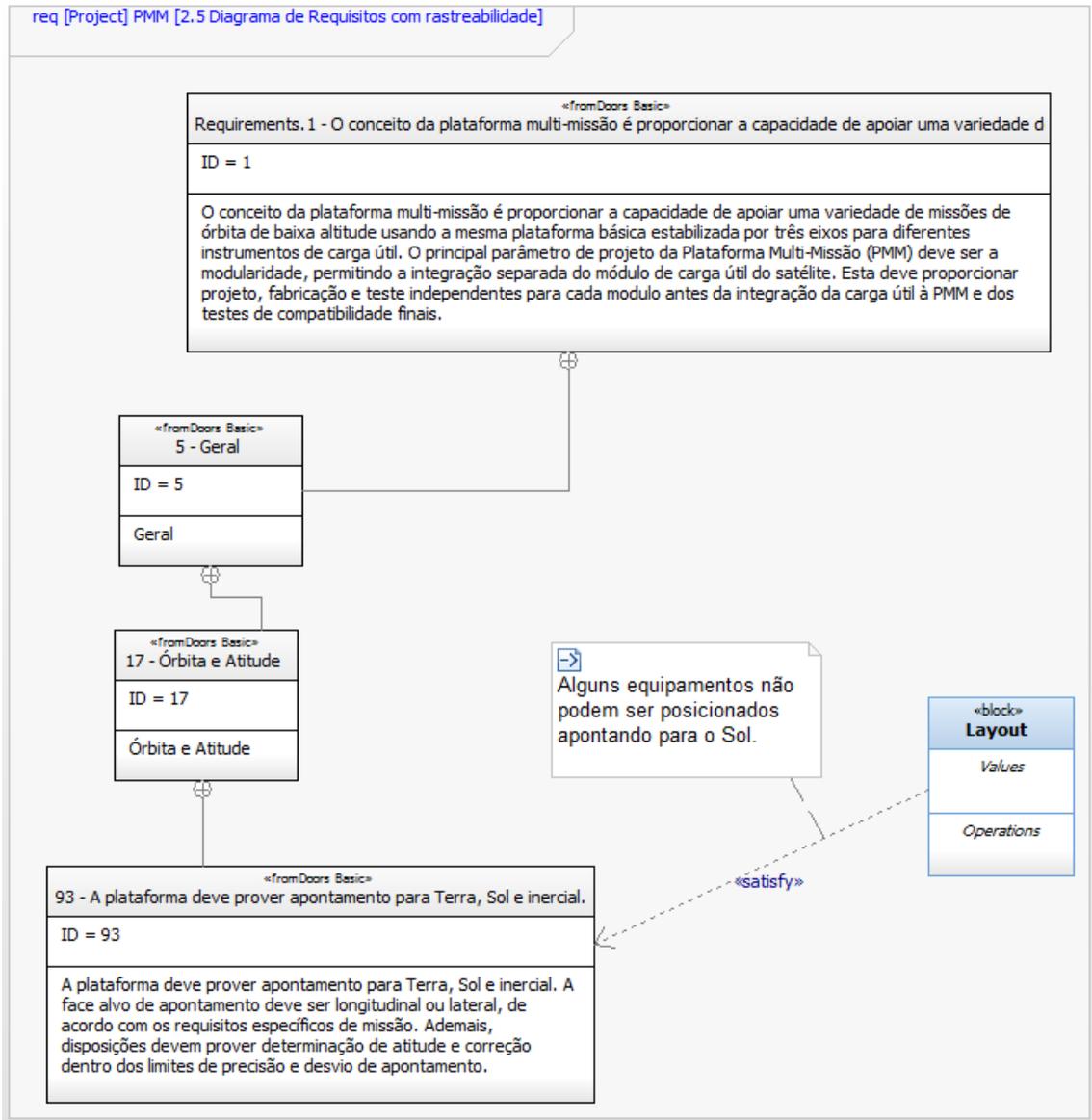


Figura 5.14 - Acoplamento com Órbita - *Layout*: posição relativa ao sol.

5.5.5 Acoplamento com o Subsistema de Telecomunicações.

O Subsistema de Telecomunicações tipicamente define o posicionamento de suas antenas para atender os requisitos da missão. Devido aos requisitos de enlace, o posicionamento de antenas afeta o diagrama de irradiação das mesmas como mostrado na Figura 5.15

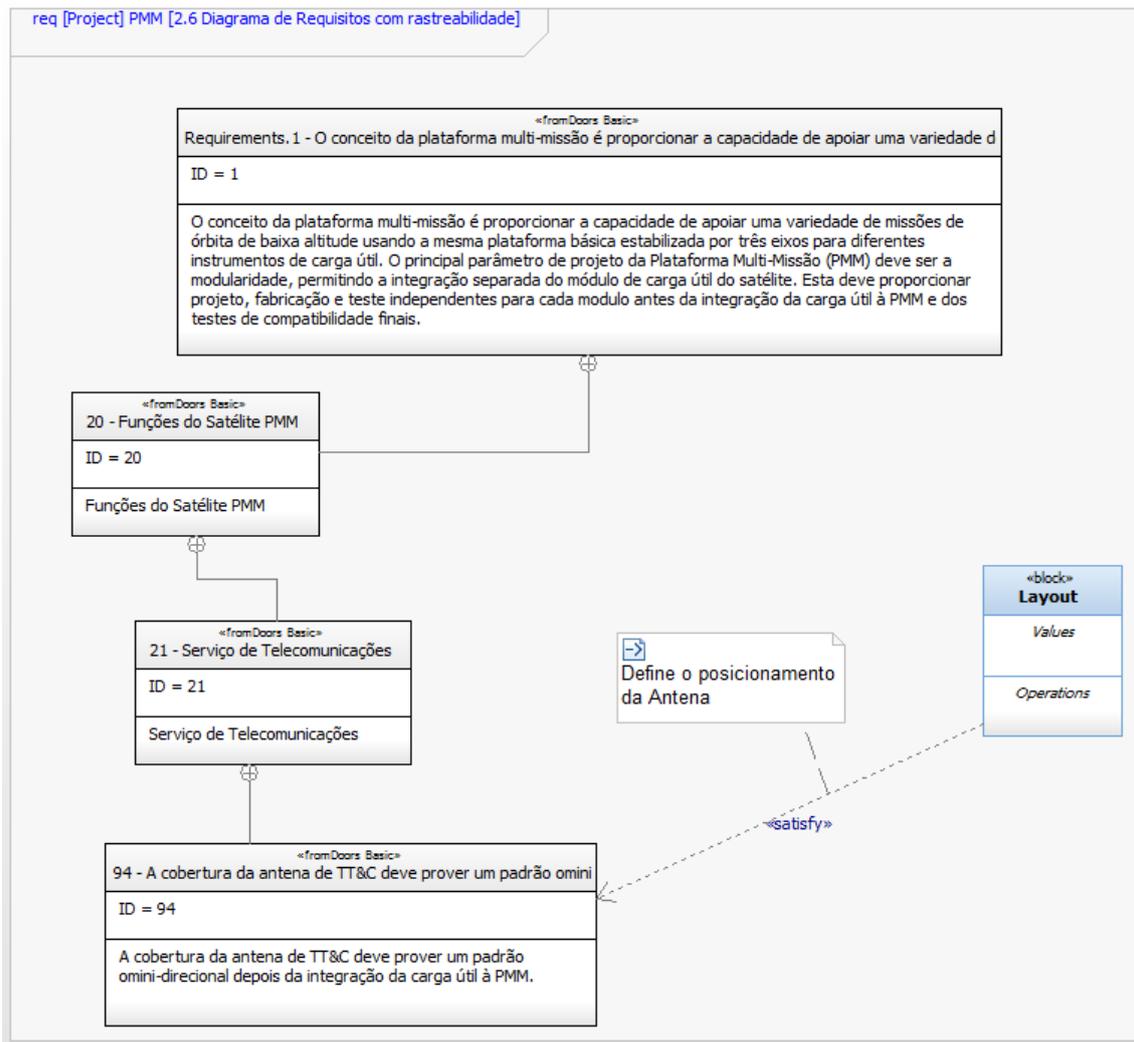


Figura 5.15 - Acoplamento com Sistema de Telecomunicações - *Layout*: posicionamento de antenas.

5.5.6 Acoplamento com o Subsistema de Controle de Atitude e Determinação de Órbita - *Layout*.

A derivação de acoplamento entre o Subsistema de Controle de Atitude e Determinação de Órbita com o resultado final de um *layout* possui vários aspectos (LARSON; WERTZ, 1999). Um desses é destacado na Figura 5.16.

Do ponto de vista do controle de atitude é desejável que os eixos principais de inércia do satélite estejam alinhados com os eixos coordenados, facilitando desta forma a manutenção de atitude.

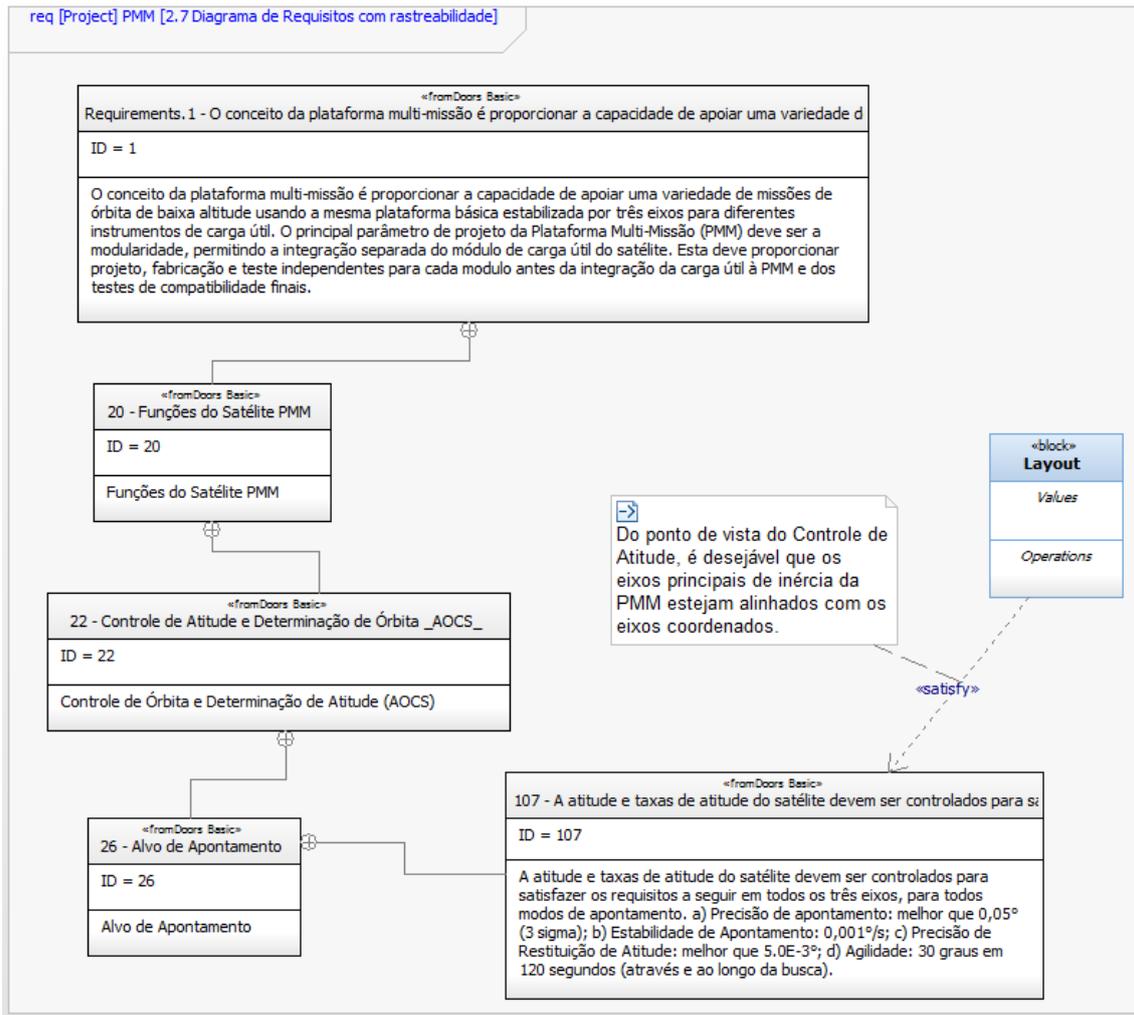


Figura 5.16 - Acoplamento com Subsistema de Controle de Atitude e Determinação de Órbita - *Layout*: eixos principais de inércia.

5.5.7 Acoplamento com o Subsistema de Propulsão.

O subsistema de Propulsão, como mostrado no diagrama da Figura 5.17 afeta no posicionamento dos equipamentos e também na massa. A massa do subsistema de propulsão varia durante a vida útil do satélite, ocasionando alterações no centro de massa, entre outros.

No caso da PMM, o tanque de combustível é posicionado no centro do satélite, minimizando os problemas causados pela variação de massa. Este gera diversas restrições ao problema de layout reduzindo o espaço de busca por soluções outras.

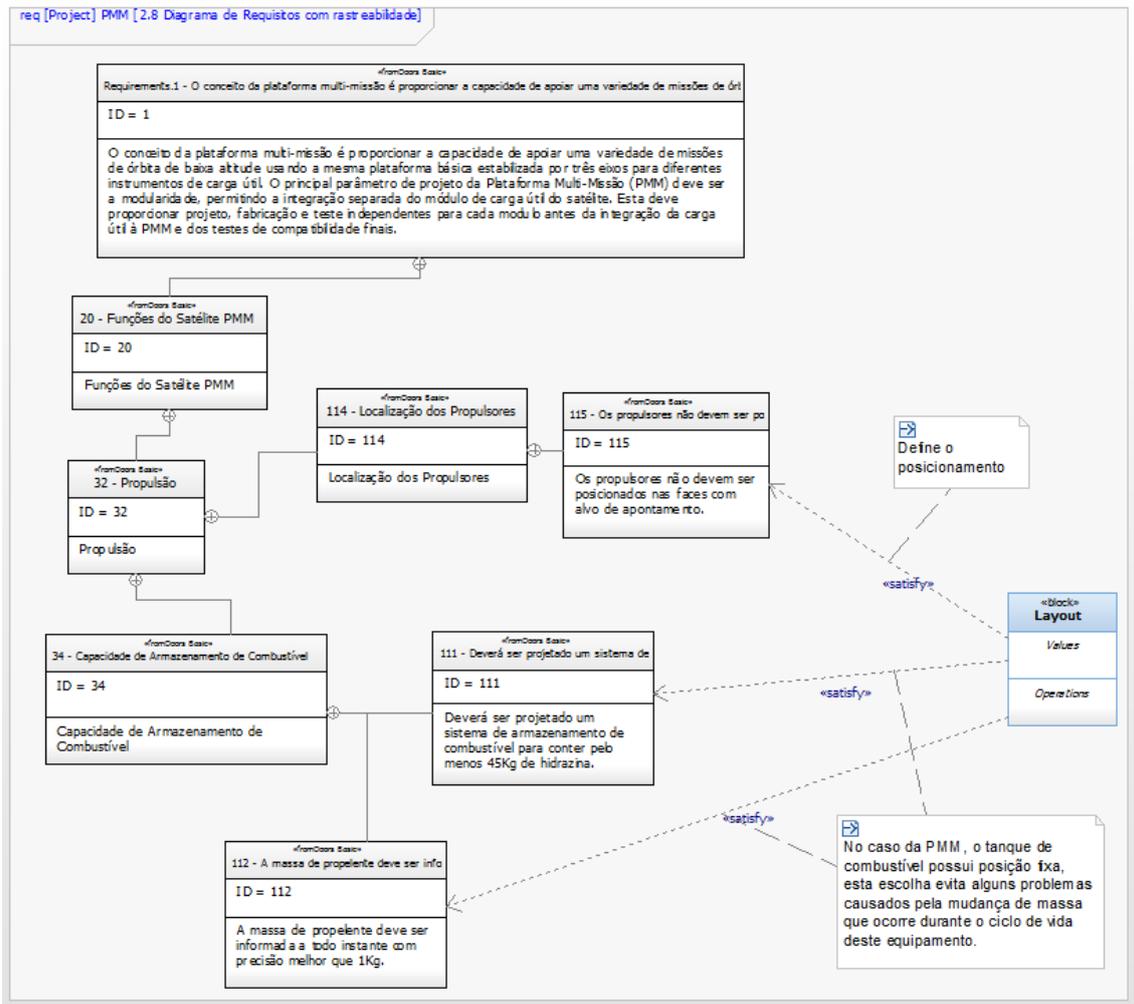


Figura 5.17 - Acoplamento com Subsistema de Propulsão - *Layout*: posição do tanque de combustível.

5.5.8 Acoplamento com o Subsistema de Potência.

O subsistema de Potência se preocupa principalmente com aspectos de geração e armazenamento de energia. Diversos de seus equipamentos possuem desempenho dependente do local em que estão no *layout* (LARSON; WERTZ, 1999). Um desses acoplamentos de projeto, o local de painéis solares é ressaltado na Figura 5.18 para a geração de potência.

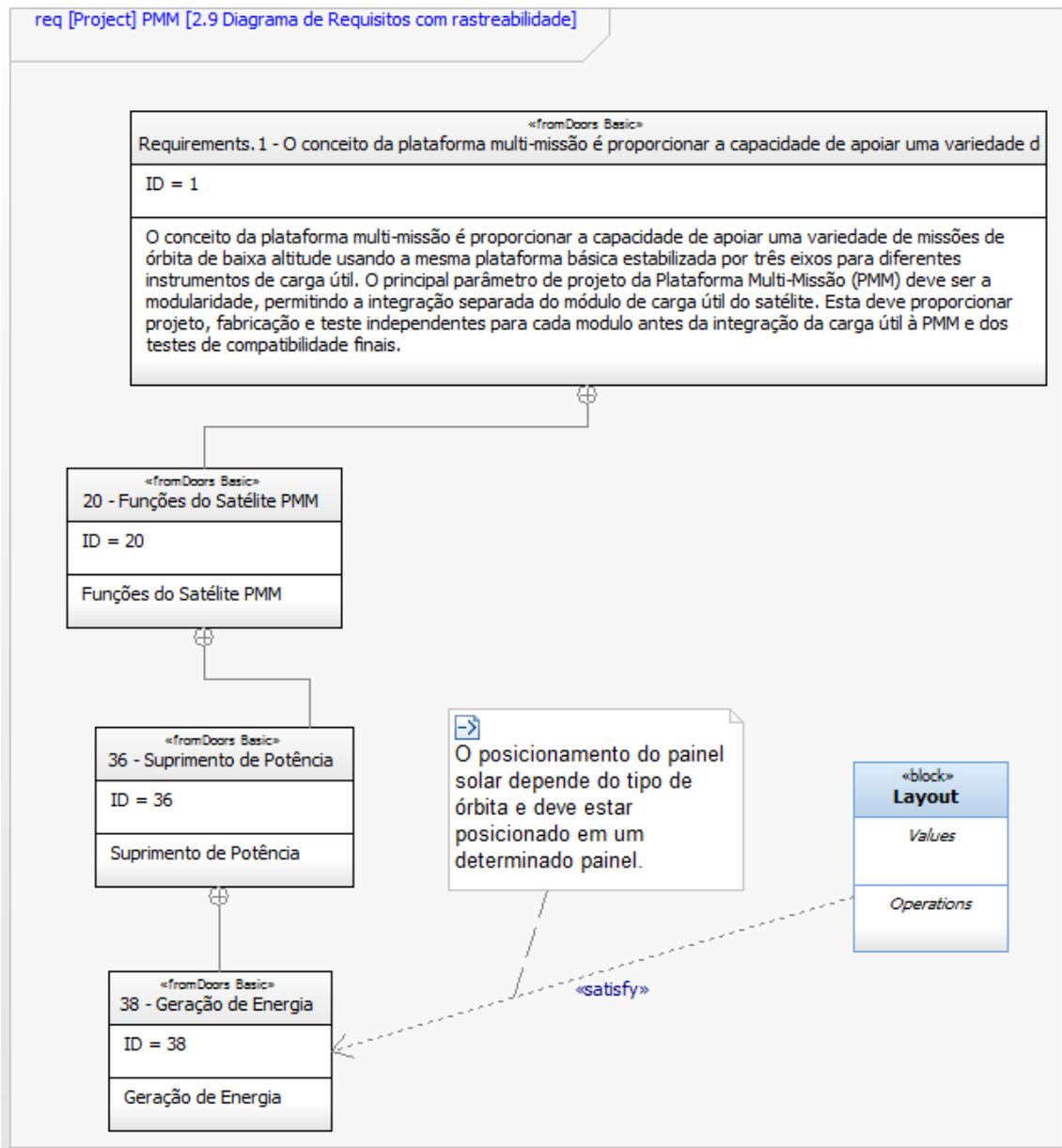


Figura 5.18 - Acoplamento com Subsistema de Potência - *Layout:hotspot*

5.5.9 Acoplamento com o Subsistema de Estrutura.

O Subsistema de Estrutura é o responsável basicamente por prover suporte mecânico e proteção a vibrações e ao ambiente espacial, entre outras funções (LARSON; WERTZ, 1999).

A estrutura sofre diversos tipos de tensão e carregamento estruturais com equipamentos e a área disponível. Este acoplamento é ilustrado brevemente na Figura

5.19.

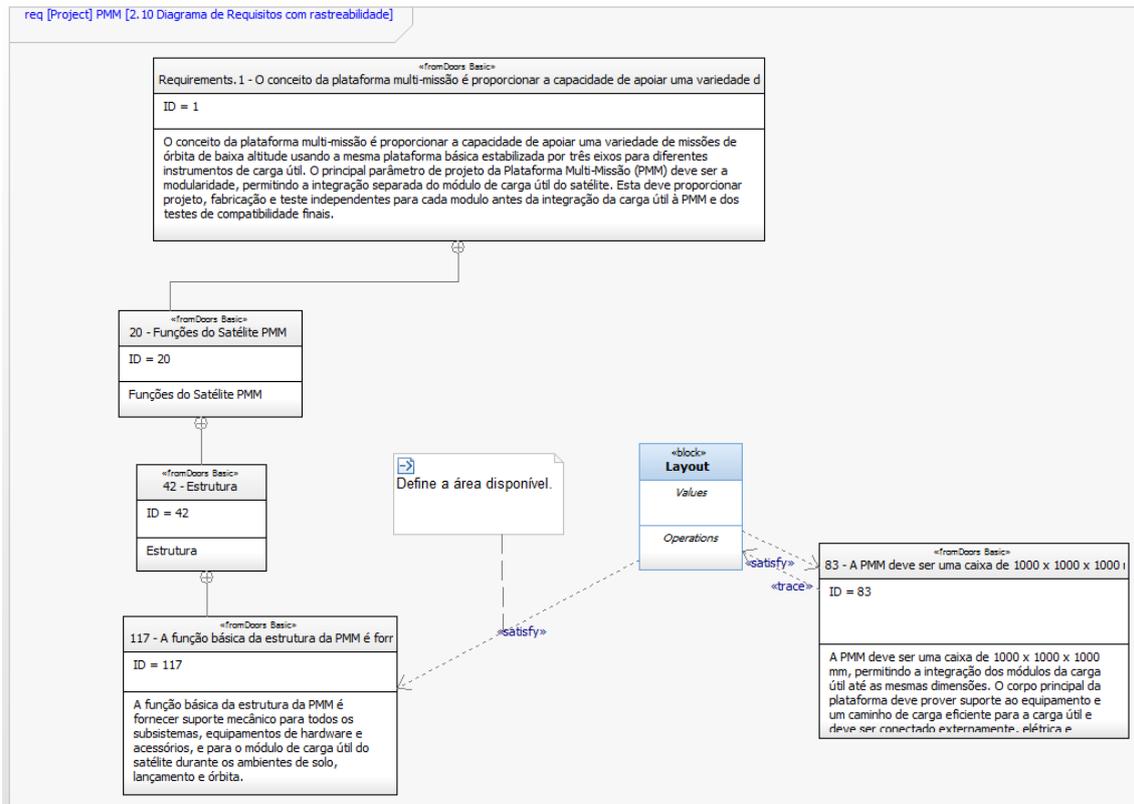


Figura 5.19 - Acoplamento com Estrutura - *Layout*: distribuição de área disponível.

5.5.10 Acoplamento com o Subsistema de Controle Térmico.

O controle térmico é definido em função das características de diversos subsistemas como potência, supervisão de bordo, estrutura entre outras (LARSON; WERTZ, 1999). Uma política de projeto é evitar *hotspots*.

A restrição da faixa de variação da temperatura deve ser respeitada e é desejável que sejam minimizadas para evitar que ocorram (*hot-spots*). Na Figura 5.20 um dos acoplamentos do *layout* é mostrado.

Os equipamentos muito dissipativos não devem ser posicionados no painel de interface da PMM com o módulo de carga útil.

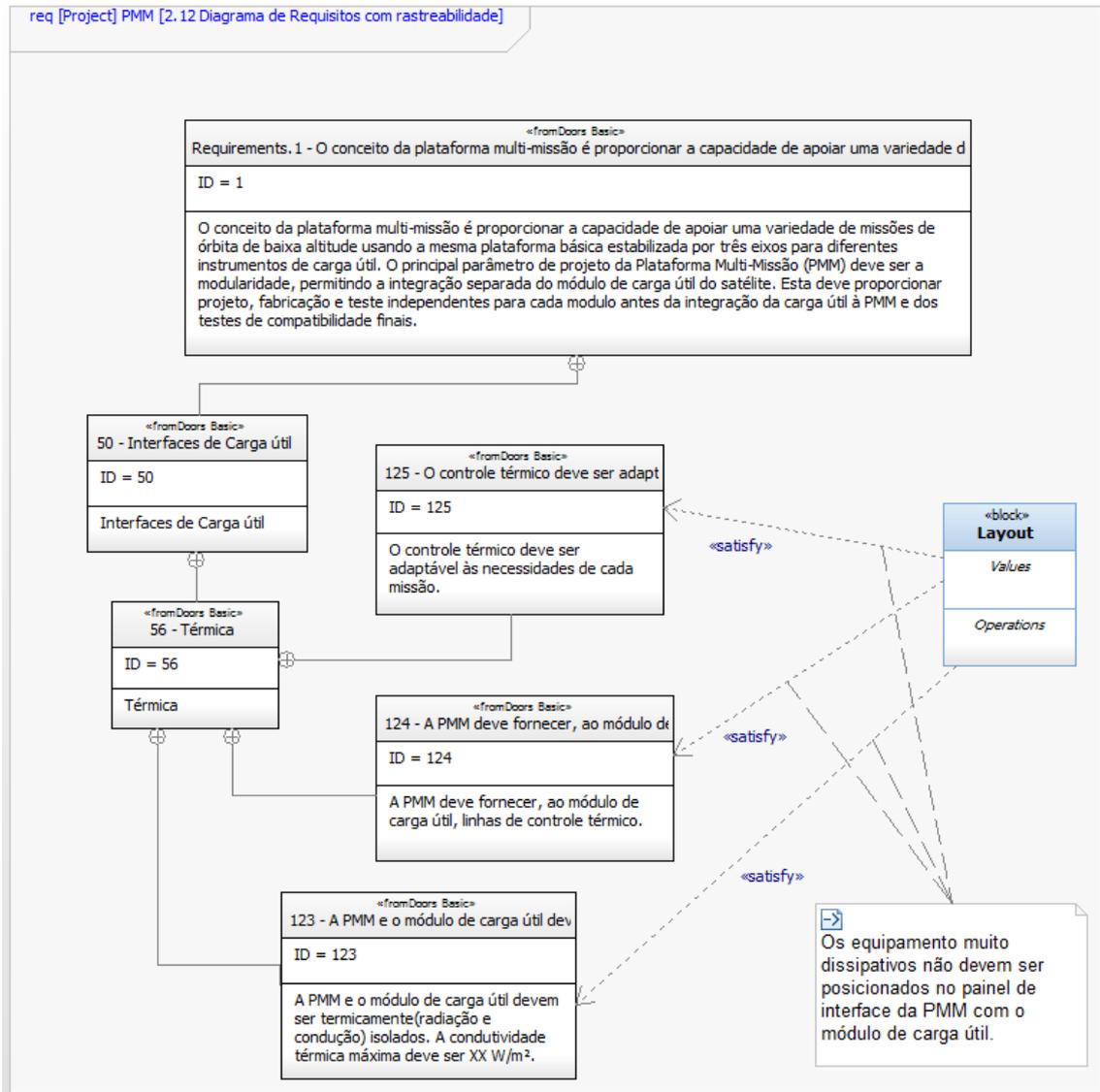


Figura 5.20 - Acoplamento com Controle Térmico - *Layout: hotspot*.

5.5.11 Acoplamento com a Carga Útil.

O subsistema de carga útil apresenta basicamente os mesmos acoplamentos identificados nos subsistemas anteriores descritos.

A Figura 5.21 mostra um acoplamento em particular relativo ao Controle de Interferência Eletromagnética (EMI) que é objeto da arquitetura elétrica de satélites.

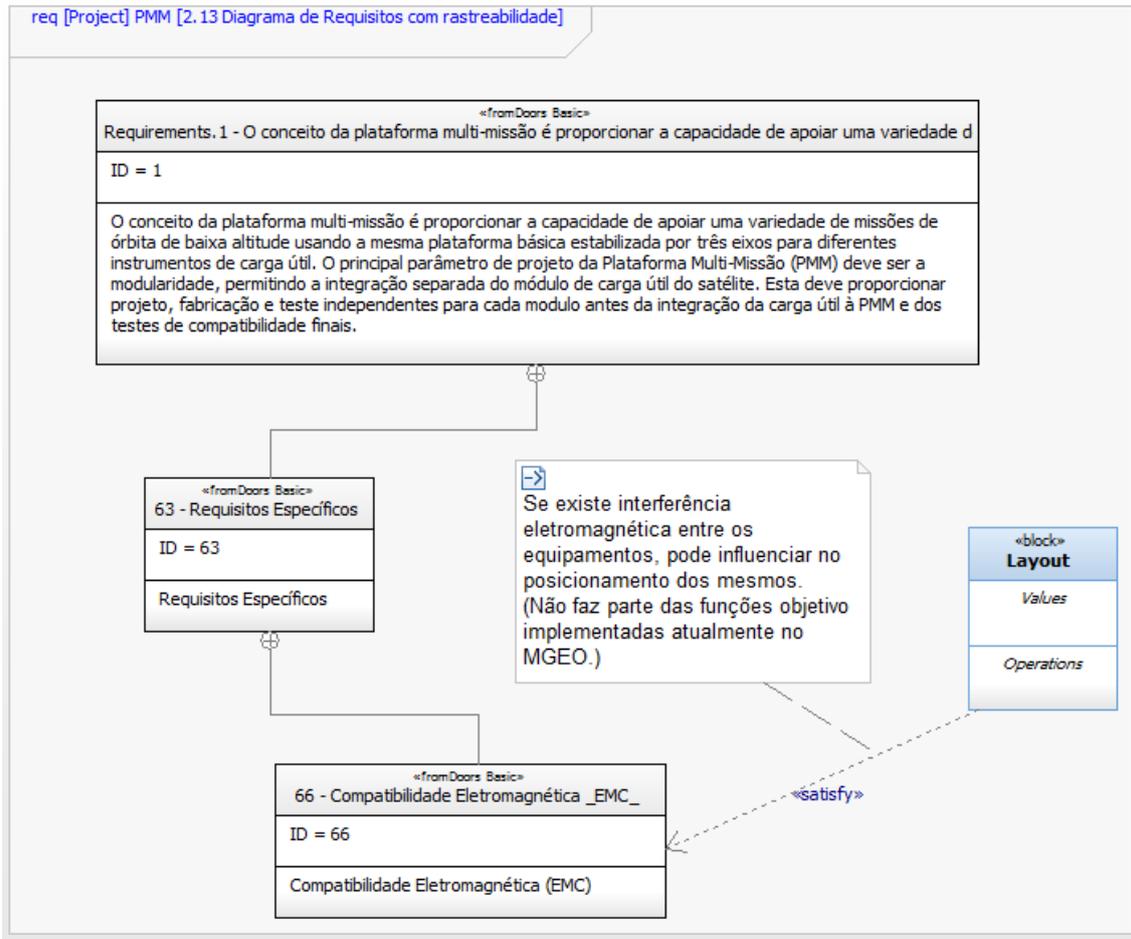


Figura 5.21 - Acoplamento com Carga Útil - *Layout*: EMI.

5.5.12 Acoplamento com o Subsistema de Cablagem.

A cablagem, em alguns satélites, pode ser complexa e longa, justificando um subsistema dedicado. Devido à variedade de sinais elétricos e equipamentos, deve ser verificado se há interferências eletromagnética entre os equipamentos, caso ocorra não devem ser posicionados muito perto uns dos outros, afetando no layout dos equipamentos. Este acoplamento é mostrado na Figura 5.22

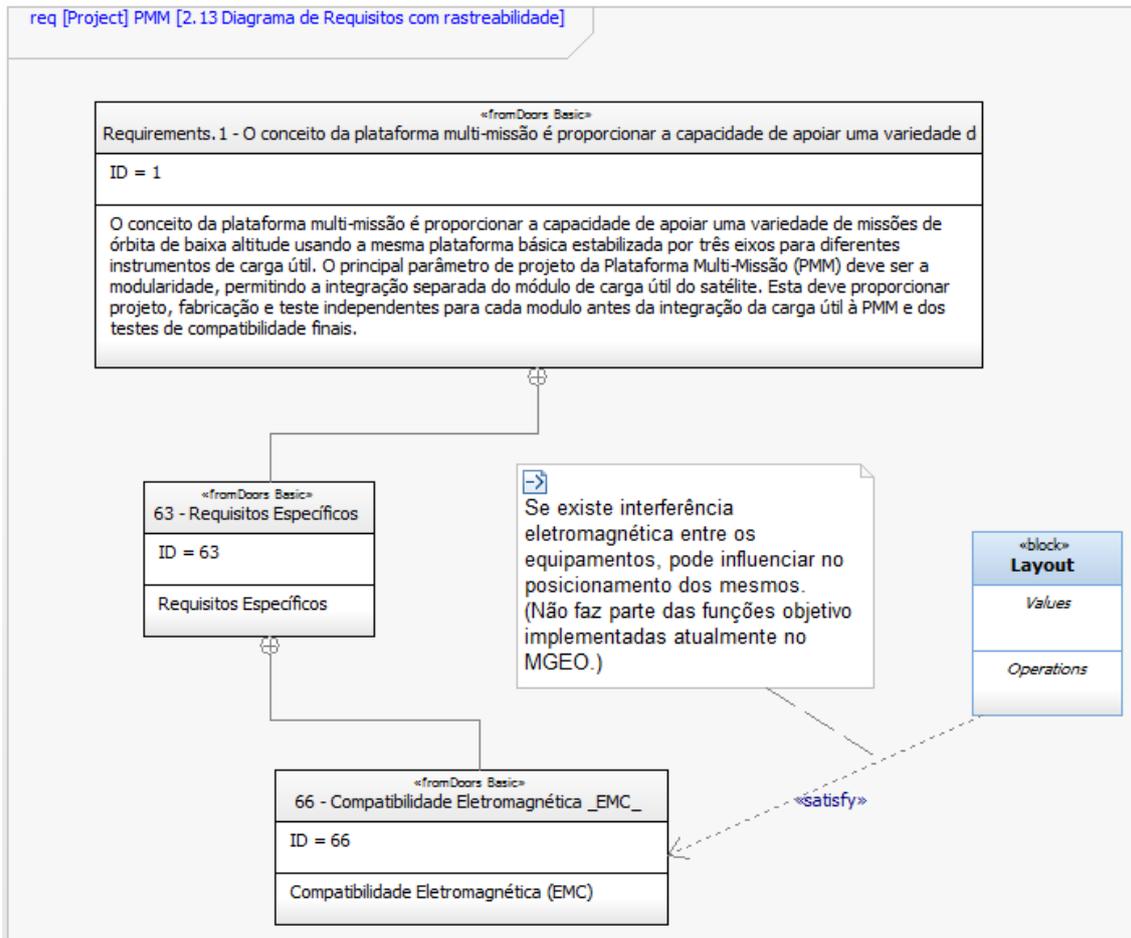


Figura 5.22 - Acoplamento com Compatibilidade Eletromagnética - *Layout*: EMI/EMC.

5.6 Redução do Escopo da Otimização.

Como o trabalho tem ênfase na metodologia e não na abrangência de sua aplicação, o espaço de estudo de caso da otimização foi reduzido a um problema de *layout* bidimensional e uma face apenas com equipamentos. Nas figuras 5.23 e 5.24 são apresentadas tabelas com os dados de alguns equipamentos da PMM. Como referência, um breve diagrama de blocos da PMM é mostrado na Figura 5.25 com 27 blocos.

Equipamento	L(m)	B(m)	H(m)	Massa (kg)	Potência Dissipada (W)	Descrição
SADA	0,14	0,14	0,18	3,37	6,4	<i>Solar Array Drive Assembly</i> : É um dispositivo que faz a interface eletro-mecânica entre o painel solar e a estrutura do satélite. Ele provê o movimento de rotação do painel solar.
SADA	0,14	0,14	0,18	3,37	6,4	
PCDU	0,45	0,265	0,225	18,8	44,3	<i>Power Control and Distribution Unit</i> . É um dispositivo que provê proteção e monitoramento para a corrente elétrica que alimenta o módulo de serviço do satélite.
Magnetotorque	0,5	0,025	0,025	1	3,6	São atuadores operados por campos eletromagnéticos, que servem usualmente para compensar pequenos momentos gerados sobre o satélite por forças externas.
Magnetotorque	0,025	0,5	0,025	1	3,6	
Magnetotorque	0,5	0,025	0,025	1	3,6	
Giro	0,235	0,235	0,17	4,7	20	São dispositivos utilizados para detectar perturbações na atitude do satélite e medir as componentes de sua velocidade angular.
Giro	0,235	0,235	0,17	4,7	20	
Magnetômetro	0,143	0,0762	0,0445	0,5	0,5	É um dispositivo que mede o campo magnético local.
Magnetômetro	0,143	0,0762	0,0445	0,5	0,5	
Transponder	0,156	0,21	0,94	2,8	29,2	É um dispositivo que serve para receber e transmitir dados entre o satélite e a Terra.
Transponder	0,156	0,21	0,94	2,8	29,2	
Diplexer	0,156	0,21	0,25	0,75	1,3	É um dispositivo que permite que uma mesma antena seja usada para recepção e transmissão.
Diplexer	0,156	0,21	0,25	0,75	1,3	
Sensor de Estrelas	0,289	0,1578	0,2093	3,1	10	É um dispositivo utilizado para determinar a atitude de um satélite, por meio da comparação entre as estrelas posicionadas no seu campo de visão e um mapa estelar.
Sensor de Estrelas	0,289	0,1578	0,2093	3,1	10	
GPS Elet	0,152	0,132	0,043	1,2	12	Em satélites, é um dispositivo que utiliza o sistema de posicionamento global (Global Positioning System) para determinar a posição e velocidade do satélite. Pode ser também utilizado para determinação de sua atitude.

Figura 5.23 - Dados dos equipamentos da PMM.

Fonte: Cuco (2011)

Equipamento	L(m)	B(m)	H(m)	Massa (kg)	Potência Dissipada (W)	Descrição
Computador Embarcado	0,44	0,321	0,269	24	62	Faz o processamento e gerenciamento dos dados e comandos necessários ao funcionamento de equipamentos de bordo, bem como a supervisão da transmissão de telemetrias e telecomandos entre o satélite e a estação terrena.
Roda de Reação	0,247	0,247	0,085	4,85	55	São dispositivos compostos basicamente por um motor elétrico que aciona um rotor, de forma a prover, por variação do momento angular, controle de atitude ao satélite.
Roda de Reação	0,247	0,085	0,247	4,85	55	
Roda de Reação	0,085	0,247	0,247	4,85	55	
Roda de Reação	0,247	0,247	0,247	4,85	55	
Bateria	0,166	0,229	0,095	4	4,71	Armazena a energia elétrica produzida pelo painel solar para ser usada quando o satélite não estiver sendo iluminado pelo Sol.
Bateria	0,166	0,229	0,095	4	4,71	
Bateria	0,166	0,229	0,095	4	4,71	
Bateria	0,166	0,229	0,095	4	4,71	
Tanque de Hidrazina	0,4841	0,4841	0,4969	6,01	0	Serve para armazenar a hidrazina utilizada pelo subsistema de propulsão do satélite.

Figura 5.24 - Dados dos equipamentos da PMM (continuação).

Fonte: Cuco (2011)

O problema foi então modelado com um diagrama de blocos com oito equipamentos da PMM onde cada um é uma instância do bloco “Equipamento” modelado, portanto, possuem os mesmos parâmetros, mas com valores diferentes conforme figura 5.26.

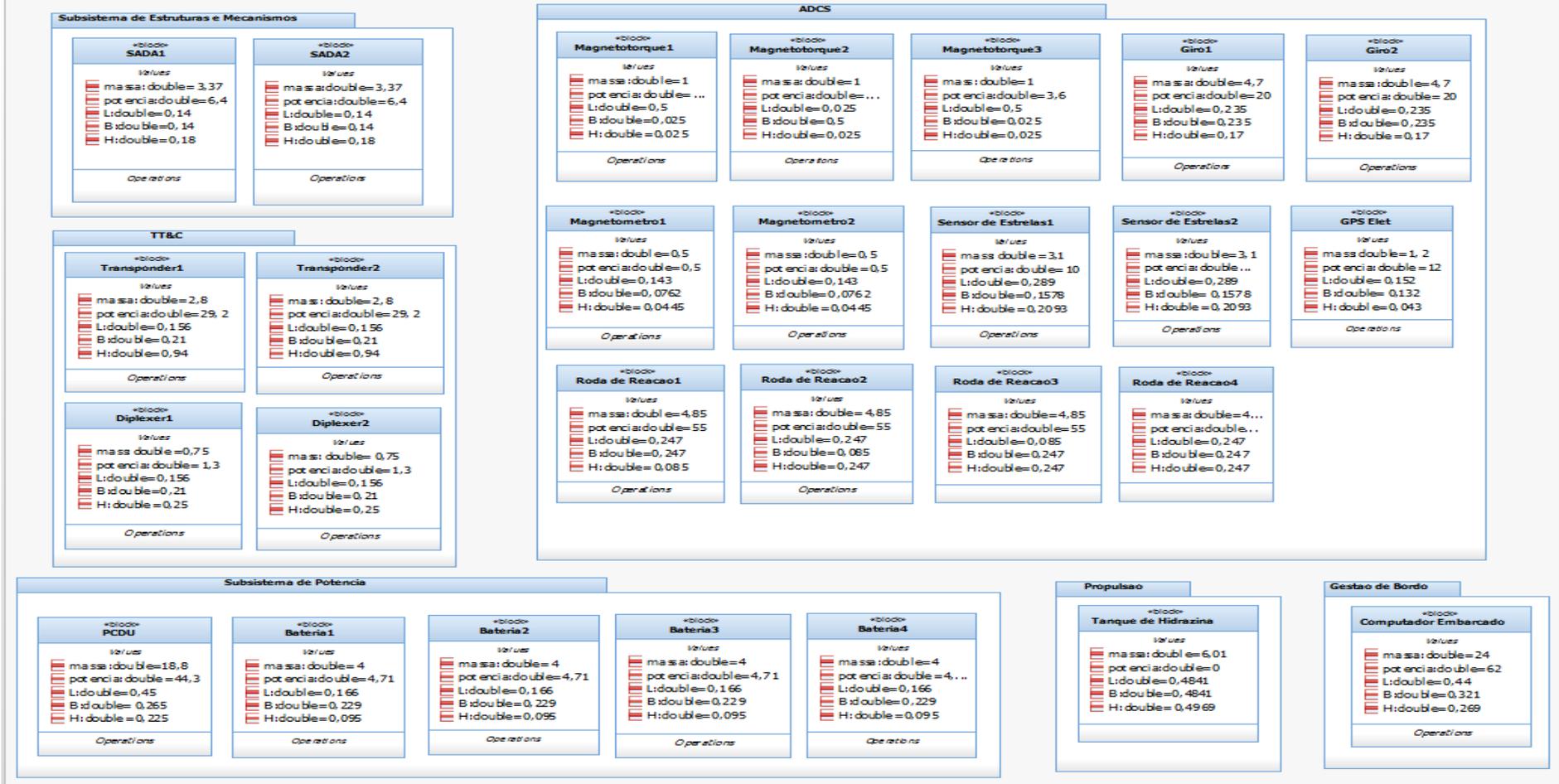


Figura 5.25 - Breve Diagrama de Blocos com os equipamentos da PMM.

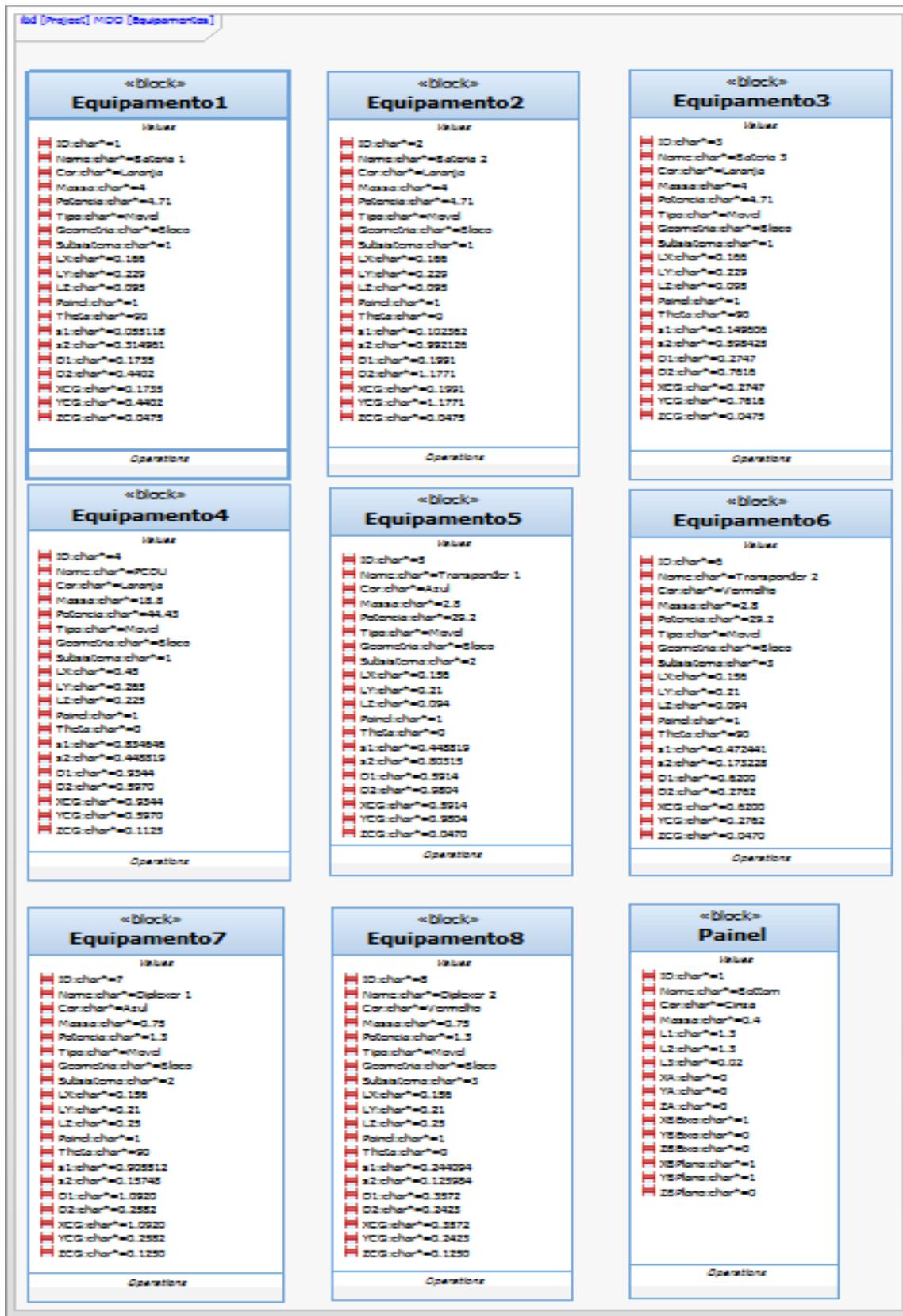


Figura 5.26 - Diagrama de Blocos com os oito equipamentos da PMM usados na otimização para escopo reduzido de layout bidimensional.

5.7 Parametrização da Execução do MGEO.

Nesta seção faz-se a parametrização no uso da infraestrutura de software, anteriormente apresentada no Capítulo 4 bem como o uso da ferramenta DOORS para popular e ligar requisitos, Rhapsody para a modelagem em SysML, e o uso do complemento (*plugin*) Rhapsody Gateway para a interoperabilidade.

Para tanto é necessário exportar o diagrama de blocos criado na Figura 5.26 para um padrão que possa ser lido pelo SatBudgets. No caso do Rhapsody, basta selecionar o diagrama com os equipamentos no formato XMI, padrão para troca de informações.

Em seguida, o arquivo XMI é gerado e salvo no destino selecionado e já pode ser usado dentro da ferramenta SatBudgets. Na interface do software SatBudgets (navegador) deve ser selecionado o arquivo XMI gerado pelo Rhapsody e inserido, nos campos da interface, as informações necessárias para a configuração do algoritmo MGEO. Conforme parametrização representada na Tabela 5.1 e 5.2 que configuram respectivamente as variáveis de projeto e o algoritmo de otimização.

Tabela 5.1 - Parâmetros das variáveis de projeto.

Parâmetros	s	θ
número de bits por variáveis	5	1
valor mínimo para as variáveis	0	0
valor máximo para as variáveis	1	90

Tabela 5.2 - Parâmetros das variáveis do algoritmo MGEO.

Parâmetro τ	1
Número máximo de soluções no arquivo externo	10
Número máximo de avaliações da função objetivo	100
Número máximo de re-inicializações ao longo da busca	5

5.8 Resultados Obtidos na Otimização MBSE-MDO

No estudo de caso foram selecionadas inicialmente para o algoritmo dez soluções de arquitetura para oito equipamentos. Os resultados destas soluções de arquiteturas de *layout* são representados em um gráfico de bolhas mostrado na Figura 5.27. No

o eixo horizontal é representada a função objetivo de que os equipamentos do mesmo subsistema fiquem próximos entre si, no eixo vertical a função objetivo de que o centro de massa do sistema esteja o mais próximo possível da posição alvo desejada, e as bolhas, representadas em amarelo, representa a função objetivo de que o calor dissipado pelos equipamentos seja distribuído de maneira uniforme sobre o painel (O tamanho de cada bolha representa proporcionalmente seu valor numérico).

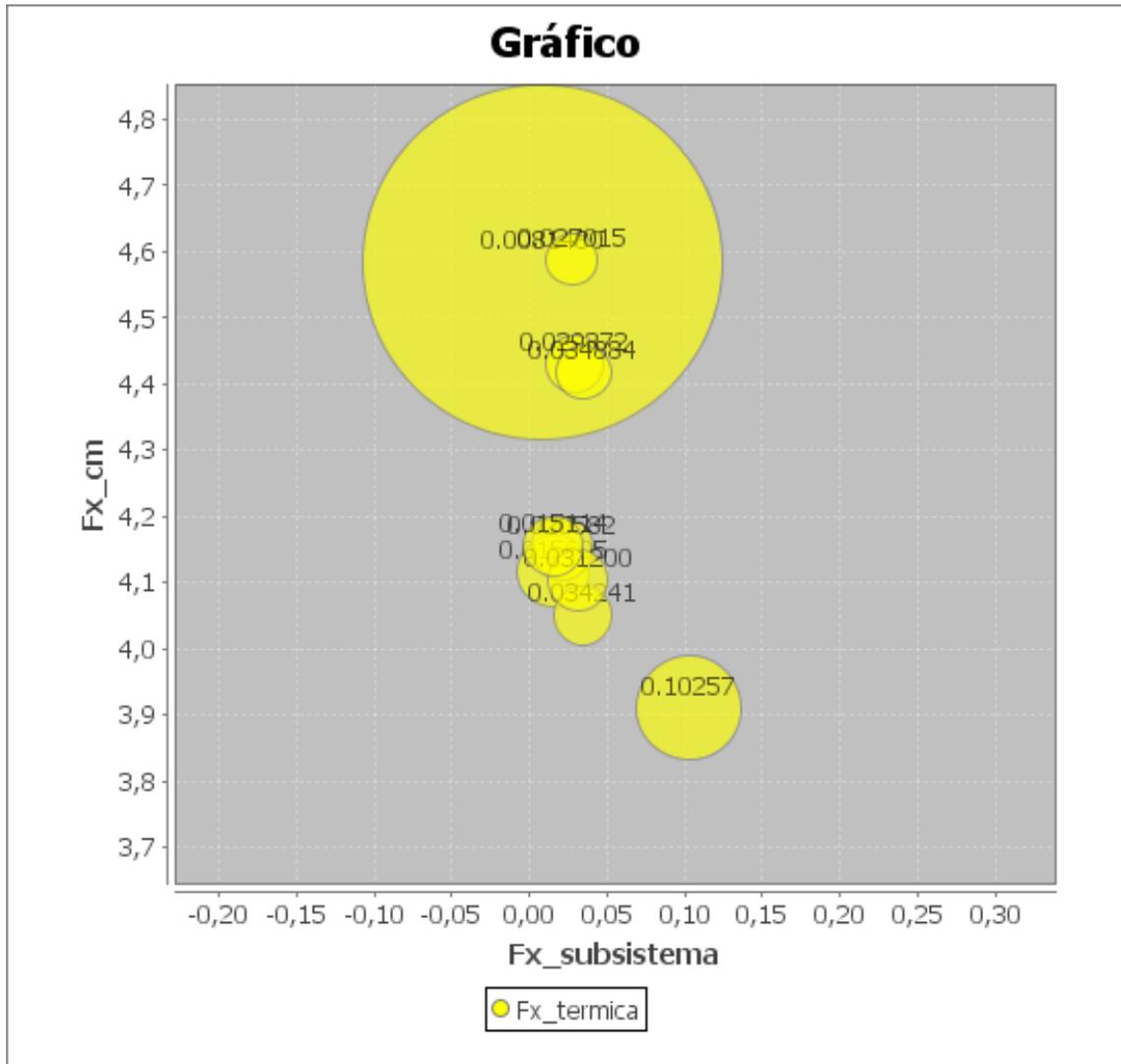


Figura 5.27 - Fronteira de Pareto representada em um diagrama de bolhas de cada resultado/arquitetura encontrado pelo algoritmo MGEO exibido na interface gerada.

Adicionalmente nas figuras os dados resultantes de cada arquitetura são apresentados

também na interface em forma de uma tabela mostrada nas figuras 5.28, 5.29 e 5.30. Cada linha desta tabela (mostrada em 3 partes) possui um *link* para a visualização do posicionamento dos equipamentos da arquitetura através da interface do SolidWorks. A cada *link* criado a interface do Solidworks é chamada e a arquitetura pode ser visualizada graficamente conforme exemplificado nas figuras 5.31(apresenta uma das arquiteturas do resultado em alguns ângulos diferentes). Na figura 5.32 é mostrada todos os *layouts* de arquitetura gerados pela otimização.

Resultados Obtidos (MGEO)

id	fx_Cm	fx_Térmica	fx_Substist	Equi	TETA	s1	s2
1	0.008147991565325197	19107.641026166326	4.58375181239222	1	0	0.36220472440944884	0.5275590551181102
				2	90	0.07874015748031496	0.007874015748031496
				3	0	0.18110236220472442	0.47244094488188976
				4	0	0.7716535433070866	0.7559055118110236
				5	0	0.4566929133858268	0.23622047244094488
				6	90	0.2677165354330709	0.2440944881889764
				7	90	0.6456692913385826	0.047244094488188976
				8	90	0.84251968503937	0.007874015748031496
2	0.10257061557399781	5553.496916088209	3.9109614065378153	1	0	0.36220472440944884	0.5275590551181102
				2	0	0.07874015748031496	0.13385826771653545
				3	0	0.18110236220472442	0.47244094488188976
				4	0	0.7716535433070866	0.5039370078740157
				5	0	0.4566929133858268	0.23622047244094488
				6	90	0.2677165354330709	0.2440944881889764
				7	90	0.6456692913385826	0.047244094488188976
				8	90	0.84251968503937	0.007874015748031496
3	0.0270148760848039	2776.514760681939	4.5879289355846495	1	0	0.36220472440944884	0.5275590551181102
				2	0	0.07874015748031496	0.13385826771653545
				3	0	0.05511811023622047	0.47244094488188976
				4	0	0.8346456692913385	0.7559055118110236
				5	0	0.4566929133858268	0.23622047244094488
				6	90	0.2677165354330709	0.2440944881889764
				7	90	0.6456692913385826	0.047244094488188976
				8	90	0.84251968503937	0.007874015748031496

Figura 5.28 - Tabela com os Resultados encontrados pelo algoritmo MGEO exibido na interface gerada. (Parte 1 de 3). (continua)

4	0.01560516520111836	3843.2540729809284	4.116196312302735	1	0	0.36220472440944884	0.5275590551181102
				2	0	0.07874015748031496	0.13385826771653545
				3	0	0.18110236220472442	0.47244094488188976
				4	0	0.7716535433070866	0.7559055118110236
				5	0	0.4566929133858268	0.11023622047244094
				6	90	0.2677165354330709	0.2440944881889764
				7	90	0.6456692913385826	0.047244094488188976
				8	90	0.84251968503937	0.007874015748031496
5	0.020581518399128765	3319.304718356992	4.151770710301982	1	0	0.36220472440944884	0.5275590551181102
				2	0	0.07874015748031496	0.13385826771653545
				3	0	0.18110236220472442	0.47244094488188976
				4	0	0.7716535433070866	0.7559055118110236
				5	0	0.4566929133858268	0.1732283464566929
				6	90	0.2677165354330709	0.2440944881889764
				7	90	0.6456692913385826	0.047244094488188976
				8	90	0.84251968503937	0.007874015748031496
6	0.0293722082597728	3056.3343972367516	4.428615197943703	1	0	0.36220472440944884	0.5275590551181102
				2	0	0.07874015748031496	0.13385826771653545
				3	0	0.05511811023622047	0.47244094488188976
				4	0	0.7716535433070866	0.7559055118110236
				5	0	0.4881889763779528	0.23622047244094488
				6	90	0.2677165354330709	0.2440944881889764
				7	90	0.6456692913385826	0.047244094488188976
				8	90	0.84251968503937	0.007874015748031496
7	0.03488363677991416	2856.1065035096312	4.417967023526358	1	0	0.36220472440944884	0.5275590551181102
				2	0	0.07874015748031496	0.13385826771653545
				3	0	0.05511811023622047	0.47244094488188976
				4	0	0.7559055118110236	0.7559055118110236
				5	0	0.4566929133858268	0.23622047244094488
				6	90	0.2677165354330709	0.2440944881889764
				7	90	0.6456692913385826	0.047244094488188976
				8	90	0.84251968503937	0.007874015748031496

Figura 5.29 - Tabela com os Resultados encontrados pelo algoritmo MGEO exibido na interface gerada. (Parte 2 de 3). (continua)

8	0.034240845251793384	3137.5449622983238	4.050263736254839	1	0	0.36220472440944884	0.5275590551181102
				2	0	0.07874015748031496	0.13385826771653545
				3	0	0.05511811023622047	0.47244094488188976
				4	0	0.7716535433070866	0.7559055118110236
				5	0	0.4566929133858268	0.23622047244094488
				6	90	0.7716535433070866	0.2440944881889764
				7	90	0.6456692913385826	0.047244094488188976
				8	90	0.84251968503937	0.007874015748031496
9	0.031199880127826166	3235.2632941765196	4.104210012864155	1	0	0.36220472440944884	0.5275590551181102
				2	0	0.07874015748031496	0.13385826771653545
				3	0	0.18110236220472442	0.47244094488188976
				4	0	0.7716535433070866	0.7559055118110236
				5	0	0.4566929133858268	0.23622047244094488
				6	90	0.2677165354330709	0.2440944881889764
				7	90	0.6456692913385826	0.2992125984251969
				8	90	0.84251968503937	0.007874015748031496
10	0.015113654062175007	3187.757403049469	4.155424558676519	1	0	0.36220472440944884	0.5275590551181102
				2	0	0.07874015748031496	0.13385826771653545
				3	0	0.18110236220472442	0.47244094488188976
				4	0	0.7716535433070866	0.7559055118110236
				5	0	0.4566929133858268	0.23622047244094488
				6	90	0.2677165354330709	0.11811023622047244
				7	90	0.6456692913385826	0.047244094488188976
				8	90	0.84251968503937	0.007874015748031496

Figura 5.30 - Tabela com os Resultados encontrados pelo algoritmo MGEO exibido na interface gerada. (Parte 3 de 3). - Conclusão

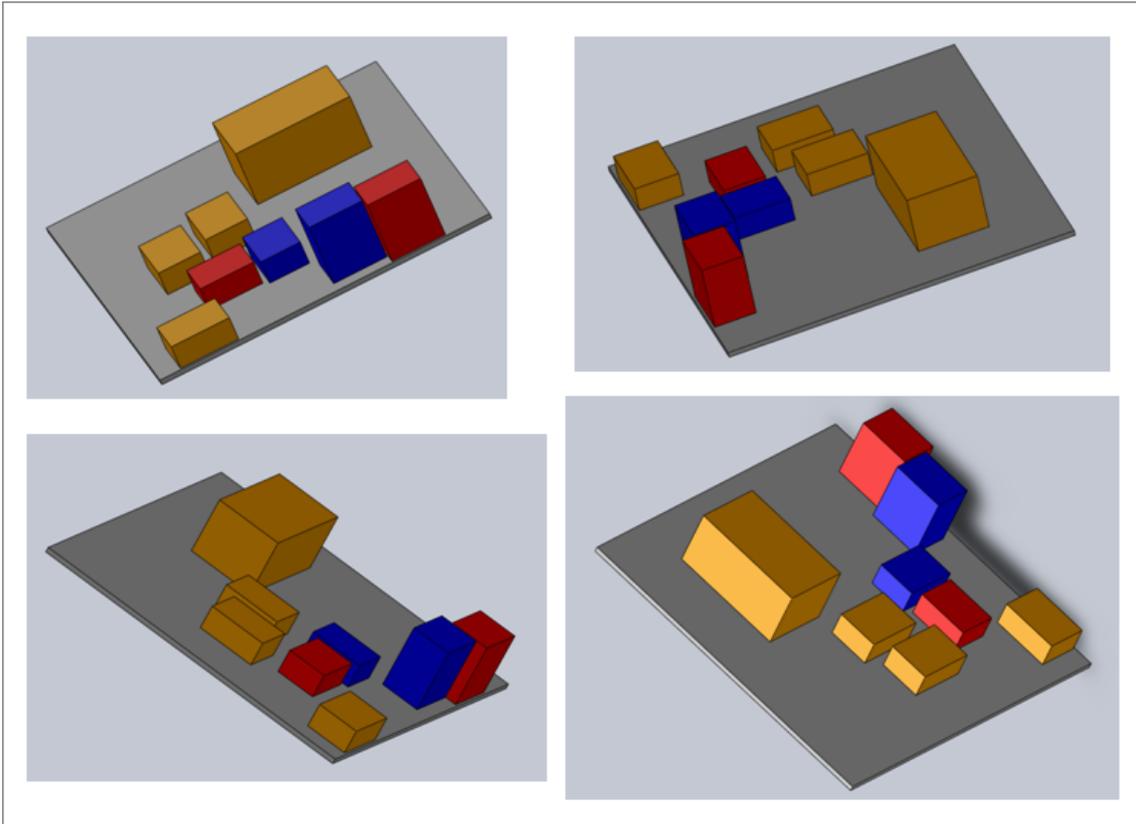


Figura 5.31 - Vistas de uma das soluções (Arquitetura id=1) de *layout* encontradas pelo algoritmo

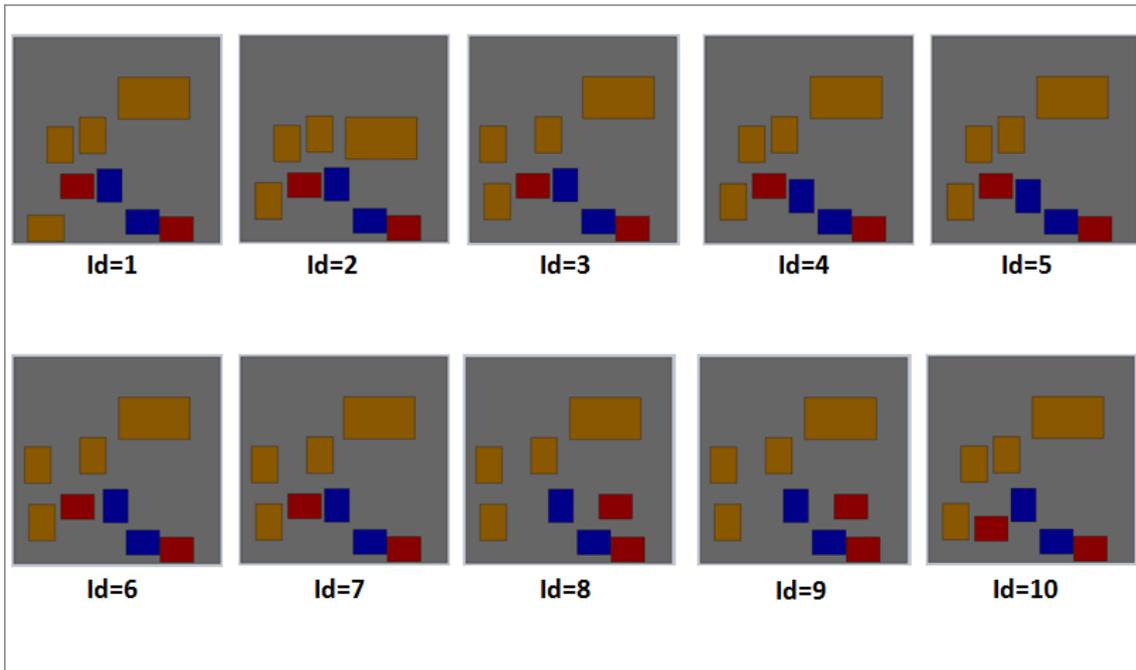


Figura 5.32 - Soluções otimizadas de *layout* encontradas pelo algoritmo MGEO

5.9 Análise dos Resultados

Os resultados obtidos validam o processo de interoperabilidade proposto uma vez que o algoritmo MGEO já havia apresentado resultados satisfatórios em Cuco (2011) e de Sousa et al. (2013).

O foco deste trabalho não é a otimização *per se*, e sim o uso de metodologias, tecnologias, padrões dentro da modelagem de satélites e a criação de uma infraestrutura que permita o uso integrado de MBSE e MDO.

De acordo com a proposta deste trabalho, foi feito o uso de MBSE através do uso de modelagem parcial da PMM através da elaboração de diagramas usando a linguagem SysML, no aplicativo Rhapsody. Também foram citadas algumas vantagens do uso de um software para gerenciamento de requisitos (DOORS) e da ligação dos requisitos e os elementos de projetos que permita a rastreabilidade e controle de mudanças.

A infraestrutura geral criada, mostrando a viabilidade de criação de ambientes de MDO e de MBSE integrados através de configuração, codificação e uso do padrão XMI, se beneficiando de ambas abordagens.

6 CONSIDERAÇÕES FINAIS

Neste capítulo são apresentados uma lista de contribuições e de futuros trabalhos, além de considerações / lições aprendidas.

6.1 Conclusões

Este trabalho apresentou uma abordagem conjunta entre Engenharia de Sistemas Baseada em Modelos e Otimização de Projeto Multidisciplinar que envolveu algumas ferramentas e algoritmos existentes. A modelagem foi implementada usando a linguagem SysML para a descrição estrutural através da ferramenta IBM Rational Rhapsody como diagramador. Esta última demonstrou-se interoperável com a ferramenta DOORS de Gestão de Requisitos maximizando reuso de artefatos e evitando retrabalhos de inclusão de todos os requisitos novamente. Uma configuração de integração das ferramentas da IBM foi realizada para o estudo de caso no projeto da PMM. A interface de integração é versátil pois mantém e garante a rastreabilidade dos requisitos e modelos. Adicionalmente, ela permite a importação de modelos e exportação dos requisitos automatizando alguns processos da MBSE. Com MBSE, o uso dos outros diagramas provê melhor entendimento do sistema e possibilita melhorar as especificações em equipes multidisciplinares. Porém isto pode gerar uma resistência cultural por alguns engenheiros com relação à linguagem SysML. Este trabalho utilizou somente o diagrama de blocos pois os outros diagramas não foram necessários neste momento e no escopo estudado. A execução da otimização foi realizada via algoritmo genético MGEO e consistiu no layout bidimensional de parte dos equipamentos da PMM, uma vez que o foco deste é na validação do método proposto.

No processo de otimização foram definidas três funções-objetivo, relativas a aspectos de projeto, a saber: (1) Centro de Massa; (2) Uniformidade do fluxo de calor dissipado e (3) co-localização de equipamentos de um subsistema. O protótipo da infraestrutura criada foi concluído com êxito e mostra que é possível a criação de aplicações mais sofisticadas onde modelos SysML podem ser usados como entrada padrão para outras ferramentas dentro de um ambiente de otimização multidisciplinar como em projeto conceituais de satélite. Como a metodologia empregada pode ser estendida para uma grande variedade de problemas similares, espera-se que esta abordagem seja internalizada aos processos da Engenharia de Sistemas Espaciais do INPE.

6.2 Principais Contribuições

(1) Foi feito um estudo adicional para a implementação do código de programação necessário para a construção da infra-estrutura e para a configuração das ferramentas usadas;

(2) Ao software de pesquisa Satbudgets foi acrescentada uma nova funcionalidade, a de otimização de *layout* de equipamentos, que é consistente com o objetivo para o qual foi criado, ajudar os projetistas nas tomadas de decisão na fase conceitual do projeto de satélites através de automatização. Desta forma, o desenvolvedor/projetista pode usar uma única interface, a do SatBudgets, para obter vários processamentos.

(3) A modificação de uma aplicação standalone para uma aplicação Web, permite que o aplicativo seja usado em diversas máquinas através da rede, bastante útil em equipes fisicamente separadas. Infelizmente neste caso, não é possível que esta modificação do Satbudgets seja usada em qualquer máquina, por 2 motivos: A ferramenta Solidworks só funciona em ambientes Windows, e precisa estar instalada na máquina que será usada. Além disso, toda a infraestrutura de software deve ser adaptada ao ambiente de software/hardware.

(4) A criação da arquitetura de software, interoperando várias ferramentas teve como requisito a reusabilidade, tanto com relação às ferramentas/softwarewares já existentes quanto as bibliotecas de código. Desta forma o trabalho realizado permitirá que extensões e novas ferramentas sejam adicionadas à infraestrutura de software.

(5) O assunto tratado é muito relevante, pois é um problema que atinge não somente a área espacial, como a integração de softwares, automação industrial, entre outras.

6.3 Trabalhos Futuros

Alguns tópicos apresentados na Fundamentação Teórica embora não tenham sido completamente utilizados apresentam assunto relacionado diretamente ao trabalho e poderão trazer alternativas para futuras extensões do protótipo.

- O Padrão STEP, apesar de não ter sido usado diretamente neste trabalho, pode ser bastante útil se forem acopladas outras ferramentas de software de engenharia específica que usem este padrão, desta forma permitindo a comunicação entre as mesmas de forma mais fácil;
- A arquitetura SOA, é muito usada atualmente no desenvolvimento de soft-

ware, através de implementação de Web Services, permitindo a reusabilidade de código através das suas interfaces definidas e do uso de suas funcionalidades através da Web. No caso deste trabalho, haviam sido desenvolvidos alguns Web Services para algumas funcionalidades, mas com o uso de ferramentas CASE de uso local, o uso de SOA foi cancelado, mas pode ser implementado para a futura criação de ambientes que necessitem de colaboração via Web;

- Os padrões criados pela OSLC parecem ser promissores para a comunicação entre ferramentas desenvolvidas por equipes e empresas diferentes. Caso este padrão seja largamente adotado, é possível que a comunicação entre ferramentas de software passem a ser feitas de modo mais fácil e rápido. Estes padrões ainda estão sendo desenvolvidos, e já existem algumas empresas/ferramentas que os utilizam, como é o caso de alguns softwares da família Rational da IBM e o software de controle de *bugs* JIRA Atlassian;
- Os memorandos técnicos da ESA apresentados na 2.1, mostram brevemente o esforço da ESA em modernizar seu ambiente de engenharia simultânea, assim como a definição de modelos de trocas dados é de grande importância e a complexidade envolvida na criação de um ambiente deste porte. Futuramente, pois este novo ambiente ainda está em fase de definição/construção, poderá tornar-se uma grande referência para o desenvolvimento de sistemas espaciais;
- Uma implementação Web do ambiente poderia ser útil para facilitar o uso em equipes separadas fisicamente, ou somente para facilitar a instalação do ambiente, ou para dispor para stakeholders externos informações não restritas do projeto;
- Existe também uma nova versão em evolução do aplicativo de pesquisa PJLayout, ele possui várias melhorias, como, o uso de mais de 1 painel, a possibilidade de equipamentos poderem ser posicionados em ambos os lados dos painéis e a implementação de outros algoritmos de otimização além do MGEO. Portanto seria vantajosa a inserção desta nova versão.
- Este trabalho também poderá contribuir como um roteiro para futuros trabalhos em interoperabilidade.

REFERÊNCIAS BIBLIOGRÁFICAS

AGTE, J.; WECK, O. L. de; SOBIESZCZANSKI-SOBIESKI, J.; ARENDSSEN, P.; MORRIS, A.; SPIECK, M. Mdo: Assessment and direction for advancement: an opinion of one international group. **Structural and Multidisciplinary Optmization**, v. 40, n. 1, p. 17–33, 2010. 12, 38

BAIER H.; PUEHLHOFER, T. Z. C. Approaches for further rationalisation in mechanical architecture and structural design of satellites. In: INTERNATIONAL ASTRONAUTICAL CONGRESS, 2003, Bremen, Germany. **Proceedings...** Bremen, 2003. 34

BANDECCHI, M.; MELTON, B.; ONGARO, F. Concurrent engineering applied to apace mission assessment and design. **ESA Bulletin**, v. 5, 1999. 3, 28, 29

BEZIVIN, J.; GERBE, O. Towards a precise definition of the omg/mda framework. In: ANNUAL INTERNATIONAL CONFERENCE ON AUTOMATED SOFTWARE ENGINEERING, 16., 2001, San Diego, USA. **Proceedings...** USA, 2001. p. 273–280. 15, 16

CORSETTI, A.; RIBEIRO, E. A.; GARBI, G. P.; ZANTA, K. M.; MEDEIROS, M.; LOUREIRO, G. Complex systems developed with system concurrent engineering. In: ISPE INTERNATIONAL CONFERENCE ON CONCURRENT ENGINEERING, 19., 2012, Trier, Germany. **Proceedings...** Springer, 2012. v. 2, p. 1057–1068. Disponível em:
<<http://plutao.sid.inpe.br/dpi.inpe.br/plutao/2012/11.28.15.26>>. Acesso em: 20 abr. 2014. 137

CUCO, A. P. C. **Desenvolvimento de uma metodologia multiobjetivo para a otimização do layout de equipamentos em satélites artificiais**. 157 p. Dissertação (Mestrado) — Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2011-09-08 2011. Disponível em:
<<http://urlib.net/sid.inpe.br/mtc-m19/2011/08.09.01.46>>. Acesso em: 14 fev. 2013. 32, 45, 47, 90, 91, 102

DASSAULT SYSTEMES. **CAD Library**. 2014. Disponível em:
<<http://www.solidworks.com/sw/products/3d-cad/cad-library.htm>>. Acesso em: 03 fevereiro 2014. 58

DE SOUSA, F. L.; GALSKI, L. R.; ROCCO, E. M.; BECCENERI, J. C.; DOS SANTOS, W. A.; SANDRI, S. A. A tool for multidisciplinary design conception of

spacecraft equipment layout. In: INTERNATIONAL CONGRESS OF MECHANICAL ENGINEERING, 2013, Ribeirão Preto, Brazil. **Proceedings...** Brazil: Department of Applied Information Technology, 2013. 1, 5, 57, 58, 59, 60, 61, 102

ECLIPSE FOUNDATION. **Site oficial do Eclipse para download**. 2013. Disponível em: <<http://www.eclipse.org/downloads/moreinfo/java.php>>. Acesso em: 01 junho 2013. 64

EUROPEAN COOPERATION FOR SPACE STANDARDIZATION. **ECSS-TM-10-25**: Engineering design model data exchange (cdf). [S.l.], 2010. 29
_____. **ECSS-TM-10-23**: Space system data repository. [S.l.], 2011. 30, 31

EUROPEAN SPACE AGENCY. **Description of the Concurrent Design Facility (CDF)**. 2012. Disponível em: <<http://www.esa.int/SPECIALS/CDF/>>. Acesso em: 24 julho 2012. 1

FAVRE, J.-m. Towards a basic theory to model driven engineering. In: UML2004 INT. WORKSHOP ON SOFTWARE MODEL ENGINEERING, 2004. **Proceedings...** [S.l.]: Department of Applied Information Technology, 2004. 16

FORTESCUE, P.; STARK, J. **Spacecraft Systems Engineering**. [S.l.]: Wiley, 2001. 38, 43, 44

FRIEDENTHAL, S. **A practical guide to SysML**. Burlington, MA, USA: The OMK/OMG PRESS, 2009. 15, 16

GALSKI, R. L. **Desenvolvimento de versões aprimoradas híbridas, paralela e multiobjetivo do método da otimização extrema generalizada e sua aplicação no projeto de sistemas espaciais**. 279 p. Tese (Doutorado) — Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2006-09-22 2006. 58

GERMAN AEROSPACE CENTER (DLR). **Description of the German Aerospace Center**. 2012. Disponível em: <<http://www.dlr.de/dlr/en/>>. Acesso em: 24 julho 2012. 1

HOLLEY, K.; ARSANJANI, A. **100 SOA questions**: Asked and answered. Indiana: Prentice Hall, 2010. 267p. 23

INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS (IEEE). **IEEE 1220**: Ieee standard for application and management of the systems

engineering process. 1998. Disponível em:

<<http://standards.ieee.org/findstds/standard/1220-1998.html>>. Acesso em: 11 fev. 2014. 9

INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS (INPE). **Multi-Mission Platform Specification**. São José dos Campos, Brazil, 2001. (INPE-10394-TDI/920). 46

_____. **Multimission Platform Data Package For System Requirements Review (SRR)**. São José dos Campos, Brazil, 2001. (INPE-10394-TDI/920). 67, 72, 76

INTERNATIONAL BUSINESS MACHINES CORPORATION. **Rational Rhapsody Getting Started User Guide**. 2014. Disponível em:

<http://publib.boulder.ibm.com/infocenter/rsdp/v1r0m0/topic/com.ibm.help.download.rhapsody.doc/pdf75/Getting_Started_Guide.pdf>. Acesso em: 03 fevereiro 2014. 5, 56

_____. **Rhapsody Coupling Notes**. 2014. Disponível em:

<http://publib.boulder.ibm.com/infocenter/rsdp/v1r0m0/topic/com.ibm.help.download.rhapsody.doc/pdf75/Getting_Started_Guide.pdf>. Acesso em: 03 fevereiro 2014. 56

INTERNATIONAL BUSINESS MACHINES CORPORATION (IBM). **Using Rational DOORS**. 2014. Disponível em:

<http://publib.boulder.ibm.com/infocenter/rsdp/v1r0m0/topic/com.ibm.help.download.doors.doc/pdf92/using_doors.pdf>. Acesso em: 03 fevereiro 2014. 5, 56

JFREE.ORG. **Site oficial do JFreeChart para download**. 2013. Disponível em: <<http://www.jfree.org/jfreechart/>>. Acesso em: 01 junho 2013. 64

KLEPPE, A.; WARMER, S.; BAST, W. **MDA explained. The Model driven architecture: practice and promise**. [S.l.]: Addison-Wesley, 2003. 16

KORTE, J. J.; WESTON, R. P.; A., Z. T. Multidisciplinary optimization methods for preliminary design. **The AGARD Symposium on "Future Aerospace Technology in the Service of the Alliance"** Ecole Polytechnique, AGARD-CP-600, v. 3, p. C-40:1-10, April 1997. 14

LARSON, W. J.; WERTZ, J. R. **Space mission analysis and design**.

California: Space Technology Library, 1999. 3 ed. 4, 10, 11, 39, 44, 82, 84, 85, 86

LEONOR, B. B. F. **Um enfoque baseado em conhecimento e dirigido a modelos de engenharia de requisitos para projeto conceitual de satélites.** Dissertação (Mestrado) — Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2010-02-23 2010. 5, 32, 33, 57, 58

LIU, A.; GORTON, I. Process and criteria for evaluating services-based integration technologies. **The Architecture Journal**, 2012. 24

LOUREIRO, G. **A systems engineering and concurrent engineering framework for the integrated development of complex products.** Thesis (Doctoral) — Loughborough University, Leicestershire, 1999. 9

MARTINS, J. R. R. A.; LAMBEY, A. B. Multidisciplinary design optimization: A survey of architectures. **AIAA**, v. 51, 2013. 1, 12

MEDEIROS, M.; BASTOS, A. B.; BARBOSA, C. A. M.; SIMÕES, M. C.; SANTOS, W. A. D. Exploração do espaço de projeto de satélites para um cenário reduzido de análise. In: SIMPÓSIO AEROESPACIAL BRASILEIRO, 1. (SAB), 2012, São José dos Campos-SP. **Anais...** 2012. Disponível em: <<http://urlib.net/dpi.inpe.br/plutao/2012/06.21.18.35>>. Acesso em: 12 fev. 2014. 139

MEDEIROS, M.; SANTOS, W. A. dos. Labe3s: A prototype of an experimental concurrent design facility for space systems engineering. In: WORKSHOP EM ENGENHARIA E TECNOLOGIA ESPACIAIS, 3. (WETE), 2012, São José dos Campos. **Anais...** São José dos Campos: Instituto Nacional de Pesquisas Espaciais (INPE), 2012. Disponível em: <<http://urlib.net/dpi.inpe.br/plutao/2012/06.21.18.36.25>>. Acesso em: 12 fev. 2014. 141

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION (NASA). **Description of the Team X.** 2012. Disponível em: <<http://jplteamx.jpl.nasa.gov/>>. Acesso em: 24 julho 2012. 1

OBJECT MANAGEMENT GROUP. **Unified Modeling Language Specification.** 2013. Disponível em: <<http://www.omg.org/spec/UML/>>. Acesso em: 20 fevereiro 2013. 2

_____. **What is OMG SysML?** 2013. Disponível em: <<http://www.omgsysml.org/>>. Acesso em: 20 fevereiro 2013. 2, 18

ORACLE. **The Java EE 5 Tutorial**. 2013. Disponível em:
<<http://docs.oracle.com/javaee/5/tutorial/doc/bnagx.html>>. Acesso em:
01 junho 2013. 64

_____. **The Java EE 6 Tutorial**. 2013. Disponível em:
<<http://docs.oracle.com/javaee/6/tutorial/doc/bnafnd.html>>. Acesso em:
01 junho 2013. 64

OSLC WORKGROUPS. **OSLC Primer - Learning the concepts of OSLC**.
2012. Disponível em:
<<http://open-services.net/resources/tutorials/oslc-primer/>>. Acesso
em: 24 julho 2012. 25, 27

PRESSMAN, R. S. **Engenharia de software**. Indiana: Prentice Hall, 2011. 776p.
22

ROSEN, M.; LUBLINSKY, B.; SMITH, K. T.; BALCER, M. **Applied SOA:**
Service-oriented architecture and design strategies. Indiana: Wiley Publishing,
2008. 23, 24

SCHMIDT, D. C. Guest editor's introduction: Model-driven engineering.
Computer, v. 39, n. 2, p. 0025–31, 2006. IEEE Computer Society. 1

SEIDEWITZ, E. What models mean. **Software, IEEE**, v. 20, n. 5, p. 26–32,
2003. ISSN 0740-7459. 16

SILVA JÚNIOR, A. C. **Desenvolvimento integrado de sistemas espaciais -
design for AIT- projeto para a montagem, integração e teste de satélites
D4AIT**. Tese (Doutorado em Engenharia de Produção) — Instituto Tecnológico
de Aeronáutica, São José dos Campos, 2011. 41

SOBIESZCZANSKI-SOBIESKI, J. Multidisciplinary design optimization: An
emerging new engineering discipline. In: HERSKOVITS, J. (Ed.). **Advances in
Structural Optimization**. Springer Netherlands, 1995, (Solid Mechanics and Its
Applications, v. 25). p. 483–496. ISBN 978-94-010-4203-1. Disponível em:
<http://dx.doi.org/10.1007/978-94-011-0453-1_14>. 12

SOBIESZCZANSKI-SOBIESKI, J.; HAFTKA, R. Multidisciplinary aerospace
design optimization: survey of recent developments. **Structural optimization**,
Springer-Verlag, v. 14, n. 1, p. 1–23, 1997. ISSN 0934-4373. Disponível em:
<<http://dx.doi.org/10.1007/BF01197554>>. 12

SOMMERVILLE, I. **Engenharia de software**. [S.l.]: Pearson Education - Br, 2011. 16

SOUZA, A. C. C. d. **SpaceESB: um ambiente colaborativo para apoio ao projeto conceitual de satélites usando barramento corporativo de serviços**. 105 p. Dissertação (Mestrado) — Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2011-12-21 2011. 33, 34

SOUZA, P. N. **Missões e Segmentos**. São José dos Campos, SP, 2011. Unidade 1/ Parte 1.2/ Versão 4.2. 42

THE APACHE POI PROJECT. **Site oficial do Apache POI para download**. 2013. Disponível em: <<http://poi.apache.org/>>. Acesso em: 01 junho 2013. 64

THE APACHE SOFTWARE FOUNDATION. **Site oficial do Tomcat para download**. 2013. Disponível em: <<http://tomcat.apache.org/>>. Acesso em: 01 junho 2013. 64

WALL, S. D. **Use of Concurrent Engineering in Space Mission Design**. 2000. Disponível em: <<http://hdl.handle.net/2014/15133>>. Acesso em: 19 fevereiro 2013. 15

WALSH, N. A. **A Technical Introduction to XML**. 2012. Disponível em: <<http://www.w3.org/standards/xml/core>>. Acesso em: 19 setembro 2012. 20

WANG, Y.-S.; TENG, H.-F.; SHI, Y.-J. Cooperative co-evolutionary scatter search for satellite module layout design. **International Journal for Computer-Aided Engineering and Software**, v. 26, p. 761–785, 2009. 45

_____. Cooperative co-evolutionary scatter search for satellite module layout design. **Engineering Computations**, v. 26, p. 761–785, 2009. 46

WECK, O. de; WILLCOX, K. **Multidisciplinary System Design Optimization (MSDO)**. 14

WOLKL, S.; SHEA, K. A computational product model for conceptual design using sysml. In: INTERNATIONAL DESIGN ENGINEERING TECHNICAL CONFERENCES AND COMPUTERS AND INFORMATION IN ENGINEERING CONFERENCE, 29., 2009, San Diego, USA. **Proceedings...** USA: ASME, 2009. 17

WORLD WIDE WEB CONSORTIUM (W3C). **XML Essencials**. 2012. Disponível em:

<<http://www.xml.com/pub/a/98/10/guide0.html?page=2#AEN58>>. Acesso em:
19 setembro 2012. 20, 21

APÊNDICE A - UTILIZAÇÃO PASSO A PASSO DO FLUXO DE TRABALHO DO OPTMA

A seguir é descrito a sequência de passos para uso da estrutura de software provida pelo Optoma:

- a) No DOORS: Na tela de interface inicial deve ser criado um projeto
- b) No DOORS: Na tela de interface inicial deve ser criado, dentro do projeto, um módulo formal
- c) No DOORS: Na tela de interface do módulo formal, devem ser populados os requisitos, podendo ser especificada a hierarquia dos requisitos e adicionadas outros tipos de ligações entre requisitos
- d) No Rhapsody: Na tela de interface do Rhapsody, o usuário modela o diagrama de blocos com os equipamentos que deverão ser alocados no *layout*, com suas respectivas informações (massa, dimensões, etc) e de acordo com a linguagem SysML;
- e) No Rhapsody: O usuário salva o diagrama criado. Este diagrama é salvo no formato XMI, em um local indicado pelo usuário;
- f) No Rhapsody: Abrir a interface do Gateway a partir do projeto SysML criado
- g) No Gateway: Na interface inicial do Gateway devem ser criados os documentos que serão ligados, para isso cria-se os documentos, com informações do seu tipo, neste caso criar um documento do tipo DOORS. O documento SysML do Rhapsody já é criado automaticamente quando o Gateway é aberto.
- h) No Gateway: Quando os documentos são criados ele são adicionados graficamente em um modelo do Gateway, este modelo representa os documentos e as ligações que serão gerenciados por ele, portanto é importante adicionar um conector de ligação entre os documentos
- i) No Gateway: Após o modelo ser criado devem ser importados os requisitos do DOORS, isto é feito automaticamente selecionando adicionar requisitos de alto-nível na interface do Gateway.

- j) No Gateway: Após os requisitos serem importados, eles podem ser visualizados na interface do Gateway, também pode ser verificados quais requisitos estão "cobertos" no modelo SysML. A interface do Gateway possui várias funcionalidades que permite o gerenciamento da rastreabilidade entre os elementos destes documentos
- k) No Gateway: Após a configuração dos documentos, os requisitos podem ser exportados para a ferramenta Rhapsody
- l) No Rhapsody: Os requisitos exportados através do Gateway podem ser vistos agora no Rhapsody, representando cada um deles, um bloco do tipo requisitos, e é possível arrastá-los para qualquer diagrama e também associá-lo a qualquer objeto SysML, garantindo a rastreabilidade.
- m) No Satbudgets: O usuário seleciona o arquivo XML, salvo anteriormente, com o diagrama de blocos dos equipamentos.
- n) No Satbudgets: O usuário preenche os campos com as informações necessárias (customizáveis) para o funcionamento do MGEO.
- o) No Satbudgets: O usuário seleciona o botão executar otimização.
- p) No Satbudgets: Internamente é feito o parser do arquivo XML (através da API JDOM), e os dados extraídos são transformados em objetos java para poderem serem manipulados.
- q) No Satbudgets: Os dados extraídos do arquivo XML e os dados da interface de entrada são enviados para o aplicativo MGEO através da API Apache POI. Estes dados de entrada são automaticamente escritos nas planilhas de entrada do MGEO.
- r) No Satbudgets: É chamada a rotina VBScript, que serve de ponte de comunicação com o MGEO. Este VBScript chama a função de execução do MGEO que indica para dar início ao algoritmo de otimização. Neste ponto o Satbudgets fica em stand-by, monitorando o fim da execução do MGEO (para a sincronização).
- s) No MGEO: O aplicativo MGEO, faz o processamento do algoritmo e também usa algumas funções já implementadas do software SolidWorks. Isto é possível pela disponibilização de uma API, para a linguagem VB, pela empresa do SolidWorks

- t) No MGEO: Após o processamento do algoritmo de otimização, o algoritmo MGEO apresenta seus resultados em uma planilha do Excel em forma numérica. (Também é possível visualizar os resultados obtidos de forma gráfica na interface do Solidworks, para isso basta copiar manualmente os resultados obtidos para a planilha de entrada de dados dos equipamentos e executar a opção "Montagem" exibido na planilha Excel).
- u) No Satbudgets: O Satbudgets verifica que o MGEO já terminou sua execução e faz a leitura dos resultados da planilha do Excel usando novamente a API Apache POI.
- v) No Satbudgets: O SatBudgets gera um gráfico de bolhas, através da API JFreeChart, com os resultados obtidos e apresenta em uma página Web este gráfico em forma de imagem e os resultados em forma de tabela.
- w) No Satbudgets: O usuário visualiza os resultados da otimização. Nesta página também são apresentados links para cada uma das arquiteturas (configurações) resultantes, para que o usuário visualize a configuração gráfica da arquitetura solução.
- x) No Satbudgets: Quando o usuário seleciona em uma arquitetura (link). o Satbudgets passa as informações da arquitetura selecionada para o MGEO, novamente usando a API ApachePOI e chama a rotina VBScript responsável pela execução da funcionalidade de montagem do MGEO.
- y) No MGEO: O MGEO executa seu procedimento de montagem, usando a API do SolidWorks.
- z) No SolidWorks: Na tela do SolidWorks é apresentado o modelo gráfico da arquitetura/configuração selecionada.

APÊNDICE B - DIAGRAMAS DE REQUISITOS DA PMM

Uma parte significativa deste trabalho foi a elaboração dos diagramas de requisitos SysML, a partir dos requisitos populados na ferramenta IBM Rational DOORS e importados através do IBM Rational Rhapsody Gateway para a ferramenta de modelagem, SysML IBM Rational Rhapsody.

B.1 Diagramas de Requisitos da PMM

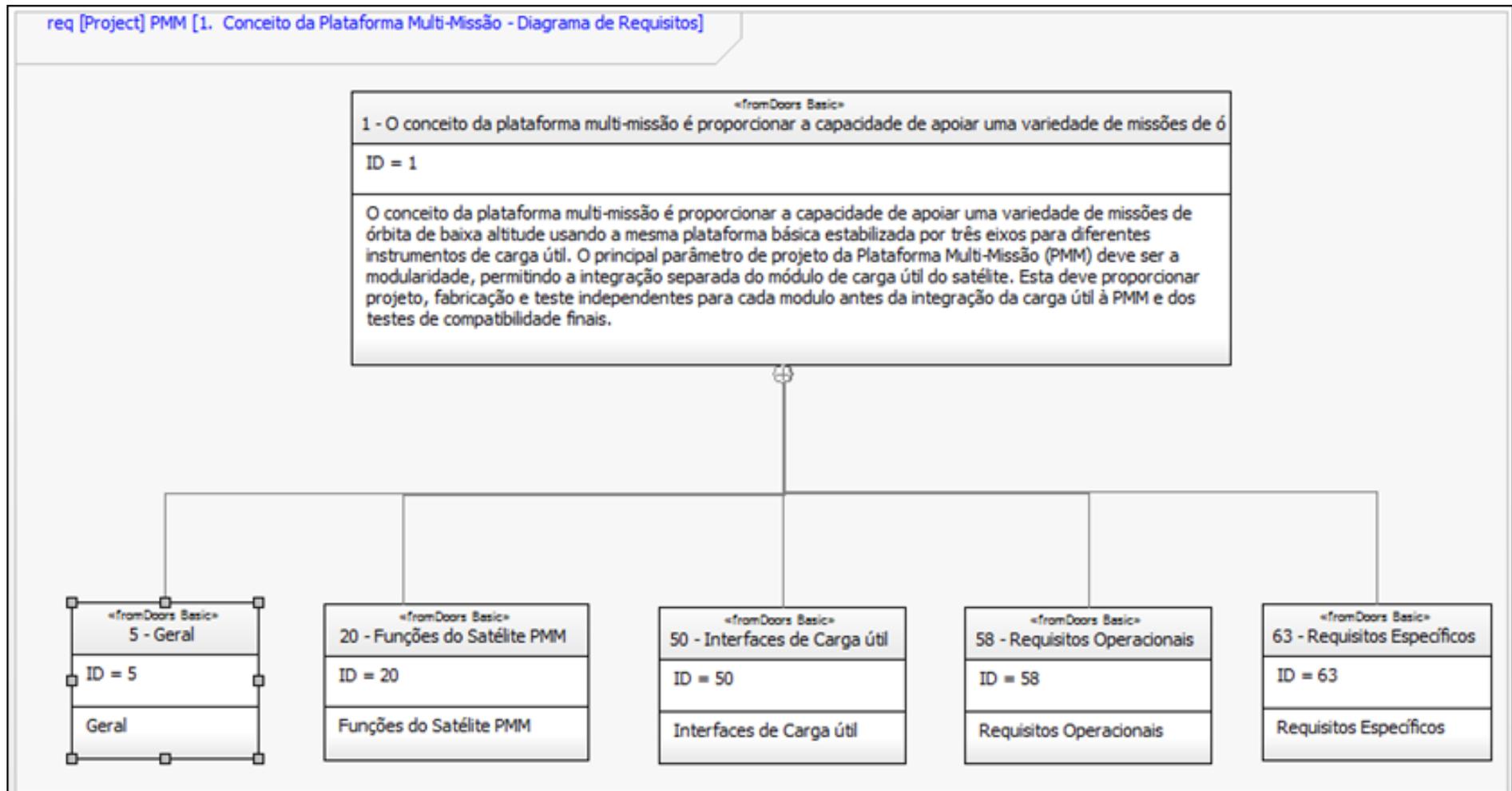


Figura B.1 - Decomposição da missão da PMM.

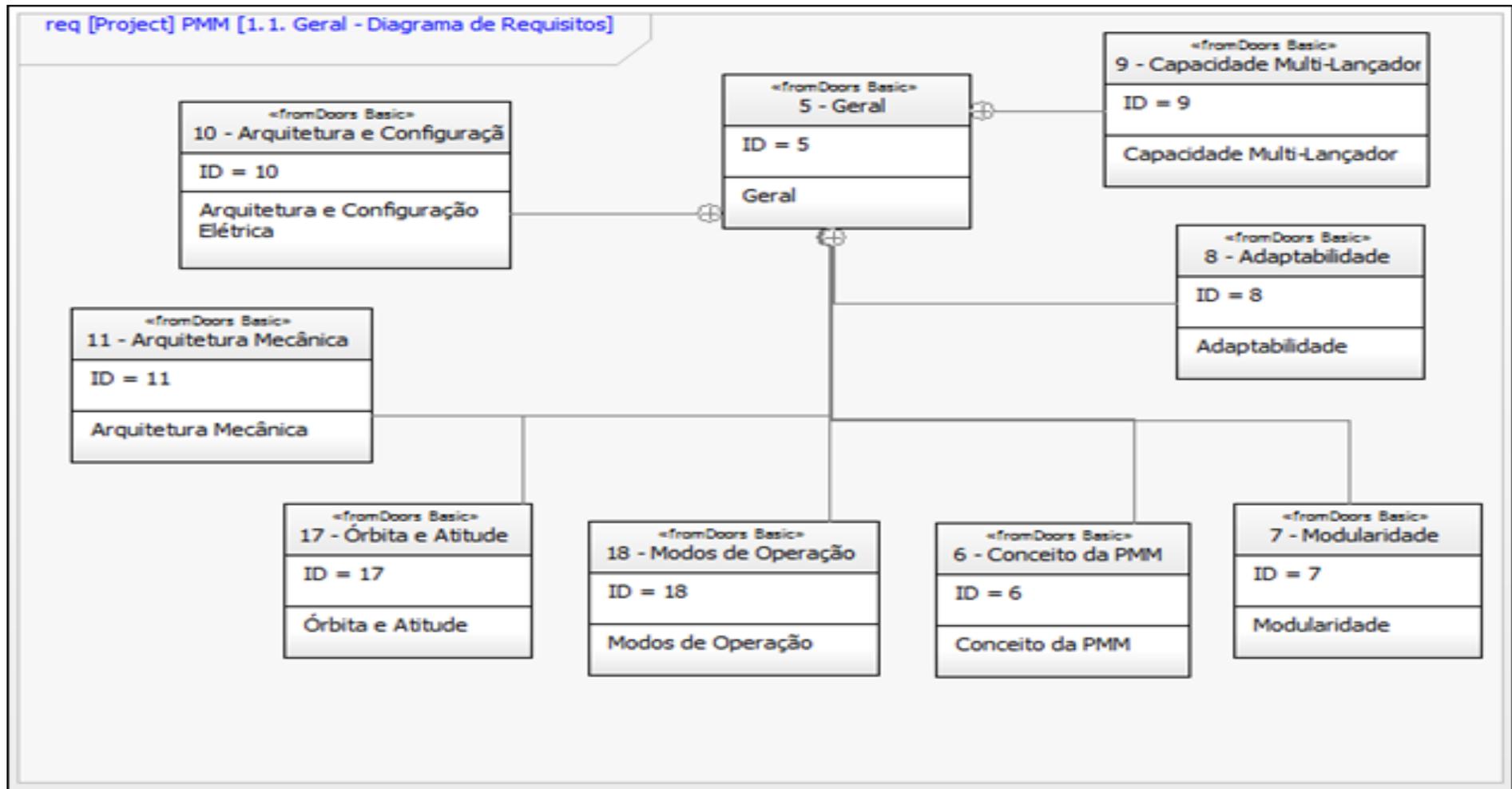


Figura B.2 - Decomposição dos Requisitos Gerais da PMM.

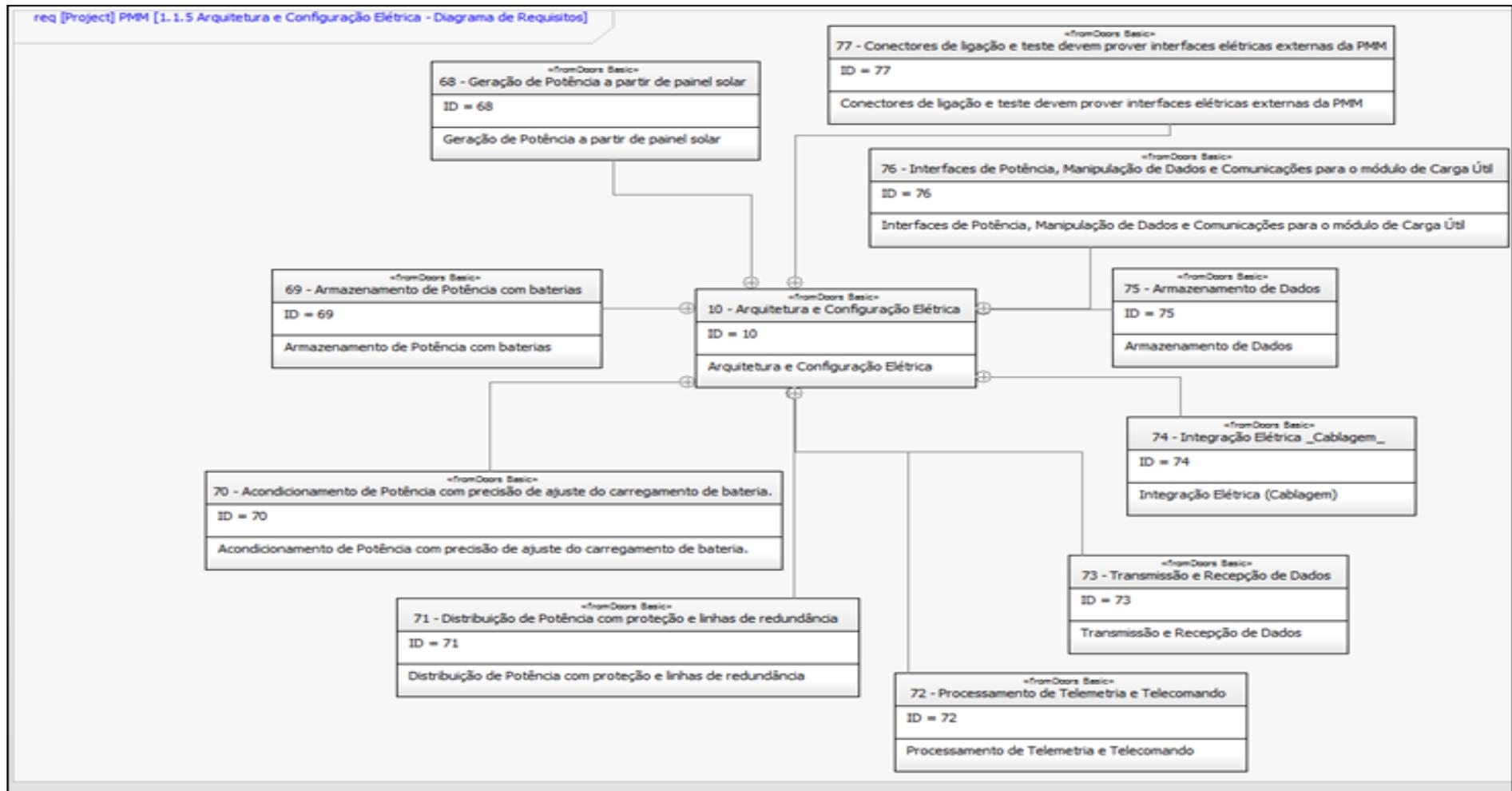


Figura B.3 - Decomposição dos Requisitos de Arquitetura e Configuração Elétrica da PMM.

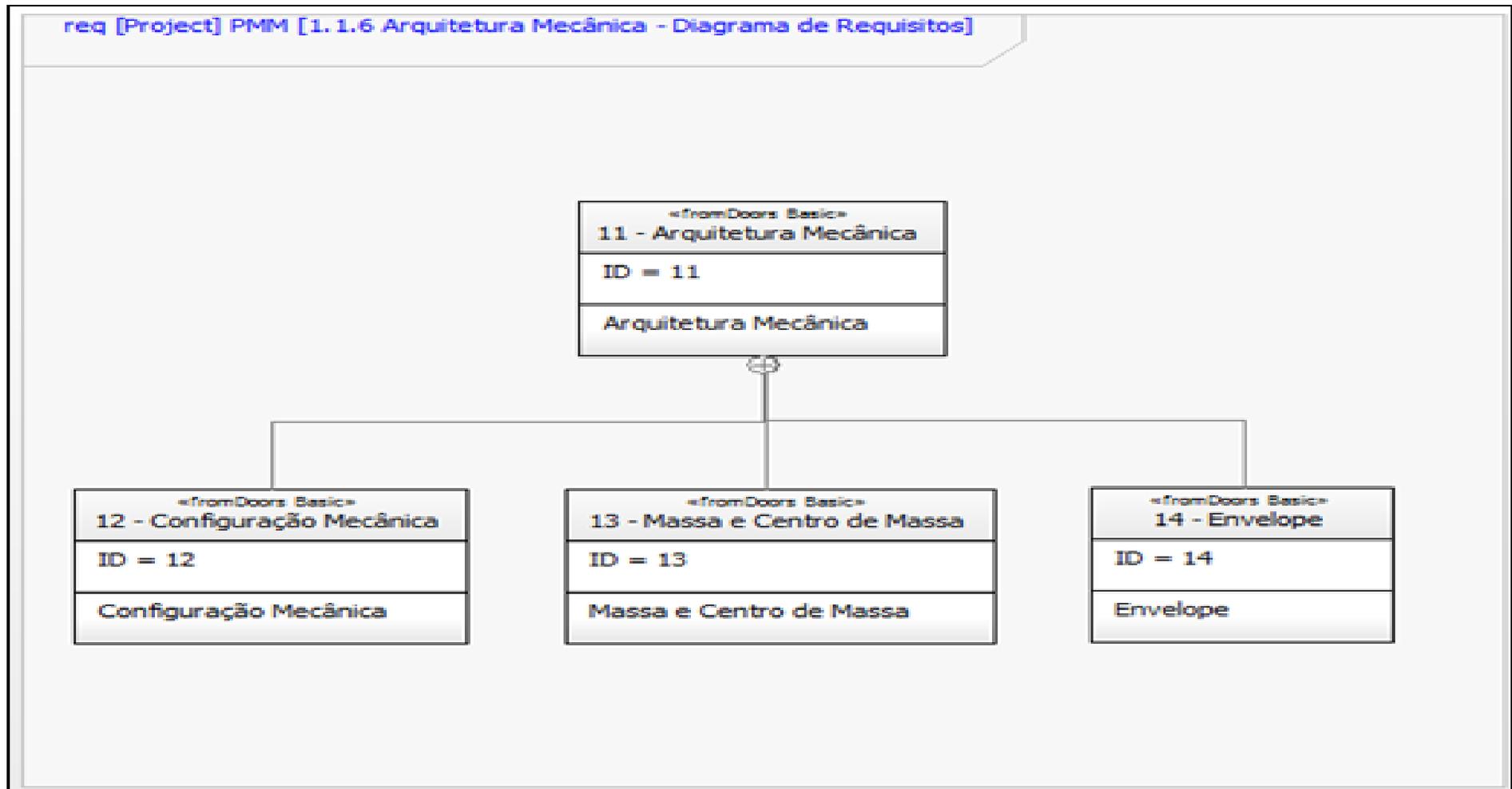


Figura B.4 - Decomposição dos Requisitos de Arquitetura Mecânica da PMM.

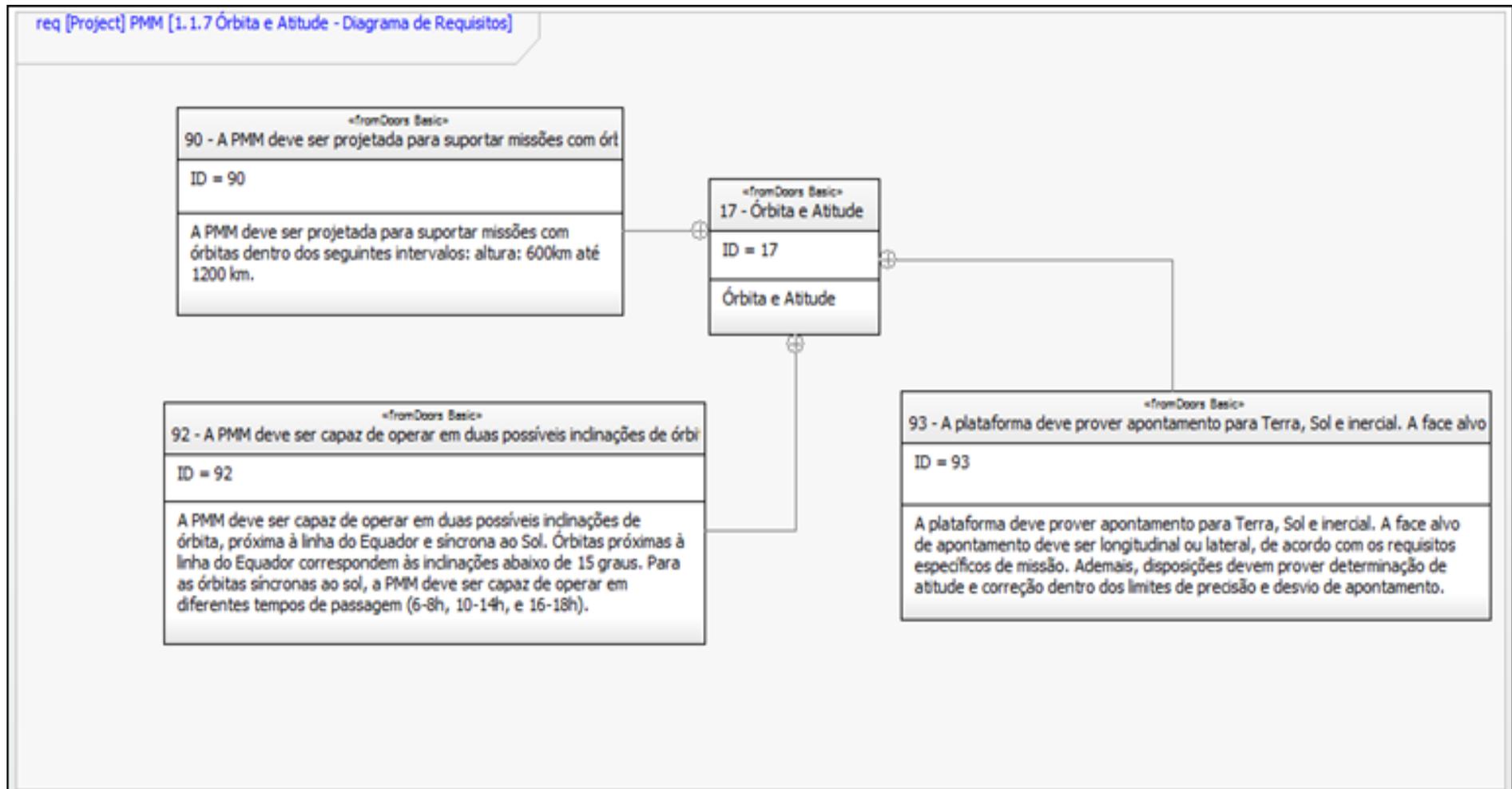


Figura B.5 - Decomposição dos Requisitos do Controle de Órbita e Atitude.

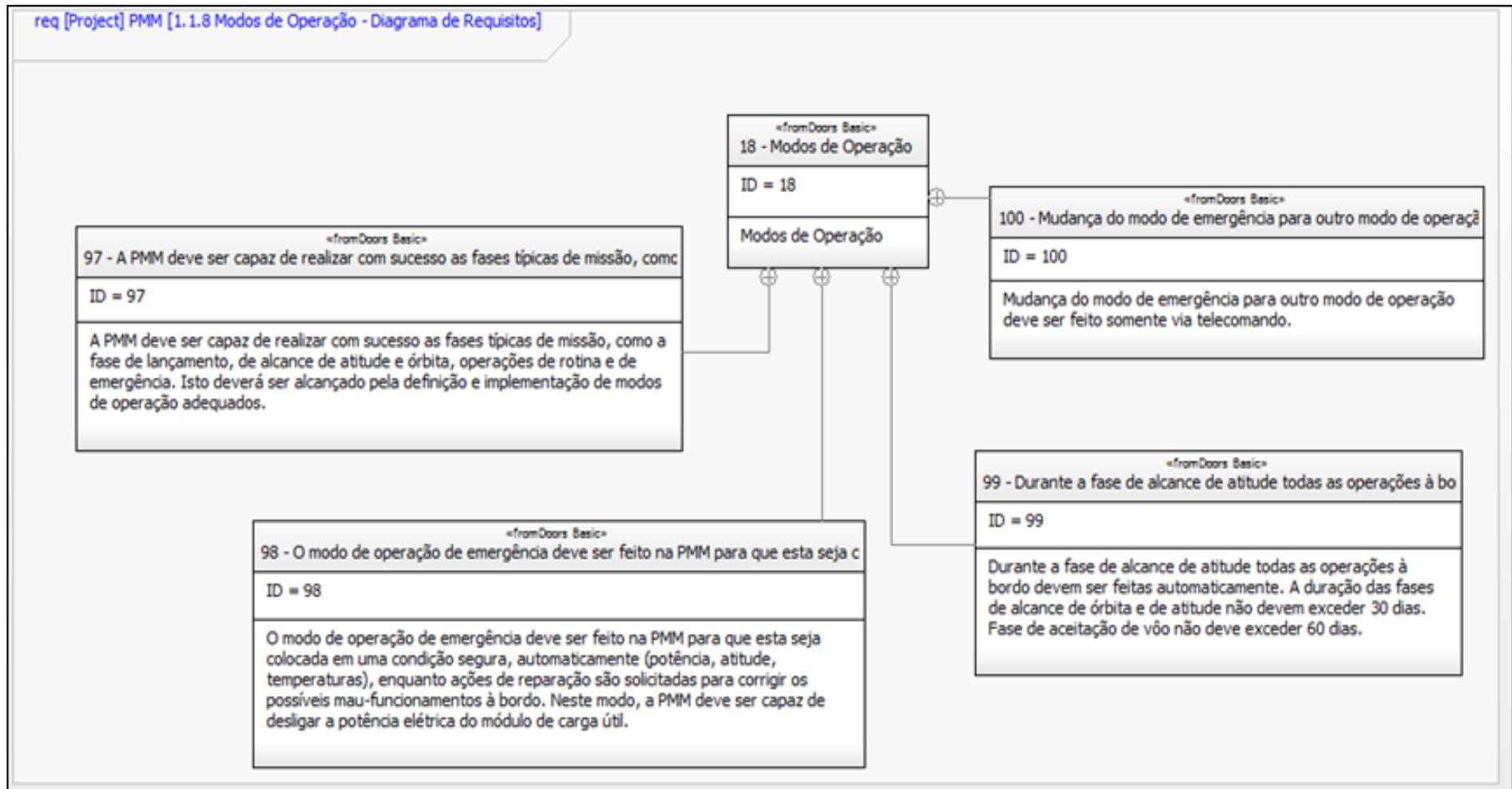


Figura B.6 - Decomposição dos Requisitos dos Modos de Operação.

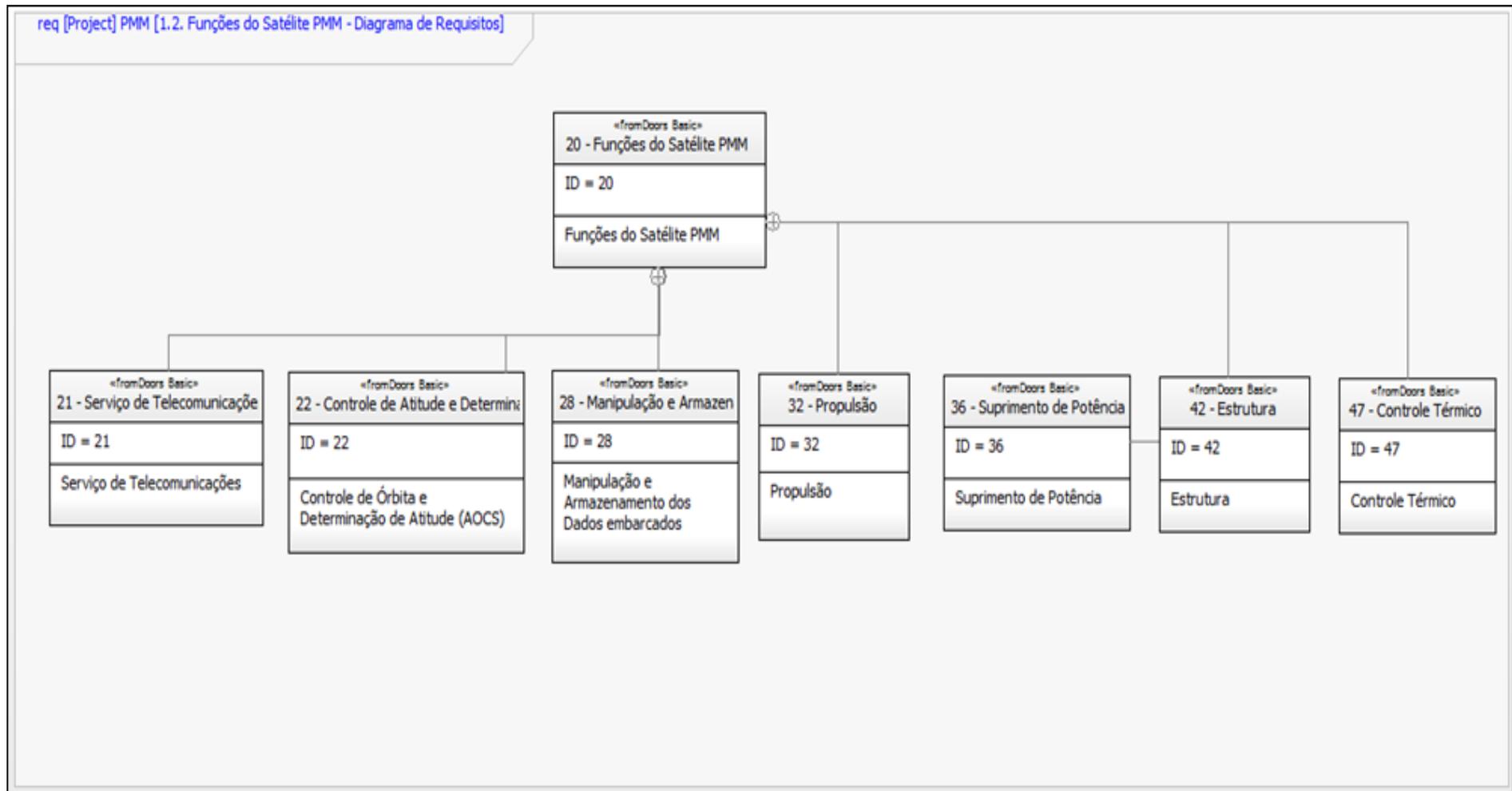


Figura B.7 - Decomposição dos Requisitos das Funções da PMM.

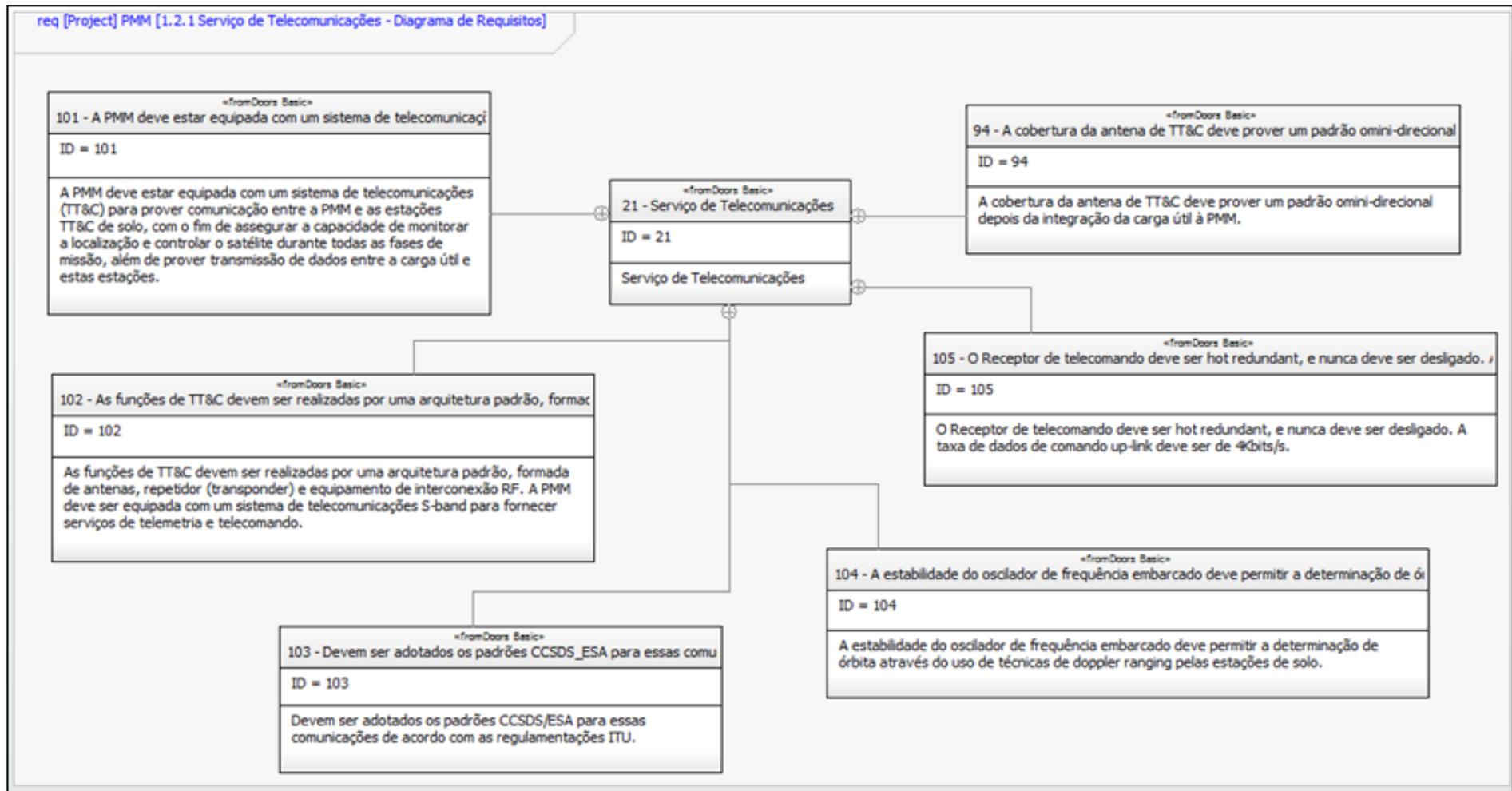


Figura B.8 - Decomposição dos Requisitos dos Serviços de Telecomunicações da PMM.

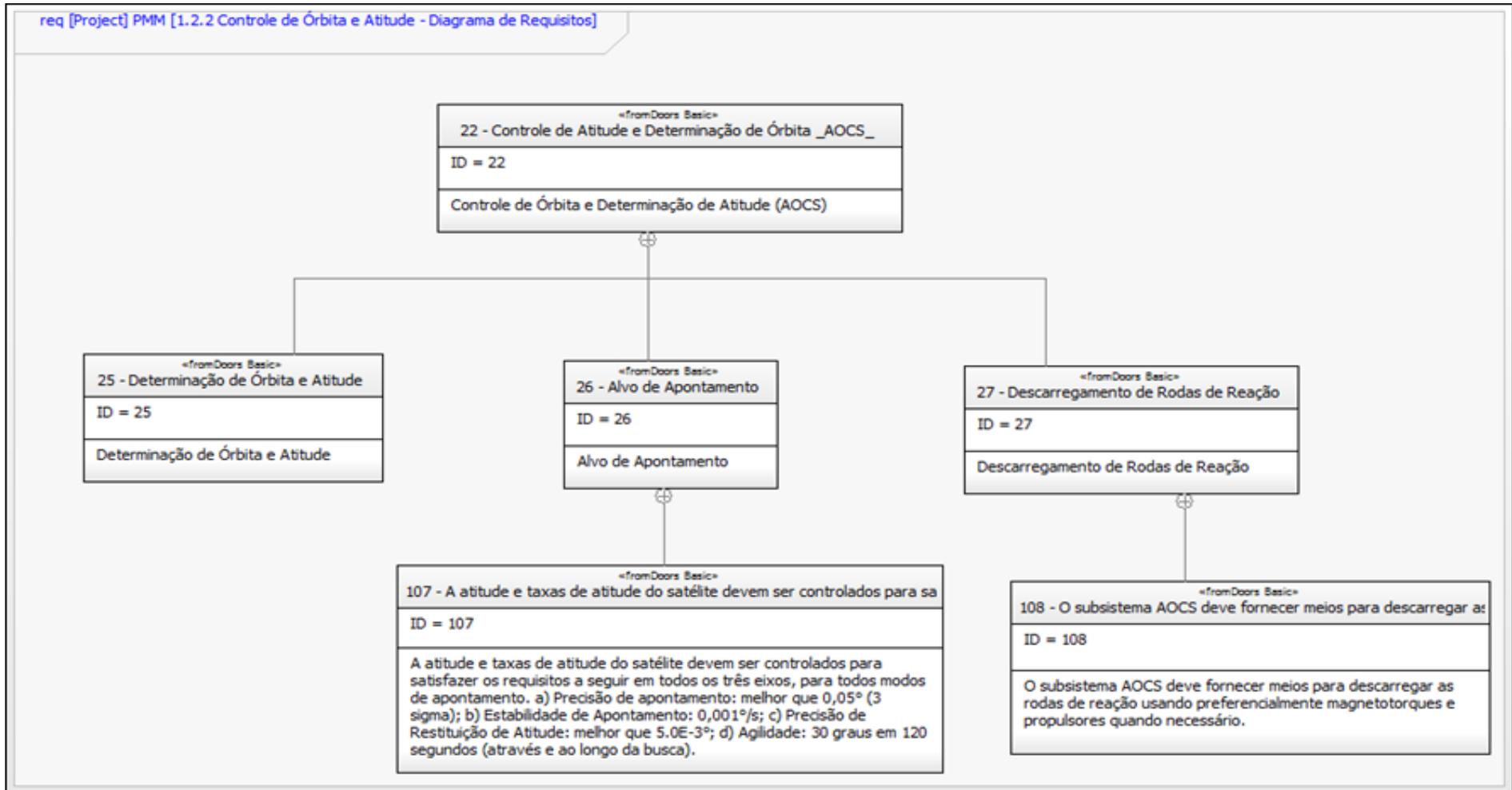


Figura B.9 - Decomposição dos Requisitos de Alvo de Apontamento da PMM.

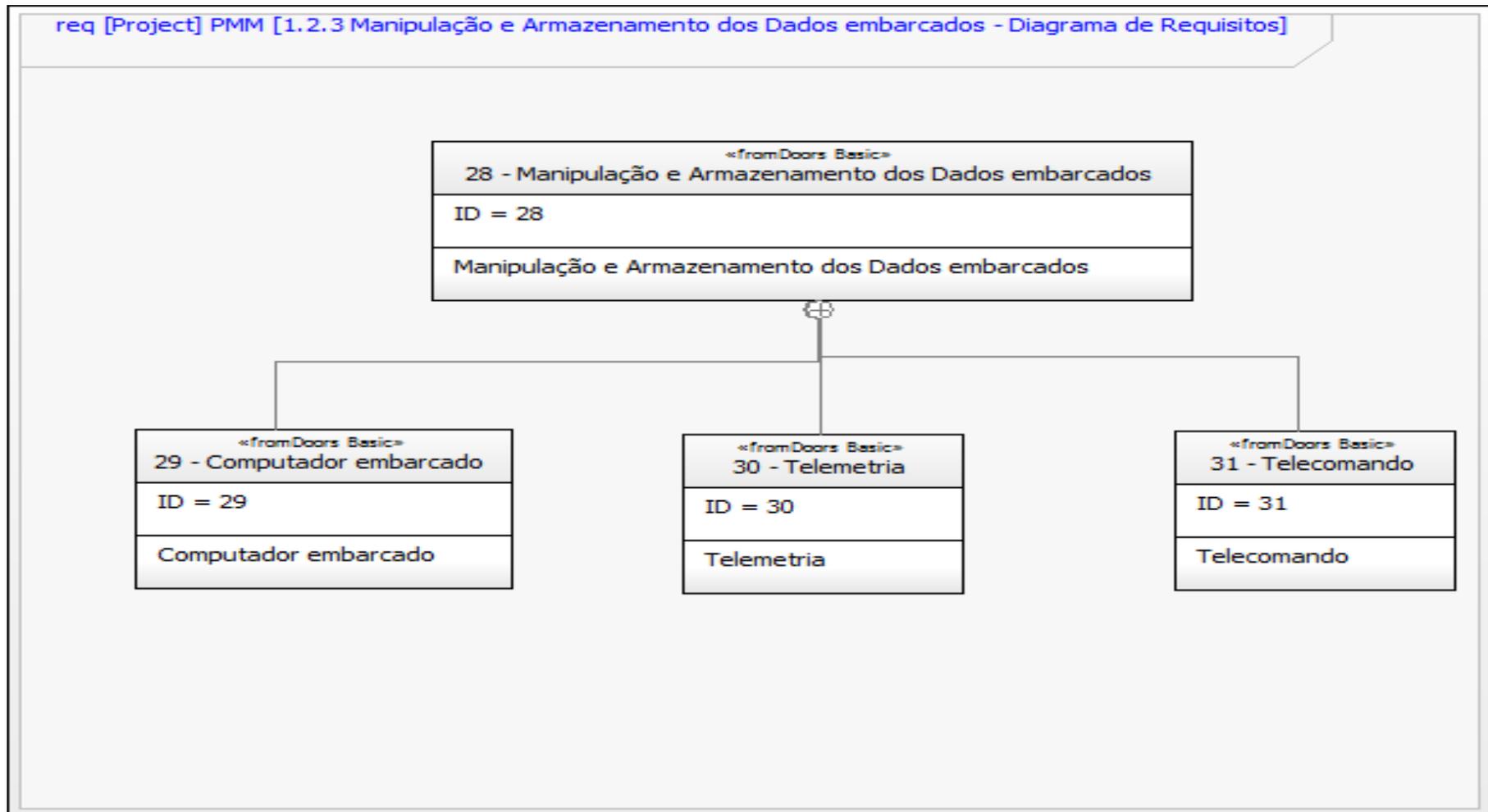


Figura B.10 - Decomposição dos Requisitos de Manipulação e Armazenamento dos Dados Embarcados da PMM.

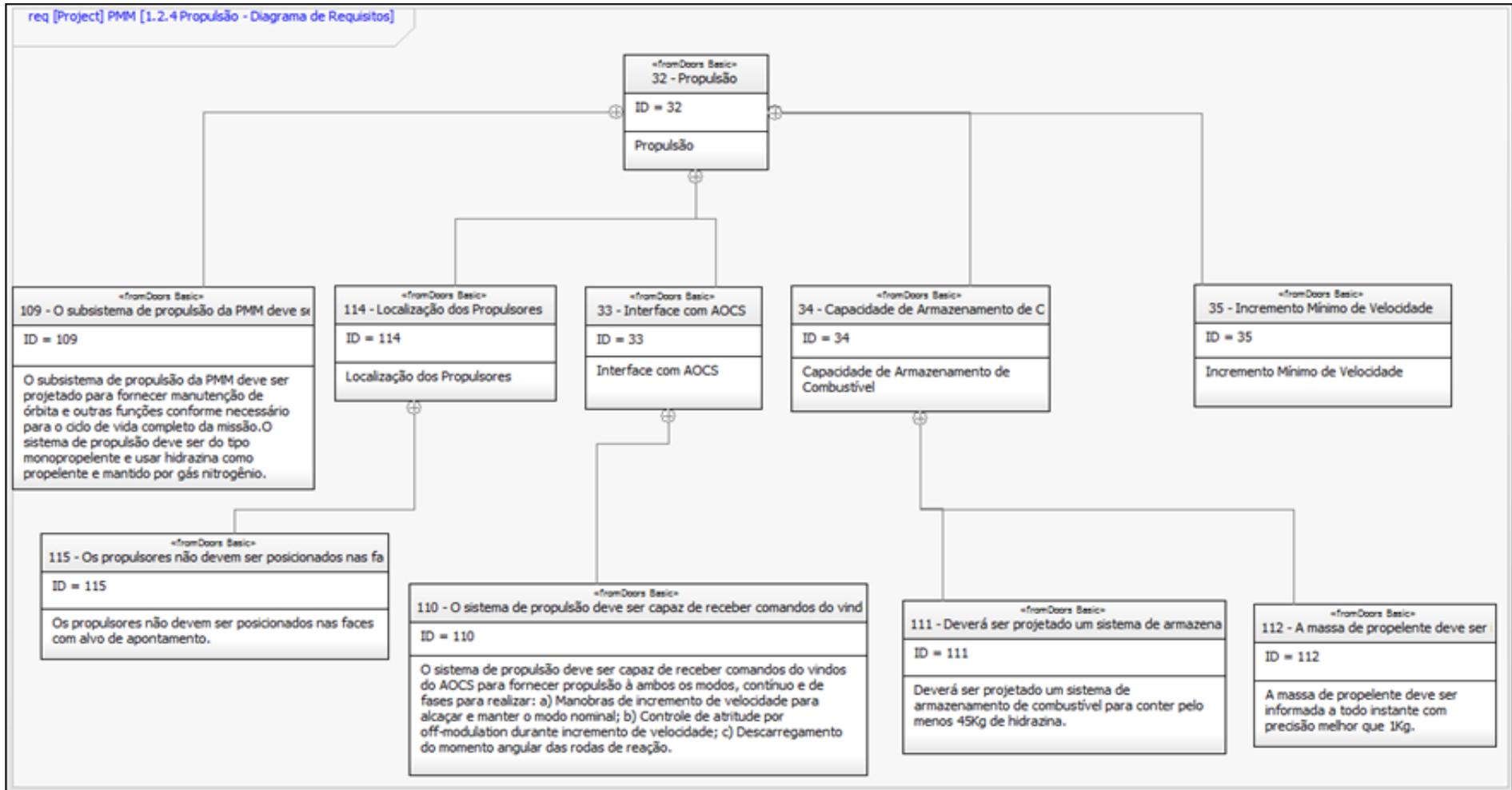


Figura B.11 - Decomposição dos Requisitos de Propulsão da PMM.

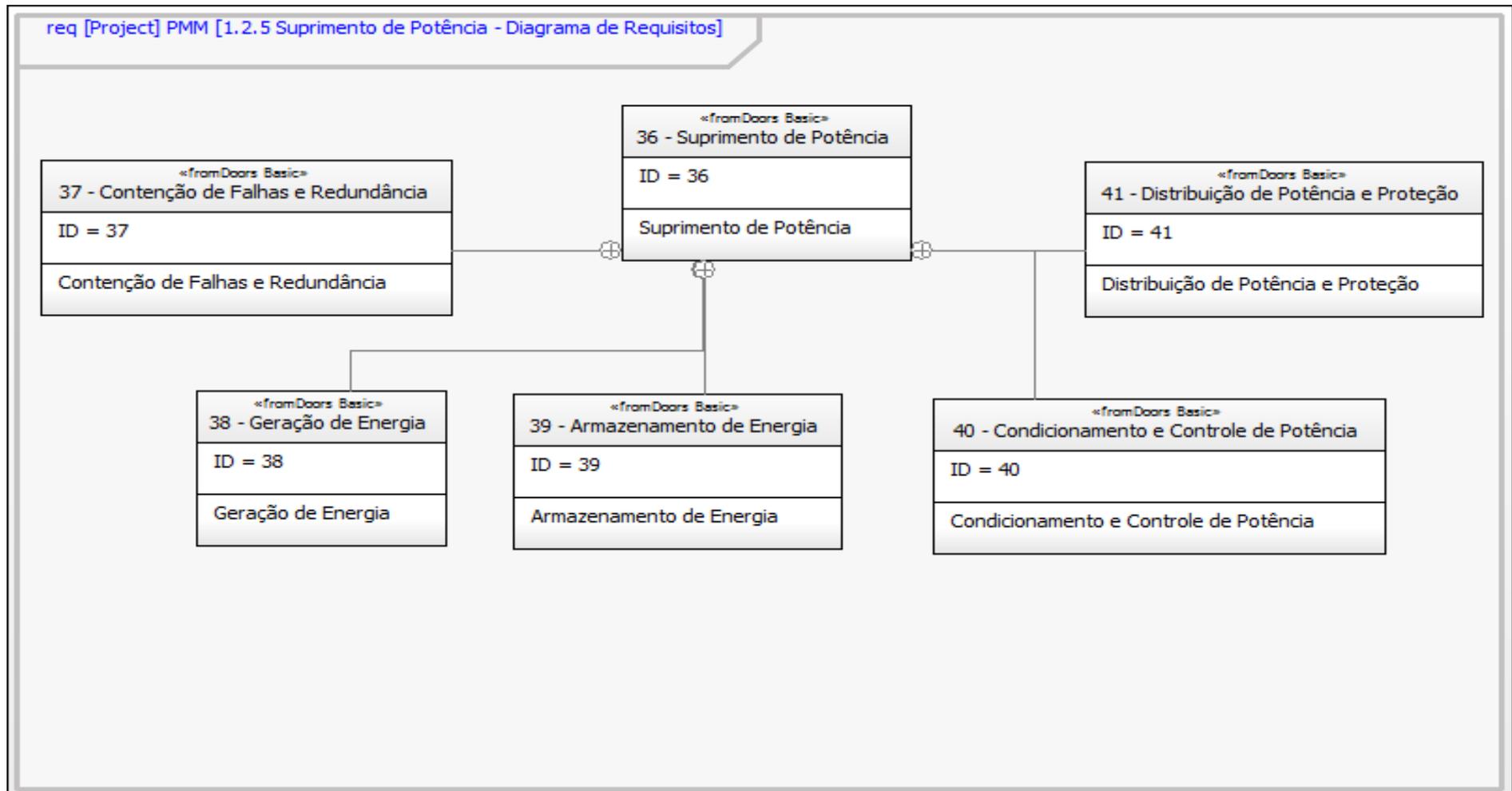


Figura B.12 - Decomposição dos Requisitos de Suprimento de Potência da PMM.

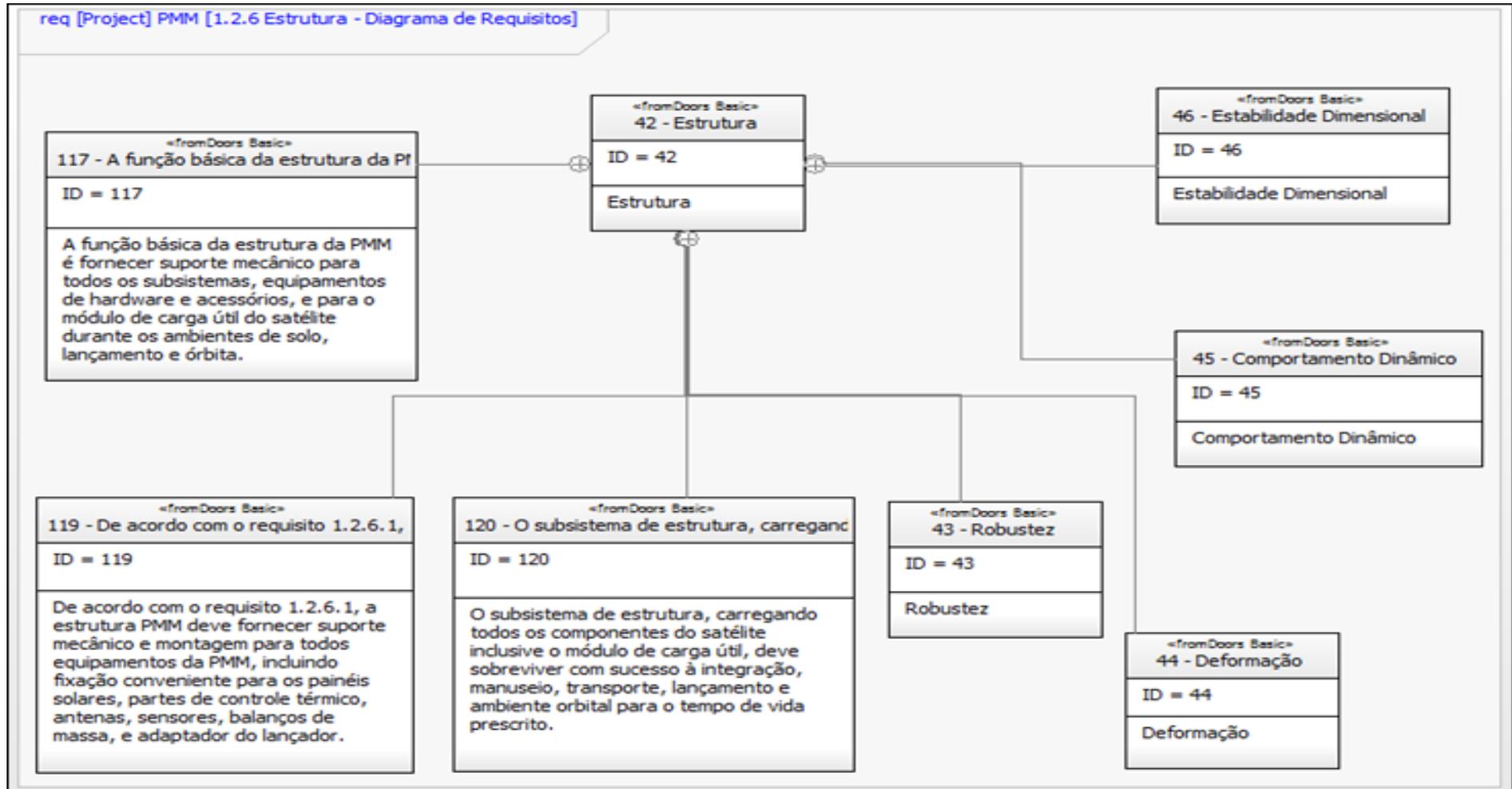


Figura B.13 - Decomposição dos Requisitos de Estrutura da PMM.

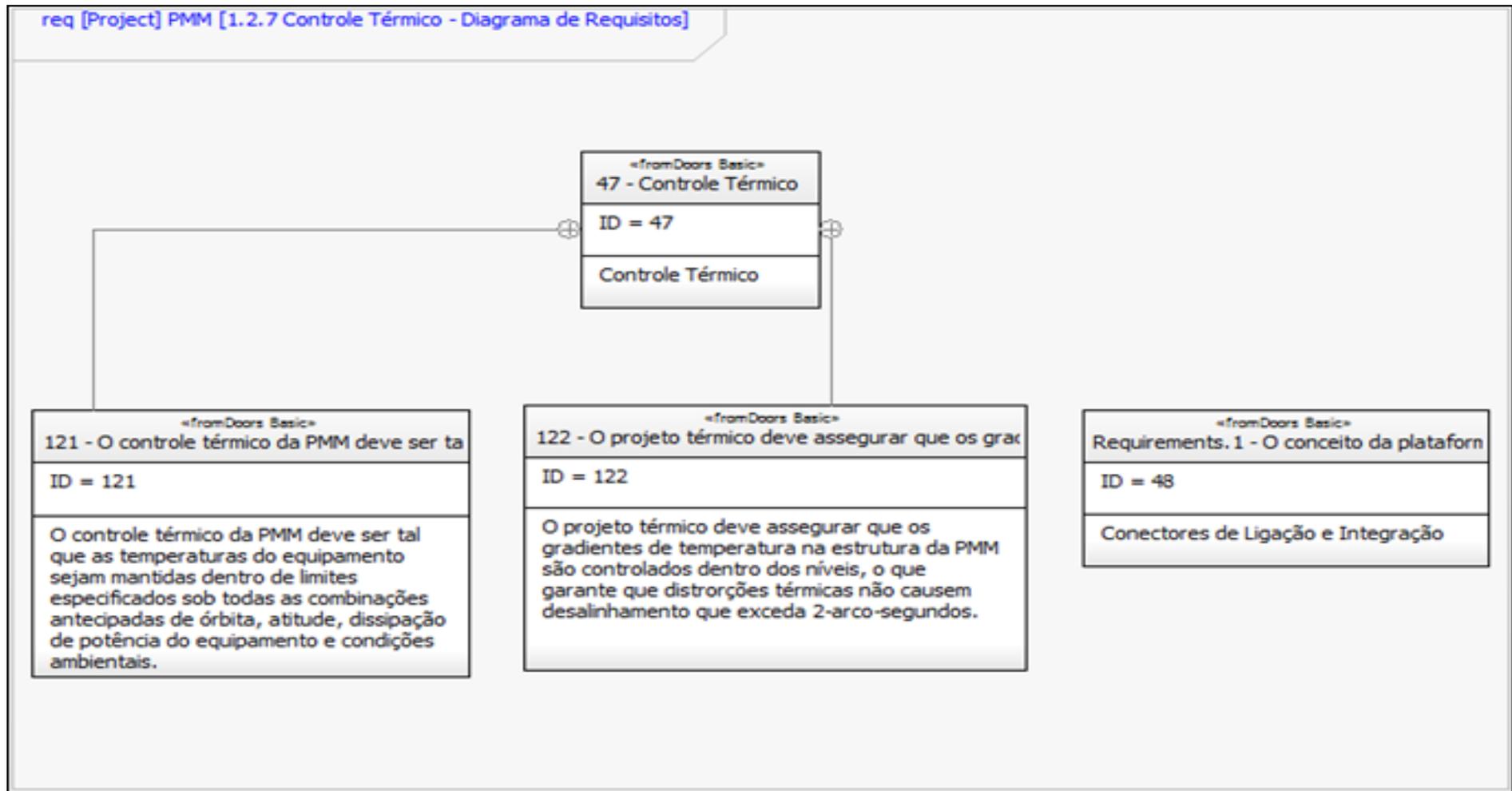


Figura B.14 - Decomposição dos Requisitos de Controle Térmico da PMM.

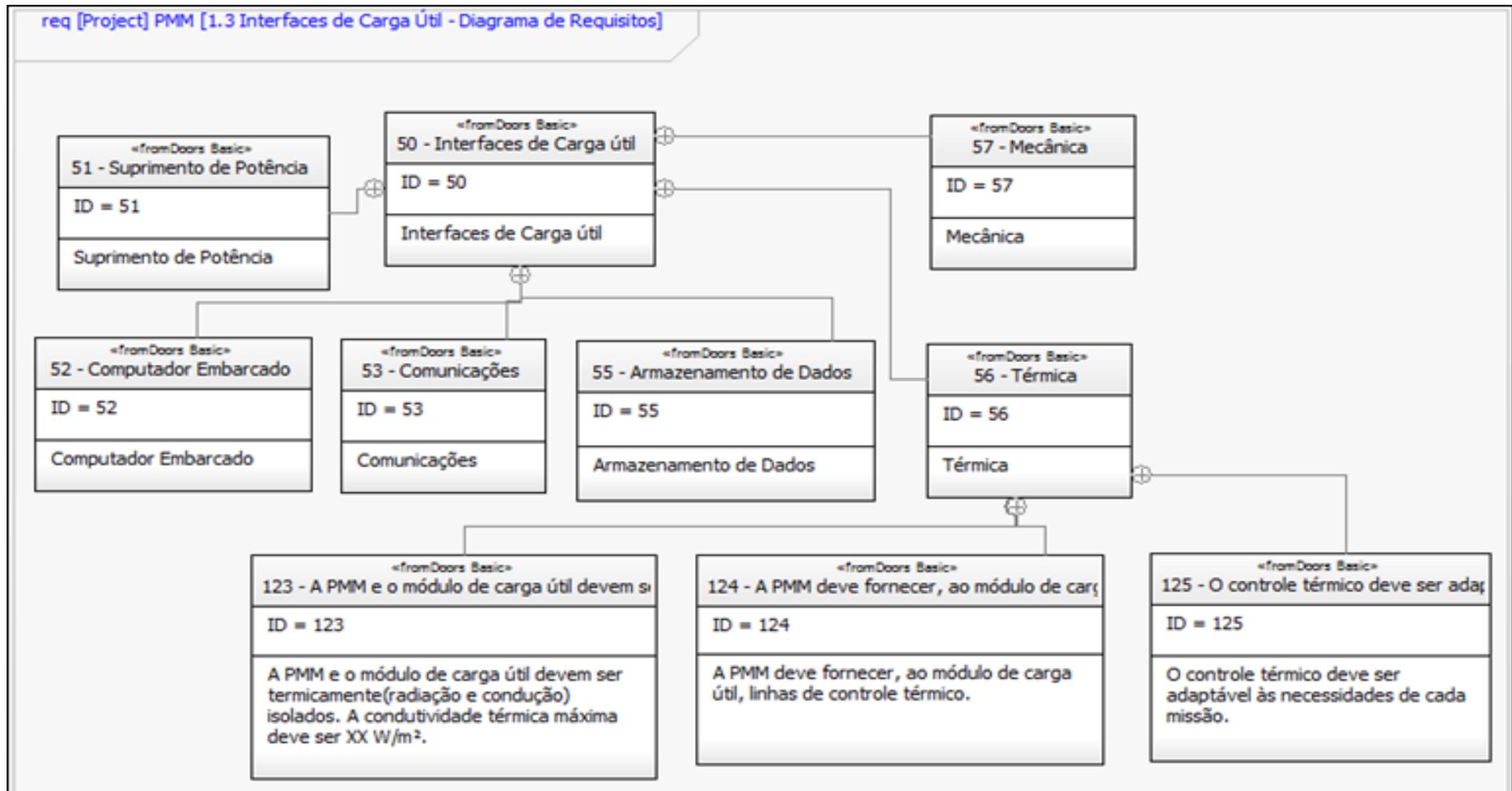


Figura B.15 - Decomposição dos Requisitos de Interface com a Carga Útil da PMM.

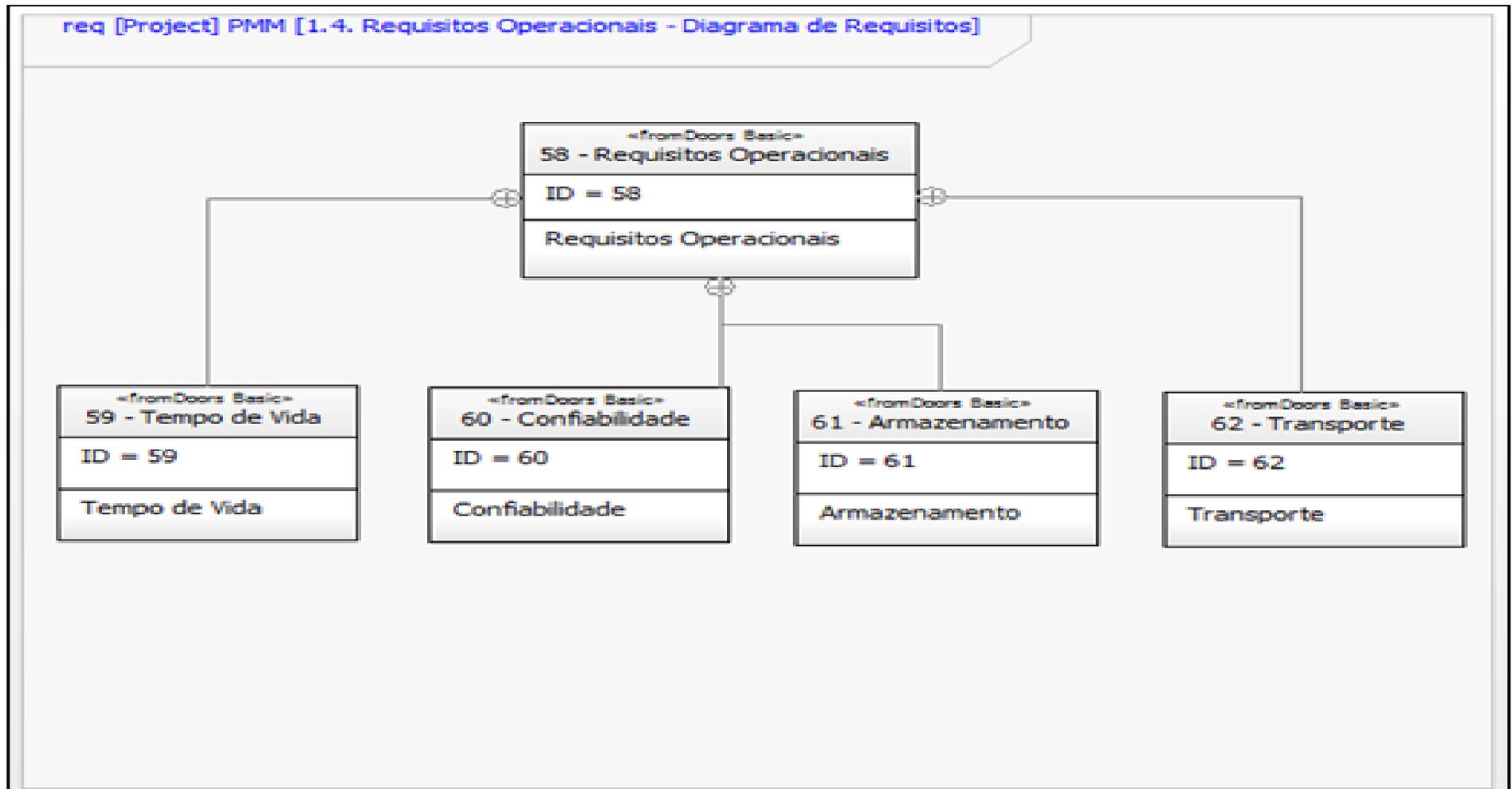


Figura B.16 - Decomposição dos Requisitos Operacionais da PMM.

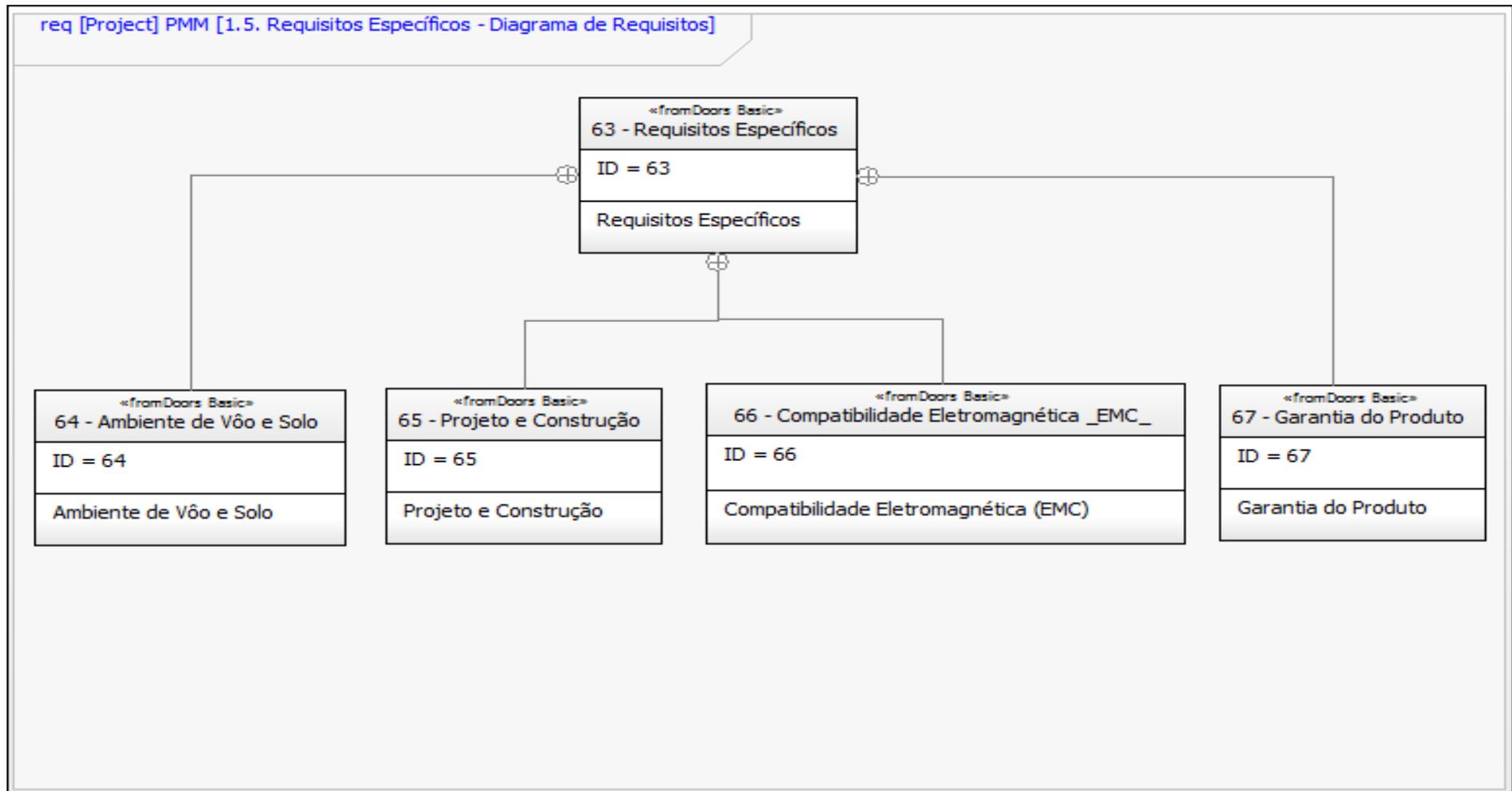


Figura B.17 - Decomposição dos Requisitos Específicos da PMM.

ANEXO A - COMPLEX SYSTEMS DEVELOPED WITH SYSTEM CONCURRENT ENGINEERING.

(CORSETTI et al., 2012)

Complex Systems Developed with System Concurrent Engineering

Andre Corsetti; Edson Alves Ribeiro; (Giuliani Paulineli Garbi)¹; Karina Zanta; Michele Medeiros. Geilson Loureiro².

Abstract. In fields such as space, aeronautics, mobility, telecommunications, banking, or defense, systems are becoming increasingly complex, consisting of a network of interrelated and interacting components, with significant heterogeneity and diverse lifecycles. The need for continuous changes in product, process and organization in order to keep manufacturing business competitiveness, make these segments seek to apply methods to innovate and develop their complex products and services with lower costs, improved productivity and quality, and in less lead-time. This paper presents the integrated and concurrent development of a complex product and their performing organizations with System Concurrent Engineering approach that performs stakeholder analysis, requirements analysis, functional analysis and implementation architecture analysis, simultaneously, for the product, its life cycle processes and their performing organization at all levels of the product hierarchy. In order to demonstrate the System Concurrent Engineering approach and the methodology, this paper shows a Satellite Container (Mechanical Ground Support Equipments – MGSE) as example of a complex systems development.

Keywords. Systems engineering, concurrent engineering, mechanical ground Support equipment, complex systems.

¹ Giuliani Paulineli Garbi (✉)
Engineering Department, College Anhanguera of São José, avenue Dr. João Batista de Souza Soares, 4121, CEP: 12236-660, Jardim Morumbi, São José dos Campos, São Paulo, Brazil
e-mail: giuliani.garbi@aesapar.com

² Geilson Loureiro (✉)
Integration and Testing Laboratory, Brazilian Institute of Space Research, avenue Astronautas, 1758, CEP: 12227-010, Jardim da Granja, São José dos Campos, São Paulo, Brazil
e-mail: geilson@lit.inpe.br

ANEXO B - EXPLORAÇÃO DO ESPAÇO DE PROJETO DE SATÉLITES PARA UM CENÁRIO REDUZIDO DE ANÁLISE.

(MEDEIROS et al., 2012)

Medeiros, M., et al.

Exploração do Espaço de Projeto de Satélites ...

EXPLORAÇÃO DO ESPAÇO DE PROJETO DE SATÉLITES PARA UM CENÁRIO REDUZIDO DE ANÁLISE

Michele Medeiros, Alex Barbosa Bastos, Carlos Alberto Monteiro Barbosa dos Santos, Marize Correa Simões, Walter Abrahão dos Santos

Instituto Nacional de Pesquisas Espaciais – Av. dos Astronautas, 1758 - CP515 - CEP 12227-010
São José dos Campos - SP, michele.medeiros@it.inpe.br, alexbbastos@gmail.com,
carlos.amsantos@gmail.com, rizesimoes@uol.com.br, walter.abrahaow@lac.inpe.br

Resumo: Este trabalho adiciona algumas áreas de conhecimento do PMBOK (*Project Management Body of Knowledge*), através de algoritmos derivados, para um estudo de caso de DSE (*Design Space Exploration*) focado em satélites. O DSE contribui particularmente na Engenharia de Sistemas durante a fase de projeto conceitual, quando se realizam a análise de diversos cenários e a geração de balanços para se verificar a viabilidade de arquiteturas candidatas para satélites. O estudo de caso aborda o DSE para uma determinada missão espacial onde o espaço de variação foi extremamente limitado pelo ênfase metodológica do trabalho: a Arquitetura A (estabilizada com 3-eixos) e a Arquitetura B (estabilizada por spin). A linguagem SysML (*Systems Modeling Language*) foi empregada para a modelagem da arquitetura juntamente com o aplicativo, SatBudgets, para automação de balanços.

Palavras-chave: Exploração do Espaço de Projetos-DSE, PMBOK, SysML, Engenharia de Sistemas Espaciais

Abstract: *This work adds some knowledge areas from PMBOK (Project Management Body of Knowledge) via a set of derived algorithms, to a DSE (Design Space Exploration) case study focused in satellites. DSE particularly contributes to Systems Engineering during the conceptual design phase where various scenario analysis are done while budgeting for viable candidate satellite architectures. The DSE case study undertaken for a certain space mission has an extremely reduced space variation since methodology is here valued: Architecture A (3-axis stabilized) and a Architecture B (spin-stabilized). SysML (Systems Modeling language) is used for architectural modeling jointly with application, named SatBudgets, for budgeting automation.*

Keywords: *Design Space Exploration, PMBOK, SysML, Space Systems Engineering*

1 Introdução

A Engenharia de Sistemas Espaciais é um processo iterativo e disciplinado que emprega práticas aceitas de engenharia para planejar, definir, controlar, projetar, analisar, integrar, verificar e então apoiar a produção, testes e operação de produtos de hardware, software, programas ou sistemas espaciais (LARSON, WERTZ, 1999). Esta atividade requer alta interação entre os engenheiros de sistemas, uma vez que suas tarefas são vinculadas entre si e um fluxo de trabalho em particular pode depender da saída de outros processos sobre responsabilidade de outros membros da equipe.

Devido à crescente demanda por sistemas espaciais, projetistas de satélites devem encontrar soluções de arquiteturas que satisfaçam as restrições naturais de tempo, qualidade, risco, custo tipicamente. Neste contexto, o projeto conceitual torna-se uma fase importante para o sucesso da missão (LARSON, WERTZ, 1999), pois é a fase inicial do processo de projeto de um produto e exige versatilidade na sua aplicação. Durante essa fase, é necessária a avaliação de diversos cenários que antevêm a arquitetura mais adequada ao sistema final. Isto requer flexibilidade na execução de processos intrínsecos e relevantes dessa fase inicial tais como análise de compromisso (*trade-offs*) e tomada de decisões. Essa fase do projeto, portanto, deve ser sistematizada com o uso do computador, para que seja possibilitada sua integração com as demais fases do projeto do produto, bem como a integração do processo global de projeto com as demais fases de produção de um produto. Com a modernização de sistemas e o consequente aumento de complexidade, faz-se necessário cada vez mais que projetos conceituais sejam trabalhados, a fim de especificar, analisar, projetar e verificar sistemas sem a necessidade de produção de artefatos.

O Guia de Conhecimento em Gerenciamento de Projetos (PMBOK) é um conjunto de normas reconhecido para a profissão de gerenciamento de projetos, ou seja, um padrão que descreve normas, métodos, processos e práticas estabelecidas (PMBOK, 2004). Devido à visão abrangente provida pelo PMBOK, os critérios de avaliação de arquiteturas neste contexto tendem a ser significativos e podem contribuir na fase de projeto

ANEXO C - LABE3S: A PROTOTYPE OF AN EXPERIMENTAL CONCURRENT DESIGN FACILITY FOR SPACE SYSTEMS ENGINEERING.

(MEDEIROS; SANTOS, 2012)

Medeiros, M., Santos, W.A.

A Proposal for a Concurrent Design...

LABE3S: A PROTOTYPE OF AN EXPERIMENTAL CONCURRENT DESIGN FACILITY FOR SPACE SYSTEMS ENGINEERING.

Michele Medeiros

INPE,
Avenida dos Astronautas, 1758, Jardim da Granja - CEP 12227-010 São José dos Campos/SP
55 (12)3208-6264,
michele.medeiros@lit.inpe.br

Walter Abrahão dos Santos

INPE,
Avenida dos Astronautas, 1758, Jardim da Granja - CEP 12227-010 São José dos Campos/SP
55 (12)3208-6000,
walter.abrahaolac@lac.inpe.br

Abstract: *This work proposes creating an computer infrastructure for supporting an experimental Concurrent Design Facility for Systems Engineering, here called LabE3S, gathering specific design tools, dedicated technologies, best practices and well-accepted standards. A first field trial has been setup to a restricted design space exploration (DSE) case study focused in satellites conceptual design where some knowledge areas from Project Management Body of Knowledge (PMBOK) have been added via derived algorithms. The case study uses two existing software tools, called SatBudgets and SpaceESB, for budgeting automation and service wrapping respectively. It is expected that a prototype will be tested at LSIS Lab at LIT/INPE and further extended to other INPE systems engineering areas. Some approaches on how to interoperate between different software tools are mentioned based on a service-oriented computing (SOC) paradigm. Finally, a round-trip model-driven engineering cycle is being pursued including satellite sub-system specific engineering tools.*

Keywords: *Space Systems Engineering, Concurrent Design Facility, SOA, Design Space Exploration..*

1 Introduction

Generally speaking, Systems Development is focused on successfully meeting stakeholder needs, and finding an optimal architectural solution for risk, time and cost reductions as any major product development. The Systems Engineering approach uses established techniques and tools that can help satisfy these constraints. This work proposes an architectural solution for a collaborative systems engineering environment integrating some automated development processes for improved product development and delivery. The architecture is intended to be used as an experimental concurrent design facility at LIT-INPE as a trial before being considered as well to other INPE systems engineering areas.

Similar initiatives have been taken by ESA for conceiving the first version of CDF as showed in [Bandedecchi et al. \[1999\]](#). The ESA CDF Approach was based on a process; a multidisciplinary team; an integrated design model; a facility and a software infrastructure. In this work the main focus is the software infrastructure, although the other elements are also considered due to their influence in the software infrastructure itself. It is important to mention that the Quality Assurance Processes and the Corporative Policy have to be respected as well.

A common database for storing the data extracted from different tools that are mapped using a Knowledge Base is proposed by [Moser and Biffi \[2012\]](#) and a similar approach is already used in this work, when the SatBudgets tools is used, this application extracts the data in a System Modeling language (SysML) model and uses as inputs in a rule based system that generates reports, but the data is not stored nowhere.

As a case study of the proposed architecture, is considered the satellites conceptual design using a DSE where some knowledge areas from PMBOK are applied via a set of derived algorithms. DSE particularly contributes to Systems Engineering during the conceptual design phase where various scenario analysis are done while budgeting for viable candidate satellite architectures. The DSE case study undertaken for a certain space mission has an extremely reduced space variation since methodology is here valued: Architecture A (3-axis stabilized) and a Architecture B (spin-stabilized). The SysML (Systems Modeling language) is used for architectural modeling jointly with an application, named SatBudgets, for budgeting automation.

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.