PEDRO SPADA GLOGOWSKY

**ENGENHARIA DE SISTEMAS BASEADA EM
MODELOS - MODELAGEM ORIENTADA A
OBJETOS DE SISTEMAS LOGÍSTICOS DE
ARMAZENAMENTO E RECUPERAÇÃO**

Dissertação apresentada à Escola Politécnica
da Universidade de São Paulo para obtenção
do Título de Mestre em Ciências.

São Paulo - SP
2018

PEDRO SPADA GLOGOWSKY

**ENGENHARIA DE SISTEMAS BASEADA EM
MODELOS - MODELAGEM ORIENTADA A
OBJETOS DE SISTEMAS LOGÍSTICOS DE
ARMAZENAMENTO E RECUPERAÇÃO**

Dissertação apresentada à Escola Politécnica
da Universidade de São Paulo para obtenção
do Título de Mestre em Ciências.

Área de Concentração:

3141 - Engenharia de Computação

Orientador:

Prof. Dr. José Sidnei Colombo Martini

São Paulo - SP
2018

Este exemplar foi revisado e alterado em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo - SP, 8 de março de 2018.

Assinatura do autor

Assinatura do orientador

FICHA CATALOGRÁFICA

Glogowsky, Pedro Spada

ENGENHARIA DE SISTEMAS BASEADA EM MODELOS - MODELAGEM ORIENTADA A OBJETOS DE SISTEMAS LOGÍSTICOS DE ARMAZENAMENTO E RECUPERAÇÃO/ P. Spada Glogowsky. – ed. rev. – São Paulo - SP, 2018.

159 p.

Dissertação (Mestrado) — Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Computação e Sistemas Digitais (PCS).

1. SysML. 2. Centros de Distribuição. 3. Engenharia de Sistemas Baseada em Modelos - MBSE. I. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Computação e Sistemas Digitais (PCS). II. t.

Dedico este trabalho à minha família.

AGRADECIMENTOS

Ao Prof. Dr. José Sidnei Colombo Martini, pela orientação em meus trabalhos de pesquisa e pelo tempo dispendido em aconselhamentos profissionais e pessoais.

Aos Profs. Drs. José Reinaldo Silva, Marcos Pereira Barretto, José Eduardo Deboni e Valter Castelhana de Oliveira, pelas inúmeras ajudas na área de engenharia de sistemas baseada e modelos, e em exemplos de utilização das linguagens OMG SysMLTM e UMLTM.

À minha família, Martin, Mônica e Luisa, pelo apoio constante, e pela compreensão nos momentos que trocamos nossa convivência pela minha dedicação a este trabalho.

A todos os profissionais das empresas visitadas, por terem me recebido tão cordalmente, e terem contribuído com conhecimentos de domínio sem os quais a escrita deste trabalho não teria sido possível.

À Escola Politécnica da Universidade de São Paulo e ao Programa de Pós-Graduação da Engenharia Elétrica, pela possibilidade de trabalhar em suas dependências, e pelas ferramentas e materiais fornecidos para a realização deste trabalho.

RESUMO

Este trabalho desenvolve um método para a comparação de soluções logísticas de armazenamento e recuperação, com aplicações em centros de distribuição, depósitos, armazéns e demais estruturas equivalentes. Tais soluções podem implicar desde o uso de paleteiras manuais e empilhadeiras contra-balanceadas, até em arranjos mais complexos, envolvendo trans-elevadores operando em corredores de prateleiras com vários metros de altura. A literatura existente para o design e a escolha de tais soluções ressalta o prevalecimento de métodos proprietários e *ad-hoc*, auxiliados por ferramentas de software demasiadamente genéricas. Assim, o método aqui proposto é elaborado seguindo os princípios da Engenharia de Sistemas Baseada em Modelos (MBSE), sendo expresso através da linguagem OMG SysMLTM, e montado com o auxílio de ferramenta de software *CASE* (*computer aided systems engineering*) disponível comercialmente. Utilizando-se das técnicas mencionadas, este trabalho demonstra o passo-a-passo da construção do método proposto, incluindo a formulação de um *template* de requisitos e de um modelo de referência, orientado a objetos, para sistemas logísticos de armazenamento e recuperação. Concluída a apresentação do método, o mesmo é aplicado em dois exemplos de estudos de viabilidade (*trade-studies*) que determinam soluções ótimas para um dado conjunto de requisitos de negócio. No primeiro exemplo tem-se como fator limitante o n° de endereços de armazenamento, e no segundo a área disponível para construção do armazém. O principal resultado obtido com esse trabalho é capacidade de simular, em um único ambiente, escolhas de soluções logísticas de armazenamento que consideram parâmetros do sistema como um todo, e não apenas de seus sub-sistemas isoladamente. Isto tornou possível mensurar como alterações nas especificações de um dado ponto de vista, como o estrutural, impactam na satisfação dos requisitos de outros pontos de vista, como o dinâmico ou financeiro. A MBSE, entretanto, ainda não pode ser considerada uma disciplina madura. As ferramentas de software que a ela dão suporte, bem como as listas de melhores práticas de suas aplicações estão em constante evolução e aprimoramento. Dessa forma, a aplicação dos princípios da MBSE no design e seleção de soluções logísticas de armazenamento, com adoção da orientação a objetos, pode ser tida como uma ideia inovadora.

Palavras-chave: SysML, Logística, Centros de Distribuição, Engenharia de Sistemas Baseada em Modelos, MBSE.

ABSTRACT

This work presents a method for warehouse storage solutions comparison. The existing research regarding the design and selection of such logistic solutions highlights the predominance of ad-hoc procedures, as well as the use of generic software tools. Therefore, the method herein presented shall be developed according to the model-based systems engineering (MBSE) principles, being describe through the system modeling language (SysML), and built inside a computer-aided system engineering (CASE) software tool, commercially available. The method's steps shall be thoroughly detailed, including the creation of a reference model for warehouse storage systems, and its further use in trade studies execution. Once the method is properly described, its validation is demonstrated through two case studies designed to compare storage solutions according to the number of pallet-positions offered, and its dimensional footprint. This work's main achievement is the possibility to simulate, in a single environment, warehouse storage solution's options that take into account parameters of the system as a whole, and not only its sub-systems separately. With that, it is possible to measure how changes in the specifications of a given view point, such as structural, impact the requirement's satisfaction of other view points, such as dynamic or financial. The MBSE, however, still can not be considered a mature discipline. The software tools that support it, as well as the lists of best practices of its applications are constantly evolving and improving. Thus, the application of MBSE's principles in the design, and comparison, of warehouse storage solutions, with the adoption of object orientation, can be considered an innovative idea.

Key-words: *SysML, Logistics systems, Warehousing, Model-based Systems Engineering, MBSE.*

LISTA DE ILUSTRAÇÕES

1	Sequência de operações em um sistema de armazenamento logístico .	21
2	Exemplo de sistema de armazenamento logístico 1 - AS/RS	22
3	Exemplo de sistema de armazenamento logístico 2 - empilhadeiras contra-balanceadas	22
4	Exemplo de sistema de armazenamento logístico 3 - empilhadeiras tri- laterais	23
5	Maneiras de se unitizar as mercadorias armazenadas	24
6	Tipos de soluções para sistemas de armazenamento logístico	25
7	Tipos de pesquisa científica	34
8	Contexto de um centro de distribuição	37
9	MBSE num contexto de design de centros de distribuição	38
10	Funções de um centro de distribuição	39
11	Decomposição lógica de um centro de distribuição em um diagrama de classes	40
12	Problemas de design de centros de distribuição, e número de ocorrên- cias de publicações	47
13	Arquitetura de referência para sistema de <i>picking</i>	49
14	Exemplos de organização de modelos SysML	56
15	Exemplo de definição de escopo	58
16	Exemplos de especificação caixa-preta	65

17	Exemplo de análise funcional	67
18	Exemplo de análise causal	69
19	Exemplo de análise causal, expressa em diagramas paramétricos . . .	70
20	Exemplo de decomposição estrutural de um sistema de interesse . . .	72
21	Exemplo de GDA representando categorias de propriedades de valor em SysML	75
22	Exemplo de modelo de referência (<i>MDref</i>) finalizado	76
23	Correspondência entre <i>template</i> e requisitos de projeto.	80
24	Extensões de modelos em SysML	84
25	Modelagem de requisitos utilizando blocos.	87
26	Exemplos de montagens de estudos de viabilidade	90
27	Equacionamento dos estudos de viabilidade	91
28	Instanciação e execução	95
29	Resultado do estudo de viabilidade	96
30	Organização do modelo de referência do sistema de armazenamento logístico.	99
31	Definição dos objetivos do sistema e do modelo de referência.	101
32	Escopo do sistema de armazenamento logístico.	102
33	<i>Template</i> TR de requisitos para o sistema de armazenamento logístico.	103
34	Requisitos não-funcionais para o sistema de armazenamento logístico, exibidos em formato tabular.	104
35	Requisitos funcionais para o sistema de armazenamento logístico, exi- bidos em diagrama.	105

36	Requisitos de interface para o sistema de armazenamento logístico, exibidos em diagrama.	105
37	Premissas e restrições para o sistema de armazenamento logístico, exibidos em formato tabular.	106
38	Relatórios com relações de derivação entre requisitos.	107
39	Especificação caixa-preta para o sistema de armazenamento logístico.	108
40	Relatório com relações de refinamento entre requisitos e especificação caixa-preta.	108
41	Análise funcional para a operação de armazenagem.	109
42	Análise funcional para a operação de retirada.	110
43	Análise causal para a MOE Área.	112
44	Análise causal para as MOEs de custos.	113
45	Listagem de blocos para composição estrutural.	113
46	O modelo de referência para sistemas de armazenamento logístico, <i>MDref</i>	115
47	Requisitos não funcionais para o estudo <i>TSpa</i>	117
48	Premissas para o estudo <i>TSpa</i>	118
49	Objetivo do estudo <i>TSpa</i>	118
50	Estudo <i>TSpa</i> inserido no diagrama de pacotes.	119
51	Mapeamento de requisitos do projeto <i>Ppa</i> (linhas) para o <i>template</i> TR (colunas)	119
52	Requisitos não-funcionais do <i>template</i> candidatos a parâmetros de entrada.	120

53	Requisitos apreendidos em bloco para o estudo <i>TSpa</i>	121
54	Os blocos constituintes do estudo de viabilidade <i>TSpa</i>	122
55	Equacionamento do estudo de viabilidade <i>TSpa</i>	124
56	Instanciação de <i>MDref</i> para a solução AS/RS	125
57	Instanciação do estudo de viabilidade <i>TSpa</i>	126
58	Organização das instâncias de <i>TSpa</i> na estrutura de pacotes.	127
59	Resolvedor paramétrico configurado para as instâncias do estudo <i>TSpa</i> , ainda sem valores.	128
60	Resolvedor paramétrico configurado para as instâncias do estudo <i>TSpa</i> , já com valores calculados.	129
61	Resolvedor paramétrico configurado para a execução em lote de múlti- plos cenários, com conexão a repositório de dados externo em planilhas.	130
62	Repositório de dados externo, em planilha, antes da execução dos ce- nários do estudo de viabilidade.	130
63	Resultados da execução dos cenários para a solução AS/RS.	131
64	Resultados da execução dos cenários para a solução com empilhadeiras contrabalanceadas.	132
65	Resultados da execução dos cenários para a solução com empilhadeiras tri-laterais para corredores estreitos.	133
66	Requisitos não-funcionais para o estudo <i>TSdf</i>	135
67	Premissas para o estudo <i>TSdf</i>	136
68	Objetivo do estudo <i>TSdf</i>	137

69	Mapeamento de requisitos do projeto <i>Pdf</i> (linhas) para o <i>template</i> TR (colunas). Três premissas não puderam ser mapeadas.	137
70	Diagrama paramétrico do bloco <i>RackStructure</i> para cálculo de tempos de ciclo.	139
71	Extensão de <i>MDref</i> através de relações de generalização.	139
72	Requisitos candidatos a parâmetros de entrada para o estudo <i>TSdf</i> . . .	140
73	Requisitos apreendidos em bloco para o estudo <i>TSdf</i>	140
74	Os blocos constituintes do estudo de viabilidade <i>TSdf</i>	141
75	Equacionamento de conversão de parâmetros do estudo de viabilidade <i>TSdf</i>	142
76	Instanciação do estudo de viabilidade <i>TSdf</i>	143
77	Resultados da execução dos cenários para a solução AS/RS.	144
78	Resultados da execução dos cenários para a solução com empilhadeiras contrabalanceadas.	145
79	Resultados da execução dos cenários para a solução com empilhadeiras tri-laterais para corredores estreitos.	146
80	Os quatro pilares da linguagem OMG SysML™	156
81	Fluxograma principal do método OOSEM	158

LISTA DE ABREVIATURAS E SIGLAS

CD	Centro de distribuição
AS/RS	Sistema Automatizado de Armazenamento e Recuperação
SysML	System Modeling Language
UML	Unified Modeling Language
SE	Engenharia de Sistemas
MBSE	Engenharia de Sistemas Baseada em Modelos
OO	Orientação a Objeto
OOSEM	Metodologia para Engenharia de Sistemas Orientada a Objeto
CASE	Engenharia de Sistemas Assistida por Computador
INCOSE	International Council of Systems Engineering
OMG	Object Management Group
MOE	Measure of Effectiveness

SUMÁRIO

1	Introdução	21
1.1	Motivação	21
1.1.1	Sistemas de armazenamento logístico	21
1.1.2	Engenharia de Sistemas - SE	27
1.1.3	Engenharia de Sistemas baseada em Modelos (MBSE)	28
1.2	Objetivo	30
1.3	Metodologia	32
1.4	Organização do trabalho	34
2	Revisão Bibliográfica	37
2.1	MBSE aplicada ao design de sistemas logísticos	37
2.1.1	McGinnis, Schmidt e Spee (2013) e McGinnis (2012)	37
2.1.2	Abdoli e Kara (2016)	39
2.1.3	Limère et al. (2010)	41
2.1.4	McGinnis (2014)	42
2.2	Métodos de escolha para soluções de armazenamento logístico	42
2.2.1	Baker e Canessa (2009)	43
2.2.2	James M., Meller e John A. (2010)	45
2.2.3	Gu, Goetschalckx e McGinnis (2007)	46

2.2.4	Ellinger (2012)	49
2.2.5	Goetschalckx, Mital e Huang (2014)	50
2.3	<i>Gap</i> de pesquisa	50
3	Desenvolvimento	53
3.1	Construção do modelo de referência	53
3.1.1	Atividades introdutórias	54
3.1.1.1	Organização do modelo	55
3.1.1.2	Objetivo do sistema	55
3.1.1.3	Objetivo do modelo	55
3.1.2	Elaboração do modelo	57
3.1.2.1	<i>Template</i> de requisitos e escopo do modelo	57
3.1.2.2	Requisitos não-funcionais, funcionais, de interface e premissas	60
3.1.2.3	Especificação caixa-preta	64
3.1.2.4	Análise funcional	67
3.1.2.5	Análise causal	68
3.1.2.6	Decomposição estrutural	71
3.1.2.7	Classificação de parâmetros e relações	72
3.1.2.8	Finalização do modelo de referência	76
3.2	Construção dos estudos de viabilidade	77
3.2.1	Atividades introdutórias	78

3.2.1.1	Elicitação de requisitos do cliente	78
3.2.1.2	A extensão do modelo de referência	81
3.2.2	Elaboração do estudo de viabilidade	85
3.2.2.1	Parâmetros de entrada e de saída	85
3.2.2.2	Modelagem dos requisitos	86
3.2.2.3	Diagrama de blocos	87
3.2.2.4	Equacionamento	90
3.2.2.5	Instanciação e execução	92
4	Resultados	97
4.1	Comparativo de soluções em função do número de posições de armazenamento	98
4.1.1	Construção do modelo de referência	98
4.1.2	Construção do primeiro estudo de viabilidade	116
4.2	Comparativo de soluções em função da área ocupada	134
4.2.1	Reutilização do modelo de referência	134
4.2.2	Construção do segundo estudo de viabilidade	134
5	Conclusões	147
	Referências	151
	Apêndice A - A Linguagem SysML	155
	Apêndice B - O método OOSEM	157

1 INTRODUÇÃO

Este trabalho tem como tema central o uso dos princípios da engenharia de sistemas baseada em modelos - MBSE - para a escolha de soluções de armazenamento e recuperação em sistemas logísticos, de modo a agregar valor em seus projetos.

1.1 Motivação

1.1.1 Sistemas de armazenamento logístico

Os sistemas logísticos de armazenamento e recuperação tem como função receber cargas unitizadas de produtos, armazená-las pelo tempo necessário, e passá-las a diante para os próximos elos da cadeia logística tão logo sejam requisitadas.

Figura 1: Sequência de operações em um sistema de armazenamento logístico (BARTHOLDI; HACKMAN, 2014).

Tais sistemas logísticos, também conhecidos coloquialmente como armazéns, estoques ou depósitos, são comuns nas cadeias logísticas e se fazem presentes sempre que há a necessidade de armazenagem de produtos em larga escala, tal como ocorre

em centros de distribuição, linhas de produção fabris, mercados atacadistas, centros de gestão de documentos, dentre outros (figuras 2, 3 e 4).

Figura 2: Exemplo de sistema de armazenamento logístico 1 - AS/RS (SOLUTIONS, 2016).

Figura 3: Exemplo de sistema de armazenamento logístico 2 - empilhadeiras contra-balanceadas (DACO, 2016).

As cargas unitizadas com as quais esses sistemas operam recebem denominações

Figura 4: Exemplo de sistema de armazenamento logístico 3 - empilhadeiras trilaterais (EQUIPAMENTOS, 2016).

de acordo com a forma, tamanho e capacidade que apresentam, sendo os pallets, as caixas e as bandejas as mais comuns (figura 5).

Os sistemas de armazenamento logístico, independentemente da finalidade a qual se prestem, têm seus custos de instalação e, principalmente, operação fortemente atrelados à área total que ocupam. Em contrapartida, o retorno financeiro que oferecem depende diretamente da quantidade de unidades que conseguem armazenar, e do quão rápido conseguem colocá-las e retirá-las do seu estoque.

Assim, é possível afirmar que um sistema de armazenamento logístico bem projetado deverá, essencialmente, minimizar tanto quanto possível a área ocupada, ao mesmo tempo em que maximiza o número de posições disponíveis e a rapidez com que as unidades podem ser lá postas e removidas. Um sistema de armazenamento logístico mal projetado poderá se transformar no gargalo de toda uma cadeia de fornecimento, o que se torna ainda mais crítico se considerarmos o aumento da exigência do nível de serviço pelo qual passaram, e ainda passam, os centros de distribuição a partir

Figura 5: Maneiras de se unitizar as mercadorias armazenadas (BARTHOLDI; HACKMAN, 2014).

do início dos anos 2000. Como resultado do aumento da participação do comércio eletrônico nas vendas de produtos (BARTHOLDI; HACKMAN, 2014), centros de distribuição que antes realizavam majoritariamente grandes envios, em baixa frequência, e com uma certa uniformidade e previsibilidade de itens para redes varejistas, passaram a ter que lidar com envios para consumidores finais, que têm por característica serem mais frequentes, compostos por uma ampla gama de produtos, e em quantidades mais fracionadas. Neste cenário, conseguir garantir uma capacidade adequada de giro de estoque ganha um peso ainda maior para os sistemas de armazenamento logístico.

Um sistema de armazenamento logístico pode se apresentar sob a forma de diversas soluções (figura 6), as quais variam nos tipos de cargas unitizadas com as quais operam, nos equipamentos empregados para movimenta-las e na maneira como são estocadas. Podem partir desde configurações simples, como movimentação com uso de empilhadeiras manuais e empilhamento diretamente sobre o piso, até configurações sofisticadas, como os sistemas automatizados de armazenamento e recuperação

(AS/RS)¹ os quais disponibilizam posições de armazenamento em prateleiras com dezenas de metros de altura, bem como operações de colocada e retirada de materiais totalmente automatizadas.

Figura 6: Tipos de soluções para sistemas de armazenamento logístico (ELBOWROOM, 2016).

Qual dessas soluções se mostrará a mais adequada para um sistema de armazenamento logístico em particular é algo que dependerá de como forem apresentados os requisitos de negócio por parte do cliente. O mesmo deverá apontar quais parâmetros devem ser levados em conta, e qual o peso que será dado a cada um deles. Centros de distribuição alimentícios frigorificados poderão priorizar alta capacidade de giro em detrimento do nº de posições, uma vez que os produtos não permanecerão no sistema de armazenamento logístico por muito tempo por conta de períodos de validade, e também do alto custo de mantê-los a baixas temperaturas. No outro extremo, centros de gestão de documentos certamente exigirão um grande nº de posições de armazenagem e uma não tão elevada capacidade de giro, dado que esses locais normalmente servem para manter arquivos-mortos de empresas, que não tem a necessidade de consultá-los com muita frequência. Entretanto, em ambos os exemplos citados, pode-se requerer que a área utilizada fique contida dentro de determinadas dimensões, pois, no primeiro caso, quanto maior for a ocupação maior será o custo com a refrigeração; e no segundo, o local de instalação normalmente se dará dentro de áreas urbanas, próximo a

¹*Automated Storage and Retrieval Systems*

empresas, onde o custo do metro quadrado tende a ser elevado.

Não é tarefa trivial listar e replicar as metodologias existentes para a seleção de soluções de sistemas de armazenamento logístico, pois muitas dessas boas práticas são conduzidas por empresas privadas, e o fazem por meio de ferramentas e métodos proprietários (MCGINNIS; SCHMIDT; SPEE, 2013). Trabalhos conduzidos por Baker e Canessa (2009) realizam uma consolidação das metodologias mais empregadas, tanto através de análises de publicações do setor, como através de entrevistas a especialistas. Suas conclusões apontam que simplesmente não há um protocolo padrão para o *design* de soluções de armazenamento logístico, e que tais empreitadas são essencialmente conduzidas através de processos *ad-hoc*, e suportadas por ferramentas de software genéricas, como planilhas, bancos de dados, programas de CAD e de simulação (MCGINNIS; SCHMIDT; SPEE, 2013).

Isto leva a diversos inconvenientes e transtornos na condução dos projetos de armazenamento logístico, pois planilhas, apesar de apresentarem facilidade de uso e de serem flexíveis, tendem a crescer em número e tamanho até o ponto de ficarem demasiadamente complexas, mesmo para as equipes diretamente envolvidas na sua administração. Já ferramentas como programas de CAD, de simulação e de gerenciamento de projetos, abordam, mesmo que de forma eficiente, apenas um ponto de vista do projeto, como o estrutural, dinâmico ou financeiro, pouco colaborando para assegurar que todas as partes do futuro sistema de armazenamento logístico se encaixem corretamente, e que o projeto funcione como um todo.

Tal situação não se verifica em outros tipos de sistemas de engenharia, tais como em projetos de aviões, satélites, telescópios ou mesmo veículos terrestres, onde o *design* é conduzido de forma a integrar as especificações de cada uma das disciplinas presentes, e, ao mesmo tempo, tendo os seus *workflows* com alto grau de formalização, e com o devido suporte de ferramentas de software específicas para tal (MCGINNIS; SCHMIDT; SPEE, 2013). Um sistema de armazenamento logístico compartilha mui-

tas das características com os sistemas mencionados, no sentido de que há diversas disciplinas envolvidas no seu design, inclusive de fora da engenharia, e que pode haver também diversos tomadores de decisão representando organizações distintas, cujas decisões de design interagem de forma complexa e, por vezes, pouco óbvia. A disciplina de Engenharia de Sistemas, a qual emergiu na segunda metade do século 20, tem como intuito prover padrões e boas práticas para dar suporte à essas decisões de design de sistemas.

1.1.2 Engenharia de Sistemas - SE

De acordo com o Conselho Internacional de Engenharia de Sistemas (INCOSE)², Engenharia de Sistemas (SE) é uma abordagem interdisciplinar que tem como intuito fornecer meios para se construir sistemas de forma bem-sucedida. A SE tem como foco definir as necessidades do cliente e as funcionalidades requeridas do sistema bem no início do ciclo de desenvolvimento do mesmo, documentando requisitos para então proceder para as etapas de design e validação, mas sempre tendo em perspectiva o problema como um todo (BKCASE Editorial Board, 2014).

A SE implica na adoção simultânea de processos técnicos e gerenciais, para que se chegue a uma solução balanceada para o sistema a ser construído, mitigando os riscos que possam impactar no sucesso do projeto do mesmo. Os processos gerenciais da SE são aplicados para garantir que os objetivos de custo, tempo e performance sejam atingidos. Tais processos tipicamente incluem controle da linha de base do projeto, monitoramento da performance técnica e gerenciamento de riscos. Já os processos técnicos da SE são aplicados para se efetivamente especificar, desenhar, construir e verificar o sistema (BKCASE Editorial Board, 2014).

A motivação por trás da adoção da SE se baseia no fato de que não importa se o sistema em questão é um avião militar, um novo veículo híbrido, um *smartphone* ou

²*International Council on Systems Engineering*

mesmo um sistema logístico de armazenamento; espera-se que os sistemas modernos, projetados e construídos principalmente a partir do início dos anos 2000, performem de maneiras inimagináveis a uma ou duas gerações atrás. As pressões competitivas demandam que os sistemas, quaisquer que sejam, façam uso dos avanços tecnológicos para entregar contínuos aumentos de performance, redução de custos e períodos de desenvolvimento cada vez mais curtos. Essas mesmas pressões direcionam os requisitos de negócio para um aumento de funcionalidades, interoperabilidade, confiabilidade e redução de tamanho (FRIEDENTHAL; MOORE; STEINER, 2012).

As práticas de desenvolvimento de sistemas devem dar suporte a esses aumentos de demanda. A SE é uma abordagem que tem sido dominante nas indústrias aeroespacial e de defesa como forma de fornecer soluções sistêmicas a problemas críticos e desafiadores do ponto de vista tecnológico. Tais soluções, com bastante frequência envolvem disciplinas relacionadas a hardware, software, dados, pessoas e infraestrutura. As práticas da SE tem evoluído continuamente para conseguir tratar a crescente complexidade dos sistemas mais atuais, os quais não se limitam mais apenas a aeroespaciais e de defesa. Como resultado, a abordagem da SE tem ganhado uma ampla aceitação em outros ramos da indústria, como a automotiva, de telecomunicações, de equipamentos médicos e logística, para citar alguns (FRIEDENTHAL; MOORE; STEINER, 2012).

1.1.3 Engenharia de Sistemas baseada em Modelos (MBSE)

Com frequência, modelos são utilizados para se projetar e analisar sistemas complexos e interdisciplinares. Isto permite capturar características de todos os elementos do sistema, interrelacioná-las, e descrever o sistema em qualquer nível de abstração. Os modelos podem ser analisados e refinados, e ser utilizados como fontes de descrições detalhadas para, eventualmente, permitir a construção do sistema (PEARCE; FRIEDENTHAL, 2013).

Rápidos avanços em termos de hardware e software, a partir da década de 2000, têm permitido a evolução da SE em direção a Engenharia de Sistemas Baseada em Modelos (MBSE), na qual os modelos, com o apoio de ferramentas de software *CASE*, substituem os tradicionais documentos como produtos principais dos processos de engenharia. Uma das promessas da MBSE é que a vasta coleção de modelos que representa um sistema pode ser concebida de forma mais consistente do que uma coleção de documentos, sendo eventualmente até interoperável e reduzindo significativamente os riscos e o tempo de desenvolvimento do referido sistema (MCGINNIS; SCHMIDT; SPEE, 2013).

Tradicionalmente, grandes projetos de engenharia empregam a abordagem da SE baseada em documentos, a fim de executar os processos da mesma conforme descrito na seção 1.1.2. Essa abordagem é caracterizada pela geração de especificações textuais e documentos de design, tanto em formato de papel quanto de arquivos de computador, que são trocados entre usuários, clientes, desenvolvedores e responsáveis por testes. Os requisitos do sistema a ser desenvolvido, bem como informações sobre seu design, são expressos nesses documentos e desenhos. A ênfase da SE é colocada sobre o controle da documentação, e em garantir que os documentos e desenhos são válidos, completos e consistentes, e que o sistema sendo desenvolvido está em concordância com os mesmos (FRIEDENTHAL; MOORE; STEINER, 2012).

A abordagem baseada em documentos pode ser rigorosa, conhecida e tradicional, mas tem algumas limitações fundamentais, conforme já mencionado nos métodos existentes para escolha de soluções de sistema de armazenamento logístico.

A completude, consistência e relações entre requisitos, design, análises de engenharia e dados sobre testes de um sistema são difíceis de avaliar, uma vez que tais informações se encontram espalhadas por vários documentos, sejam eles físicos e/ou em formato digital. Isto torna difícil a compreensão de um determinado aspecto do sistema, bem como a execução de rastreabilidades entre seus componentes e avaliações

sobre o impacto que mudanças no seu design poderiam ocasionar. O que, por sua vez, leva a uma sincronização falha entre os requisitos e o design no nível de sistema, e o design de baixo nível de componentes de software e hardware. Adicionalmente, também acaba se tornando complicado a reutilização de requisitos e de soluções de design, para o caso de sistemas em constante evolução ou para suas arquiteturas variantes. Essas limitações podem resultar em ineficiências e em problemas sérios de qualidade, os quais normalmente vem a se fazer notar durante as fases de integração e de testes, ou então, ainda pior, quando o sistema já foi entregue ao cliente (FRIEDENTHAL; MOORE; STEINER, 2012).

A ideia por trás da abordagem via modelos é que todos os dados e decisões relevantes sobre o design de um sistema, como, por exemplo, de sistemas de armazenamento logístico, sejam capturadas em um modelo central, compreensível a todos os tomadores de decisão e, principalmente, que seja único. As decisões tomadas pelos líderes de cada equipe deverão ser obrigatoriamente consistentes com esse modelo central, e deverão ter como base dados provenientes deste mesmo modelo. Dessa forma, documentos e planilhas podem se prestar a fins mais adequados, como *output* de dados e criação de relatórios gráficos.

1.2 Objetivo

Com base nas motivações expostas na seção anterior, o objetivo deste trabalho é a construção de um método automatizado, fundamentado na MBSE e expresso em linguagem SysML, para a comparação de soluções logísticas de armazenamento. Tal método será designado pela sigla *MTcsl*.

O método *MTcsl* proposto consistirá na aplicação da seguinte sequência de etapas:

1. Elaboração de um modelo de referência paramétrico, denominado pela sigla *MDref*, para sistemas logísticos de armazenamento.

2. Utilização de *MDref* na montagem de estudos de viabilidade *TSx*³. Dado um projeto de armazenamento logístico *Px* qualquer, o mesmo implicará numa lista de requisitos *LRx* a ser elicitada do cliente. A lista *LRx* deverá ser usada para a formatação de um estudo de viabilidade *TSx*, que fará uso das equações e parâmetros de *MDref* para produzir o comparativo entre as possíveis soluções de armazenamento.

Tais etapas, a serem pormenorizadas no capítulo 3 - Desenvolvimento, conduzirão o engenheiro de sistemas a partir do ponto em que os requisitos são elicitados do cliente, até o momento em que uma solução logística de armazenamento (figura 6, seção 1.1), que atenda àqueles requisitos, possa ser indicada.

A avaliação de *MTcsl* para escolha de soluções logísticas de armazenamento consistirá na sua comparação com a abordagem tradicional, não-sistêmica e baseada em documentos. A fim de se mostrar como uma solução de fato mais eficiente ao problema apresentado neste trabalho, é desejável que o novo método atenda às seguintes condições:

- C_1 Unicidade. Deverá existir um único modelo para concentrar as informações das soluções sendo avaliadas;
- C_2 Informações gráficas. O entendimento do modelo *MDref*, bem como dos estudos *TSx* sendo conduzidos, deverá poder ser feito primordialmente de forma visual, e não textual;
- C_3 Múltiplas instâncias. Deverá haver uma separação clara entre o modelo de referência abstrato *MDref*, e suas diversas instâncias representando as soluções sendo avaliadas pelos estudos *TSx*;

³*TS* - sigla para termo em língua inglesa *trade-study*.

- C_4 Equacionamento discriminado. Os cálculos não poderão ficar ocultos e distribuídos entre diversos documentos e planilhas, mas sim visivelmente organizados dentro de *MDref* e dos diversos *TSx*.
- C_5 Execução de cálculos. Deverá ser possível calcular soluções com o método *MTcsl* sem a necessidade de se criar rotinas em linguagens de programação para tal.
- C_6 Extensões. Deverá ser possível realizar a extensão do modelo de referencia *MDref*, isto é, inserir, modificar ou remover componentes e cálculos do mesmo, sem que se tenha de refazer-lo completamente. Assim, o método *MTcsl* proposto poderá ser aplicado com pouco retrabalho em novos cenários, cujas listas *LRx* incluam novos requisitos não previstos originalmente.
- C_7 Rastreabilidade. A aplicação do método *MTcsl* deverá permitir diversas operações automatizadas de rastreabilidade, tais como rastreabilidade de requisitos, diagramas de causa-e-efeito e análises de impacto.

1.3 Metodologia

Esta seção descreve a metodologia de pesquisa científica adotada para a realização deste trabalho. Aqui, a pesquisa efetuada será classificada de acordo com sua natureza, objetivos e procedimentos técnicos.

Primeiramente, quanto a sua natureza, este trabalho trilhou o caminho da pesquisa científica dita *aplicada*, em oposição à *teórica*. A pesquisa aplicada tem como finalidade gerar conhecimentos para aplicações práticas, dirigidos à solução de problemas específicos (PROVDANOV; FREITAS, 2013). Esta descrição recai na situação exposta na seção 1.2 - Objetivos deste trabalho, onde é explicitado que a sua contribuição será um método automatizado, apoiado na MBSE, que visa resolver o problema específico

de escolha de soluções logísticas de armazenamento, porém de forma mais eficiente do que os métodos atuais (ver capítulo 2).

Em seguida, e do ponto de vista de seus objetivos, a pesquisa aqui realizada pôde ser classificada como do tipo *exploratória*, pois sendo a utilização do paradigma da MBSE para a resolução de problemas logísticos considerada uma abordagem ainda inovadora (MCGINNIS; SCHMIDT; SPEE, 2013), pesquisas na área ainda se encontram na fase preliminar. Assim, a pesquisa exploratória aqui adotada teve como finalidade proporcionar mais informações sobre o assunto investigado, possibilitando sua definição e seu delineamento de uma forma mais efetiva. A pesquisa exploratória possui planejamento flexível, o que permite o estudo do tema sob diversos ângulos e aspectos (PROVDANOV; FREITAS, 2013). A mesma, neste trabalho, envolveu:

- Levantamento bibliográfico específico para sistemas de armazenamento logístico;
- Entrevistas com especialistas de domínio, detentores de experiências práticas em projetos de sistemas de armazenamento logístico;
- Estudo de projetos bem-sucedidos de sistemas de armazenamento logístico.

Por fim, os procedimentos técnicos adotados para esse trabalho de pesquisa, em decorrência de sua natureza exploratória, foram combinações de *pesquisas bibliográficas* e de *estudos de caso* (figura 7).

Figura 7: Tipos de pesquisa científica (PROVDANOV; FREITAS, 2013)

1.4 Organização do trabalho

O restante deste trabalho é organizado conforme descrito nos parágrafos a seguir.

No capítulo 2 são apresentados, e resumidos brevemente, todas as publicações que foram mais relevantes para a elaboração deste trabalho. Além da descrição sucinta de cada uma, ao final do capítulo é apresentado também o *gap* de pesquisa, isto é, os pontos específicos que as publicações citadas deixaram de explorar, e que serão cobertos por este trabalho.

Em seguida, no capítulo 3, é realizado o desenvolvimento do trabalho, onde são pormenorizadas todas as etapas de modelagem necessárias para se chegar ao modelo de referência proposto, *MDref*, bem como aos estudos de viabilidade.

Na sequência, no capítulo 4, os resultados da aplicação do método *MTcsi* são analisados através da utilização de *MDref*, previamente elaborado no capítulo 3, em dois estudos de viabilidades distintos, sendo um para análise de número de posições de armazenamento, e o outro para área ocupada.

Por fim, o capítulo 5 conclui o trabalho, fazendo um resumo das contribuições do mesmo. São mencionados os pontos que ainda podem ser aprimorados no processo de escolha de soluções de armazenamento logístico usando as técnicas da MBSE, bem como os direcionamentos para trabalhos futuros.

Nos apêndices A, B e C tem-se descrições resumidas da linguagem OMG SysML™, do método de modelagem OOSEM e da ferramenta *case* MagicDraw™, respectivamente, a fim de contextualizar o leitor com o ferramental utilizado para o desenvolvimento deste trabalho.

2 REVISÃO BIBLIOGRÁFICA

Neste capítulo serão apresentados os artigos que mais contribuíram para a formulação do método de seleção de sistemas de armazenamento logístico, utilizando os princípios da MBSE.

2.1 MBSE aplicada ao design de sistemas logísticos

Nesta seção são discutidos os artigos que, até o momento da escrita deste trabalho, abordaram os problemas relacionados ao design de sistemas logísticos, nos quais estão inclusos sistemas de armazenamento, através das práticas da MBSE e do uso de linguagens gráficas de modelagem, como a UML e a SysML.

2.1.1 McGinnis, Schmidt e Spee (2013) e McGinnis (2012)

Figura 8: Contexto de um centro de distribuição (MCGINNIS; SCHMIDT; SPEE, 2013).

McGinnis, Schmidt e Spee (2013) e McGinnis (2012) abordam centros de distribuição (CDs), especificamente. Em seus trabalhos, realizam descrições sucintas que ambientam o leitor a respeito do que são, e para que servem, os centros de distribuição.

Mencionam, com base em suas revisões bibliográficas, em particular Baker e Canessa (2009), a ausência de métodos padronizados para se resolver o problema de design de CDs.

Figura 9: MBSE num contexto de design de centros de distribuição (MCGINNIS; SCHMIDT; SPEE, 2013).

Ambos os artigos avaliam os CDs como sendo sistemas interdisciplinares de engenharia e que, portanto, recaem dentro do escopo da engenharia de sistemas baseada em modelos. Apontam, inclusive, as vantagens de se utilizar tal abordagem. Dão sequência, em seus desenvolvimentos, na elaboração de modelos de referência orientados a objeto para CDs, utilizando a OMG SysML™ como linguagem gráfica de modelagem. Seguem, entretanto, metodologias distintas para a elaboração dos referidos modelos, sendo que McGinnis (2012) se apoia numa abordagem axiomática para tal, definindo primeiramente os princípios que norteiam as operações de CDs, para em seguida formalizá-los através da modelagem. Já McGinnis, Schmidt e Spee (2013) utiliza partes do método OOSEM (anexo B) para a elaboração do modelo de referên-

cia, percorrendo etapas como, por exemplo, a definição do contexto do sistema, e a listagem de requisitos funcionais.

Por fim, concluem suas contribuições apontando caminhos futuros que a utilização dos princípios da MBSE para o design de centros de distribuição poderia seguir. Ressaltam a utilização dos modelos de referência para a avaliação de alternativas de soluções através de *trade-studies* McGinnis (2012), e sua aplicação em demais sistemas que possuam pontos de vista logístico relevantes, como fábricas e hospitais McGinnis, Schmidt e Spee (2013).

Figura 10: Funções de um centro de distribuição (MCGINNIS; SCHMIDT; SPEE, 2013).

2.1.2 Abdoli e Kara (2016)

Abdoli e Kara (2016) também restringe a aplicação da MBSE aos centros de distribuição, e tem como objetivo em seu trabalho investigar como o design conceitual dos mesmos pode ser melhorado através da utilização das técnicas da MBSE, seguindo o paradigma da orientação a objeto.

Inicia seu artigo fornecendo uma breve descrição dos CDs e como a literatura existente a respeito de seu design é essencialmente *ad-hoc*. Assim como os autores

dos trabalhos da seção anterior, Abdoli e Kara (2016) também classifica os CDs como sendo sistemas de sistemas (SoS), e enquadra-os dentro do escopo da MBSE, cujas etapas também descreve sucintamente.

Menciona o trabalho de McGinnis, Schmidt e Spee (2013) e busca complementá-lo, propondo uma metodologia orientada a objeto para o design conceitual de CDs. Entretanto, opta pela utilização da linguagem OMG UMLTM para a descrição dos modelos, não formalizando a rastreabilidade de requisitos, tão pouco a relação entre os atributos dos objetos em diagramas paramétricos.

O resultado da aplicação da metodologia proposta no artigo é expresso num diagrama de classes da OMG UMLTM, onde o CD é decomposto em suas estruturas e processos principais (figura 11).

Figura 11: Decomposição lógica de um centro de distribuição em um diagrama de classes (ABDOLI; KARA, 2016).

Em sua conclusão, sinaliza que os resultados obtidos com a modelagem conceitual do CD poderá servir de base, em trabalhos futuros, para a geração de arquiteturas alternativas dos mesmos.

2.1.3 Limère et al. (2010)

Limère et al. (2010) buscam a elaboração de um modelo de custo de logística fabril, sendo o mesmo expresso em linguagem OMG SysMLTM. O objetivo de seu trabalho é minimizar os custos envolvidos na movimentação de partes e peças para o suprimento de linhas de produção.

Iniciam descrevendo resumidamente a linguagem OMG SysMLTM, mencionando o fato de que ela até então vinha sendo utilizada para a elaboração de modelos de sistemas físicos, e não abstratos como no caso do modelo de custos apresentado no trabalho. Abordam também a ferramenta CASE MagicDrawTM e o seu plug-in ParaMagicTM, utilizados para a construção e execução do modelo de custo, respectivamente.

Seguem descrevendo o escopo do problema, no qual explicam os modos como peças são transportadas dos estoques até as linhas de produção: em lote (*bulk*), onde as partes chegam a linha sem estarem destinadas a nenhuma montagem; ou em kit, onde as partes chegam em coleções já reservadas a uma montagem específica. Limère et al. (2010) parametrizam o problema e descrevem matematicamente uma função-objetivo de custo.

Por fim, transferem a função-objetivo para dentro do modelo em SysML, através da montagem de Diagramas de Blocos e Diagramas Paramétricos. Criam instâncias dos blocos estruturais e executam as equações dos blocos de restrição, conseguindo obter resultados de simulações de custo diretamente a partir do modelo gráfico.

Concluem mencionando que uma vantagem, dentre várias, de se ter o modelo de custo feito em SysML é que o mesmo não somente apresenta graficamente a estrutura dos custos logísticos da fábrica, o que poderia ser feito em ferramentas genéricas como MS PowerPointTM, mas também permite a execução de análises com valores reais. Isto elimina a necessidade de esforços duplicados para se construir primeiro um modelo estrutural, e depois outro equivalente para análises computacionais.

2.1.4 McGinnis (2014)

McGinnis (2014) segue, em seu trabalho, trajetória semelhante a McGinnis, Schmidt e Spee (2013) e McGinnis (2012), no que se refere ter como foco centros de distribuição (CDs), e buscar a concepção de uma arquitetura de referência, expressa em OMG SysML™, para os mesmos.

Entretanto, o ponto em que os últimos diferem do primeiro está no fato de que este desenvolve um modelo de referência que permite a adoção de técnicas automatizadas, e reutilizáveis, de análise e simulação, a serem implementadas durante o processo de design, tornando-o interativo.

McGinnis (2014) cria uma arquitetura de referência para um centro de distribuição baseada em Redes de Petri, e a expressa em SysML. A partir dela, torna-se possível a realização transformações do modelo de referência genérico para modelos simuláveis em linguagens especializadas (M2M - *Model to Model Transformation*), a fim de responder a questões específicas que possam surgir durante o processo de design.

Dessa forma, ele conclui ressaltando que esta é uma contribuição que visa eliminar o *gap* existente entre as metodologias de design e de análise de sistemas logísticos de movimentação de materiais.

2.2 Métodos de escolha para soluções de armazenamento logístico

Nesta seção são discutidos os artigos que, até o momento da escrita deste trabalho, abordaram as metodologias existentes para a seleção de sistemas de armazenamento logístico.

2.2.1 Baker e Canessa (2009)

Baker e Canessa (2009) realizam um trabalho de síntese bastante completo a respeito de métodos e ferramentas disponíveis para o design de centros de distribuição (CDs).

Iniciam seu artigo fazendo um resumo da importância dos CDs nas cadeias logísticas, e os motivos pelos quais são necessários. Ressaltam pontos como sua participação no total nos custos com operações logísticas, que, segundo fontes pesquisadas, gira em torno de 22%, bem como o crescente investimento em automação, acentuado a partir da primeira década dos anos 2000.

Após menção resumida dos diversos pontos que tornam CDs peças essenciais nas cadeias logísticas, apontam que tais centros deveriam ser desenhados tendo em vista a otimização de custos, e que é justamente durante a fase de design que serão definidos o impacto dos mesmos.

Baker e Canessa (2009) afirmam que pouco foi publicado visando o design geral dos CDs, mas, em contrapartida, há muito material disponível abordando aspectos específicos dos mesmos, como layout, políticas de picking, escolha de equipamentos e tempos de deslocamento de pessoas e máquinas. E que na ausência de metodologias bem definidas e universalmente aceitas, os designers de tais centros recorrem a métodos próprios, *ad-hoc*. Termos como "intuição", "experiência" e "na prática" foram registrados com frequência na elaboração de seu trabalho.

Baker e Canessa (2009) conduziram sua pesquisa sobre métodos e ferramentas de design de CDs através de consultas a bibliotecas e a bases de dados eletrônicas, científicas e comerciais, incluindo ScienceDirect, EBSCO Business, Emerald Group Publishing e ProQuest. Tais bases foram pesquisadas através do uso de palavras-chaves como "centro de distribuição", "instalações", "movimentação de materiais", "planta", "armazenagem", em combinação com termos tais como "design", "layout" e "opera-

ções". Os resultados foram filtrados pela leitura de títulos e *abstracts* e, na sequência, houve a extensão da pesquisa através da seleção dos livros e artigos citados nos resultados encontrado. Por fim, houve a classificação dos materiais em dois grupos: aqueles que esboçavam ou sugeriam passos a serem seguidos no design de CDS, e aqueles que buscavam analisar ferramentas e técnicas.

A fim de validar o material de pesquisa coletado, Baker e Canessa (2009) entraram em contato com um total de 12 empresas atuantes no ramo de design de sistemas logísticos. A elas realizaram busca semelhante, indagando sobre sequência de passos e ferramentas mais utilizadas nas etapas de design.

Tanto as sequências de passos e as ferramentas utilizadas foram apresentadas no artigo em formato tabular, facilitando a análise dos resultados. E o que se pode concluir, segundo Baker e Canessa (2009), é que as principais ferramentas são um tanto genéricas, como programas de CAD, bancos de dados e, principalmente, planilhas. E os métodos, quando agrupados e estudados, podem ser resumidos apenas em passos genéricos:

- Definir requisitos de negócio;
- Refinar os requisitos de negócio para obter os requisitos do Centro de Distribuição;
- Especificar as UoHs (*pallets*, caixas) e definir as direções de fluxo de material dentro do CD;
- Selecionar, dimensionar e configurar as tecnologias que darão suporte ao fluxo de material;
- Determinar o arranjo físico dos componentes do CD;
- Avaliação detalhada de performance;

- Repetir os passos acima, em um processo iterativo.

Concluem dizendo que há um excesso de confiança depositado na experiência individual de designers de CDs, tanto na escolha de ferramentas quanto na avaliação de alternativas de solução. Uma metodologia compreensível e universal para o design de CDs parece, portanto, uma meta ainda muito longe de ser atingida.

Aparentemente, não há nenhuma solução otimizada, estilo "caixa-preta", para todo o processo de design, onde os dados levantados a partir dos requisitos pudessem ser inseridos numa ferramenta especializada, e um design ótimo fosse produzido.

2.2.2 James M., Meller e John A. (2010)

James M., Meller e John A. (2010) propõem uma abordagem empírica para o design de centros de distribuição (CDs), baseando-se nas práticas e métodos existentes na indústria, mas buscando formalizá-los para que possam ser ensinados e replicados na prática.

Iniciam seu trabalho de forma semelhante a Baker e Canessa (2009), citando-o inclusive, e apontando a ausência de métodos existentes para abordar o design de centros de distribuição como um todo, e todas as desvantagens inerentes ao fato. Entretanto, buscam avançar no trabalho iniciado por Baker e Canessa (2009) propondo uma metodologia para suprir o *gap* apontado.

A abordagem empírica de design apresentada por eles é baseada em duas metodologias complementares. A primeira delas se define como "listas de parâmetros a se considerar", e a segunda como "matrizes-solução". Listas de parâmetros seriam tudo aquilo que o designer deveria levar em consideração ao projetar um centro de distribuição. Naturalmente, e assim apontado por James M., Meller e John A. (2010), tal lista seria gigantesca se fosse única. Na prática, a mesma é determinada pelo segmento da indústria que o centro de distribuição irá operar, sendo equivalente a um template de

requisitos. Ademais, a "lista de parâmetros a considerar" é essencialmente qualitativa.

Em contrapartida as "matrizes-solução" permitem uma análise quantitativa, onde um designer, após especificar valores para a lista de "parâmetros a considerar", buscaria nas linhas das matrizes qual a solução de movimentação de materiais que melhor se adequa àquela combinação de parâmetros, com seus respectivos valores.

Com base na experiência profissional e na pesquisa realizada por parte dos autores, e nas metodologias parciais de "lista de parâmetros" e "matrizes-solução", a seguinte metodologia empírica foi apresentada no artigo:

- Definir a missão do CD (tipo de indústria, tamanho);
- Utilizar as matrizes para identificar tecnologias e soluções para contextos similares;
- Com o auxílio de gráficos de Pareto, dividir os níveis de atividade de armazenamento logístico e de *picking*
- Montar o design conceitual do CD com o auxílio de redes de fluxo
- Dimensionar, com o auxílio das ferramentas disponíveis, componentes como o recebimento, o envio, o armazenamento logístico e o empacotamento.

Os autores concluem mencionando que o trabalho pode ser expandido com futuras contribuições para as tabelas e listas de parâmetros, a fim de que possam incluir mais soluções, e levantam o questionamento se as academias podem endossar a abordagem empírica por eles apresentada.

2.2.3 Gu, Goetschalckx e McGinnis (2007)

Gu, Goetschalckx e McGinnis (2007) abordam os problemas de planejamento das operações de um CD, e iniciam descrevendo a importância dos mesmos, o aumento

da demanda que vêm sofrendo ao longo dos anos, e de como as novas tecnologias, como os sistemas WMS¹ e as etiquetas RFID², podem ser usadas para melhorar suas operações.

O objetivo do artigo, semelhante ao que Baker e Canessa (2009) realiza para a parte de design, é mostrar um resumo dos modelos existentes que se prestam a dar suporte às decisões operacionais tomadas em um CD. O artigo procura apresentar, assim, o estado-da-arte em pesquisa de planejamento operacional de centros de distribuição.

Inicialmente, o artigo define um framework para organizar as pesquisas referentes aos diversos, mas relacionados, problemas operacionais que podem ocorrer num CD. Em cada bloco do framework, isto é, em cada ponto de vista do CD, é indicado o número de artigos que foram avaliados que fazem menção àquela área de conhecimento.

Figura 12: Problemas de design de centros de distribuição, e número de ocorrência de publicações (GU; GOETSCHALCKX; MCGINNIS, 2007).

¹WMS: *Warehouse Management System*.

²RFID: *Radio-Frequency Identification*.

Na sequência, e para cada um dos pontos de vista do CD - recebimento, armazenamento logístico, *picking* e envio - o artigo sumariza todo o material de pesquisa e descreve-o do seguinte modo: (1) Informações que devem ser dadas; (2) Pontos a determinar; (3) Pontos sujeitos à restrições e análises de performance. Para citar um exemplo, formulam que as decisões básicas nas operações de recebimento e envio devem ficar descritas da seguinte forma:

- Dados:

1. Informações sobre a mercadoria que está por vir, como a data de entrega e seu conteúdo;
2. Informações sobre as demandas do cliente, como seus pedidos e tempo esperado de entrega;
3. Informações a respeito do layout das docas e da disponibilidade de equipamentos para movimentação de materiais;

- A determinar:

1. A reserva de docas para a carga, e descarga, de mercadorias;
2. O agendamento de equipamentos de movimentação de materiais para as referidas docas;

- Sujeito à análise de performance e à restrições:

1. Recursos necessários para completar todas as operações de recebimento e envio;
2. Níveis de serviço, como tempo total de ciclo, e tempo necessário para carga e descarga dos transportadores.
3. Throughput necessário para cada doca.

O artigo segue a mesma síntese para os demais pontos de vista do framework.

2.2.4 Ellinger (2012)

Ellinger (2012) menciona que as redes de *supply chain*, o que inclui centros de distribuição, estão expostos à rápidas mudanças de requisitos por parte de seus clientes. É apontada a necessidade de se ter um planejamento de intra-logística mais ágil, possivelmente indo até um sistema automático de monitoramento, que forneça antecipadamente as mudanças necessárias nos processos internos de um centro de distribuição, de modo que se possa acompanhar a demanda.

Ellinger (2012) centraliza seu trabalho no setor de *picking*, por ser este o que mais impacta nos custos de intra-logística, representando até 50% dos mesmos, e também o mais complexo.

Figura 13: Arquitetura de referência para sistema de *picking* (ELLINGER, 2012).

Para atacar a complexidade inerente aos sistemas de picking, e poder implementar um planejamento de intra-logística ágil, Ellinger (2012) concebe e apresenta uma arquitetura de referência (figura 13) para sistemas de *picking*, representada por estruturas de blocos e suas respectivas instâncias. Os blocos, análogos à linguagem SysML, mostram os diversos componentes dos sistemas de *picking*, desde zonas, passando por equipamentos de movimentação até unidades de reabastecimento.

2.2.5 Goetschalckx, Mital e Huang (2014)

Goetschalckx, Mital e Huang (2014) abordam uma classe de problemas conhecida como AP - *Assignment Problems* - no contexto de se apontar qual a localização ótima dentro de um sistema de armazenamento qualquer, para uma dada configuração. A intenção é realizar operações de armazenagem e recuperação incorrendo no menor custo possível para o sistema de armazenamento.

De acordo com o artigo, 3 níveis de decisões são tomadas durante o design de um sistema de armazenamento. A decisão de nível mais alto diz respeito a pontos como a tecnologia utilizada na movimentação das cargas unitizadas, no layout do sistema (pontos de entrada e saída, presença ou não de forward picking areas, etc.), presença de corredores cruzados e política de armazenamento (curva ABC). As decisões de segundo nível falam sobre a configuração do próprio sistema de armazenamento, como n° de corredores, nível de empilhamento e profundidade. Por fim, as decisões de terceiros nível se referem a aspectos táticos e operacionais do sistema de armazenamento, como o n° de equipamentos empregados para movimentação de materiais.

2.3 *Gap de pesquisa*

As publicações apresentadas nas seções anteriores, 2.1 e 2.2, representaram os materiais mais relevantes utilizados para a elaboração deste trabalho, o qual foi desenvolvido tendo a atenção de explorar os espaços de pesquisa por eles não abordados.

A totalidade dos artigos expostos seção 2.1, com exceção apenas de Limère et al. (2010), desenvolve linhas de pesquisa buscando, como objetivo principal, conceber modelos de referência orientados a objeto para centros de distribuição. Diferem na maneira como o fazem, conforme apresentado nos seus resumos, porém são unânimes em ressaltar as vantagens que tais modelos de referência podem fornecer para os processos de design de tais centros. Dentre as vantagens mais citadas, estão a possível

utilização dos modelos de referência para a realização de simulações de alternativas de solução, e também a criação de arquiteturas variantes para adaptação a novos cenários. Os métodos citados em seus trabalhos, entretanto, dizem respeito apenas à concepção do modelo de referência, e não à análise de alternativas de solução. Adicionalmente, os modelos se limitam a permanecer no nível do centro de distribuição, não detalhando seus sub-sistemas, como os de armazenamento logístico, por exemplo.

Limère et al. (2010) seguem linha distinta dos artigos da seção 2.1, no qual também concebem um modelo OO, porém específico para cálculo de custos de intra-logística, utilizando a linguagem OMG SysMLTM para representá-lo. Ainda em contraste com os demais artigos da seção, Limère et al. (2010) detalham a execução de simulações de seu modelo, e mostram a obtenção de resultados numéricos a partir do mesmo. Entretanto, seu modelo de custo é apresentado como um artefato isolado do restante do contexto de um sistema de armazenamento logístico, não ficando claro como seria o processo para integrá-lo a um modelo de referência do mesmo.

Os artigos apresentados na seção 2.2 apontam a ausência de metodologias padronizadas para se resolver o problema de design de centros de distribuição (BAKER; CANESSA, 2009), e procuram desenvolver seus próprios métodos e *frameworks* (James M.; MELLER; John A., 2010), (ELLINGER, 2012) e (GU; GOETSCHALCKX; MCGINNIS, 2007). Não obstante, o fazem sem a utilização de uma linguagem de modelagem padronizada, como a OMG SysMLTM, e, mais relevante, sem a aplicação dos princípios da MBSE, não dando o tratamento sistêmico aos centros de distribuição, pouco se afastando dos já criticados métodos e modelos *ad-hoc*.

Este trabalho, tendo como objetivo já mencionado seguir os princípios da Engenharia de Sistemas Baseada em Modelos (MBSE) para a construção de um método de seleção para soluções de armazenamento logístico, busca complementar os artigos citados nos seguintes pontos:

- Focar sua contribuição acadêmica abordando e detalhando o sub-sistema de armazenamento logístico em particular. Os artigos citados na seção 2.1 não o fazem para nenhum dos sub-sistemas dos centros de distribuição (recebimento, armazenamento, *picking*, *packing* nem envio). Já na seção 2.2, apesar de haver menções a sistemas de *picking* e armazenamento, as mesmas não são feitas no contexto da MBSE;
- Ter como objetivo não apenas a elaboração de um modelo de referência orientado a objetos, e expressá-lo na linguagem OMG SysML™, mas sim ir além e utilizá-lo dentro de uma metodologia que possa auxiliar na escolha de soluções, aos moldes do que fora realizado por Limère et al. (2010), mas não se limitando apenas a custos, e sim incluindo demais parâmetros relevantes aos sistemas de armazenamento logístico.

3 DESENVOLVIMENTO

O capítulo de Desenvolvimento deste trabalho será dividido em duas seções, cada qual tendo como foco a descrição detalhada de uma das etapas do método proposto *MTcsl* (capítulo 1, seção 1.2).

Na primeira seção, 3.1, será descrita a elaboração do modelo de referência paramétrico, *MDref*, para sistemas logísticos de armazenamento. Durante sua elaboração, etapas da metodologia OOSEM (apêndice B) foram aplicadas a fim de assegurar formalismo ao processo. Entretanto, a metodologia mencionada não será integralmente seguida, pois, tendo sido concebida para abarcar uma ampla gama de sistemas de engenharia, aplicar todos os seus passos tornaria o desenvolvimento deste trabalho um tanto enfadonho. Uma descrição completa da metodologia, com exemplos comentados de sua aplicação, pode ser conferida em (FRIEDENTHAL; MOORE; STEINER, 2012).

Na seção subsequente, 3.2, será mostrada a montagem de um estudo de viabilidade *TS* genérico, e como o mesmo se utiliza de *MDref* para, de forma automatizada, produzir o comparativo entre soluções de armazenamento.

3.1 Construção do modelo de referência

O modelo de referência *MDref* consistirá de um conjunto de diagramas expressos em linguagem SysML, construídos e armazenados dentro de uma ferramenta de software *CASE* disponível comercialmente. Em termos práticos, *MDref* se apresentará sob a forma de um arquivo de computador pertencente à ferramenta *CASE* escolhida,

podendo nela ser aberto e manipulado.

Os diagramas que formarão o modelo *MDref* deverão prover informações a partir de quatro pontos de vista fundamentais da linguagem SysML: estrutural, comportamental, de requisitos e de parâmetros (ver apêndice A, figura 80). Não estarão, entretanto, limitados quanto a quantidade ou tamanho, devendo tais diagramas ser tão numerosos quanto o necessário para se transmitir as informações de *MDref* de forma eficaz.

A construção do modelo *MDref* nesta seção será descrita com o auxílio de exemplos, tanto em texto quanto em imagens, provenientes de outros setores que não o logístico. O intuito é permitir analogias com demais sistemas de engenharia, como submarinos, telescópios, sistemas de segurança residenciais, dentre outros, e tornar mais fácil a compreensão das etapas de elaboração de *MDref*. Quase a totalidade do material usado como exemplos nesta seção foi extraído da revisão bibliográfica, e podem ser acessados através do capítulo Referências.

3.1.1 Atividades introdutórias

A elaboração do modelo de referência *MDref* deve se iniciar com um conjunto de atividades introdutórias, comuns a todos os projetos que se dispõem a trabalhar seguindo os princípios da MBSE, e que fazem uso de linguagens gráficas de modelagem, como a SysML. São elas:

1. Organização do modelo.
2. Definição do objetivo do sistema;
3. Definição do objetivo do modelo;

3.1.1.1 Organização do modelo

Primeiramente, segue a atividade de se estabelecer uma organização em pacotes para o modelo dentro da ferramenta *CASE* escolhida. A organização de um modelo é reconhecida por ser um dos aspectos mais críticos na sua elaboração, pois, se for falha, a complexidade do projeto pode rapidamente sobrecarregar os usuários, e o modelo se tornar intratável. A organização em pacotes de um modelo em SysML não vem a ser um artefato de projeto propriamente dito, mais sim apenas uma boa prática, ficando sua definição a critério do time responsável pelo projeto.

Este trabalho adota a organização proposta em (FRIEDENTHAL; MOORE; STEINER, 2012) e (R.KARBAN et al., 2011), no qual os pacotes seguem uma estrutura organizacional recursiva, baseada nos quatro pilares da linguagem SysML: requisitos, comportamento, estrutura e parâmetros. Alguns exemplos de organização podem ser analisados na figura 14.

3.1.1.2 Objetivo do sistema

A definição do objetivo do sistema, abreviado pela sigla *OBJsys*, apesar de, neste caso, já ter sido feita no início deste trabalho (seção 1.1.1), deve ser resumizada e transferida para dentro da ferramenta *CASE* utilizada. *OBJsys* deve ser concisa e clara, pois servirá de base de consulta a todos os envolvidos nas tomadas de decisão sobre a escolha do sistema, além de ser o requisito raiz para todos os demais requisitos de negócio. *OBJsys* deve ser elicitada em formato textual e apreendida como um requisito, a ser colocado dentro da árvore do modelo *MDref*, no pacote correspondente.

3.1.1.3 Objetivo do modelo

Formalizado o objetivo do sistema, deve-se definir então o objetivo do modelo *MDref* a ser construído para o mesmo, pois sendo um modelo uma abstração da re-

(a) Fonte: (PLEGT, 2011)

(b) Fonte: (PEARCE; FRIEDENTHAL, 2013)

(c) Fonte: (FRIEDENTHAL; MOORE; STEINER, 2012)

Figura 14: Exemplos de organização de modelos SysML

alidade para um dado sistema de interesse, poderão existir inúmeros modelos que o representam, possuindo distintas funcionalidades.

Neste trabalho, o objetivo de *MDref*, aqui denominado pela sigla *OBJmd*, é servir de referência para o maior número possível de soluções de armazenamento logístico, de modo que se possa representar uma solução qualquer através do ajuste de seus parâmetros e execução de suas equações. E, de forma análoga a *OBJSys*, *OBJmd* também deve ser elicitado em formato textual, apreendido como um requisito e colocado na árvore do modelo.

3.1.2 Elaboração do modelo

3.1.2.1 *Template* de requisitos e escopo do modelo

Através das atividades de pesquisa do tipo exploratória, explicadas no capítulo 1, seção 1.3 e que incluíram levantamento bibliográfico, entrevistas com especialistas e estudo de projetos bem-sucedidos, foi feita a definição do escopo do modelo e a formulação de uma lista-padrão, ou *template*, para seus requisitos.

O escopo do modelo *MDref*, referenciado pela sigla *SCPSys*, é representado por um BDD¹, que têm por finalidade delimitar a separação do sistema de interesse, aqui no caso, o sistema de armazenamento logístico, do ambiente ao seu redor. Tal diagrama BDD obrigatoriamente deve conter o bloco principal do sistema de interesse, participando de uma relação de composição com um segundo bloco de nível de abstração superior, o qual simboliza o contexto do referido sistema.

É usual a representação do sistema de interesse num contexto *SCPSys* que inclua a existência de mais sistemas adjuntos, representados por outros blocos, visando garantir que o modelo *MDref* a ser elaborado a partir do bloco principal seja coerente com seu meio, ainda que a modelagem dos demais sistemas não seja apresentada no trabalho. Adicionalmente, a definição precisa de um escopo evita que a criação do modelo *MDref* se sobreponha aos domínios dos sistemas adjuntos.

¹BDD - Diagrama de Definição de Blocos. Um dos diagramas utilizados pela linguagem SysML para representação estrutural de aspectos do sistema em questão

A figura 15 exibe um BDD contendo um exemplo de escopo de modelo para um sistema de satélite. Nele, nota-se o sistema de interesse *APE*, sob o estereótipo «system», ao lado de elementos adjuntos, dentre eles outros sistemas (*Atmosphere* e *Environment*), e atores (*Scientist* e *Engineer*). O bloco *APEContext* simboliza o escopo do modelo propriamente dito.

Figura 15: Exemplo de definição de escopo (R.KARBAN et al., 2011).

A lista-padrão, ou *template*, de requisitos TR, é montada após se percorrer n vezes as etapas de levantamento bibliográfico, entrevistas com especialistas e estudo de projetos bem-sucedidos (capítulo 1, seção 1.3). Tivessem essas etapas sido percorridas uma única vez, como seria no caso de um projeto P_x tratado isoladamente, o resultado seria uma listagem de requisitos LR exclusiva para o dado projeto P_x em questão. Porém, uma única amostra não seria suficiente para se ter uma lista que possa ser considerada como padrão de projeto. Portanto, ao se repetir n vezes as etapas mencionadas de pesquisa, chega-se a um conjunto $TR = \text{MÉDIA}(LR1 \dots LRn)$ de listagens de requisitos de n projetos, do qual se pode, através de uma média qualitativa, extrair aqueles requisitos que ocorrem maior número de vezes. O número de repetições n , ou seja, o número total de bibliografias estudadas, entrevistas realizadas e projetos analisados estará limitado pelos fatores de tempo e recursos financeiros disponibilizados ao pesquisador, cabendo a este a decisão do momento certo de interromper os trabalhos

de pesquisa.

Em resumo, TR busca agrupar numa única listagem todos os requisitos que têm maior probabilidade de ocorrer em projetos de sistemas logísticos de armazenamento. Seus requisitos representam, assim, as capacidades que tais sistemas logísticos devem oferecer.

Entretanto, por mais extenso que tenha sido o trabalho de pesquisa, e por maior que tenha sido o fator n , não é razoável supor que um único *template* TR seja o suficiente para abarcar todos os requisitos possíveis de projetos de sistemas logísticos de armazenamento. Essas imprevisibilidades são contornadas com *extensões* feitas sobre *MDref*, a fim de incluir em TR requisitos originalmente não previstos. Este procedimento de extensão, previsto na linguagem SysML, é explicado em detalhes na seção 3.2. Não obstante, mesmo tendo em vista a não-completude inerente de TR, será a partir da mesma que o modelo *MDref* começará a ser montado nas seções subsequentes. São exemplos de requisitos que podem figurar no *template* TR:

1. O sistema de armazenamento deverá dispor de quantidade adequada de endereços;
2. O sistema de armazenamento deverá ser capaz de atender às demandas de giro de estoque²;
3. O sistema de armazenamento deverá trabalhar com UoHs padrão de mercado;
4. O sistema de armazenamento deverá apresentar custos razoáveis de instalação e operação;
5. O sistema de armazenamento deverá levar em consideração eficiência operacional por turno, e implementação de distribuição A-B-C na alocação dos espaços de seu estoque;

²Também denominado pelo jargão técnico *throughput*, que implica no número máximo de UoHs que podem ser movidas num dado intervalo de tempo, usualmente diário ou mensal.

6. O sistema de armazenamento deverá dispor de meios próprios para movimentar, e armazenar, as mercadorias.

Os requisitos que irão compor TR, de forma análoga a *OBJSys* e *OBJmd*, devem ser elicitados em formato textual e transferidos para dentro da ferramenta *CASE* sob o formato de elementos de requisitos, e colocados dentro do pacote correspondente. Podem ser exibidos em Diagramas de Requisitos (REQ) da linguagem SysML, em formato tabular ou matricial, de acordo com as funcionalidades oferecidas pelo fabricante da ferramenta.

3.1.2.2 Requisitos não-funcionais, funcionais, de interface e premissas

A partir do *template* TR, obtido após n iterações do processo de pesquisa, derivam-se quatro conjuntos distintos de requisitos:

1. Requisitos não-funcionais (TRnf);
2. Requisitos funcionais (TRf);
3. Requisitos de interface (TRif);
4. Premissas e restrições (TPR).

Esta separação em conjuntos é relevante para a especificação da caixa-preta de *MDref*, como será mostrado na seção seguinte, 3.1.2.3.

Os quatro conjuntos mencionados são ditos derivados, e não simplesmente subconjuntos de TR, pois seus requisitos não são necessariamente cópias exatas, extraídas diretamente de TR. Para a montagem dos conjuntos TRnf, TRf, TRif e TPR, os requisitos que compõem o *template* TR devem ser interpretados, reescritos e redeclarados como novos requisitos, porém com um nível maior de especialização. Tais processos serão pormenorizados, e exemplificados, nos parágrafos subsequentes dessa seção, e

ressalta-se que TRnf, TRf, TRif e TPR também devem ser transferidos para dentro da ferramenta *CASE*, de forma semelhante a TR.

Requisitos não-funcionais, TRnf, normalmente estão relacionados a aspectos de performance do sistema em questão, e indicam os fatores limitantes aos quais ele estará sujeito ao realizar suas funções. Mais especificamente, tais requisitos definem valores para os quais as MOEs³ do sistemas deverão ser maiores/menores que, ou exatamente iguais a, para que o sistema como um todo seja considerado bem-sucedido. A título de exemplo, em se tratando de um automóvel, propriedades como velocidade máxima, consumo urbano, nível de ruído, espaço disponível e nível de emissões são exemplos de requisitos não-funcionais. Com relação ao sistema logístico de armazenamento, alguns dos requisitos pertencentes ao seu *template* TR podem ser reinterpretados como requisitos não-funcionais como se segue:

- 'O sistema de armazenamento deverá dispor de quantidade adequada de endereços' → O sistema de armazenamento deverá dispor de quantidade de endereços igual, ou superior, a especificada pelo cliente;
- 'O sistema de armazenamento deverá apresentar custos razoáveis de instalação e operação' → O sistema de armazenamento deverá apresentar custos de instalação e operação iguais, ou inferiores, aos especificados pelo cliente.

Requisitos funcionais, TRf, listam as operações que o sistema deverá desempenhar, sem fazer qualquer menção às questões de performance, ou seja, sobre o modo como ele fará para desempenhá-las. Retomando o exemplo do automóvel, operações como "trancamento automático de portas", ou "corte de combustível em acidente" são exemplos de requisitos funcionais. Comparativamente, sistemas de engenharia que fazem grande utilização de componentes de software, também denominados de *software-intensive*, tenderão a ter uma listagem elevada de requisitos funcionais, ao contrário do

³MOE: sigla em inglês para medida de efetividade (*measure of effectiveness*)

que se verifica em sistemas físicos, ou *hardware-intensive*, onde os requisitos não-funcionais é que ocorrem em maior quantidade. É um exemplo de requisito funcional para o sistema de armazenamento logístico, extraído de TR:

- 'O sistema de armazenamento deverá dispor de meios próprios para movimentar, e armazenar, as mercadorias' → O sistema de armazenamento deverá desempenhar a função de transportar UoHs do setor de recebimento de mercadorias, ou equivalente, até um endereço de armazenagem disponível.

Requisitos de interface, TRif, dizem respeito aos elementos que o sistema terá de trocar com seu meio *SCPs*_{sys}, sejam esses elementos de natureza material, energética ou de informação. Recepção FM, tomada de ar, bocal de abastecimento, portas da cabine e cano de escape são alguns dos requisitos de interface de um automóvel, para concluir seu exemplo. O sistema de armazenamento logístico possui, dentre seus requisitos de interface, os seguintes exemplos:

- 'O sistema de armazenamento deverá dispor de meios próprios para movimentar, e armazenar, as mercadorias' → O sistema de armazenamento deverá dispor de interface de comunicação para troca de informações sobre localização de armazenagem e retirada de UoHs;
- 'O sistema de armazenamento deverá trabalhar com UoHs padrão de mercado' → O sistema de armazenamento deverá dispor de interface para a entrada e a saída de UoHs.

Nos exemplos acima é possível notar que um mesmo requisito do *template* TR pode ser derivado em mais de um requisito nos conjuntos TRnf, TRf, TRif e TPR.

Por fim, o conjunto de premissas e restrições, TPR, tem como objetivo explicitar de quais pressupostos está se partindo para a criação do modelo *MDref* do sistema de

armazenamento logístico. Estas premissas e restrições indicarão para os usuários quais as limitações de *MDref*, de modo que possam aplicar o método *MTcsl* corretamente e obter como resultado sistemas de armazenamento que atendam às suas necessidades.

Na lista TPR devem constar premissas e restrições referentes aos requisitos de TR, bem como dos seus grupos derivados {TRnf, TRf, TRif} que, por qualquer razão:

- Não puderam ser completamente esclarecidos;
- Foram propositalmente simplificados a fim de atender a um cenário de um projeto Px qualquer, no qual o método *MTcsl* será aplicado;
- Foram diretamente impostos pelo cliente.

Idealmente, a lista TPR deve ser um conjunto vazio, pois cada premissa e restrição representa incertezas e limitações a cerca de um requisito do sistema a ser construído, podendo acarretar em custos extras e atrasos, tanto nas fases de construção quanto de testes.

Podem ser tidos como exemplos de premissas e restrições a figurarem em TPR as seguintes entradas:

- O sistema de armazenamento logístico deverá trabalhar com UoHs do tipo Pallet PBR-padrão;
- Não será considerada a distribuição A-B-C no armazenamento de mercadorias;
- Não poderão ser utilizados equipamentos à combustão para movimentação de materiais.

O primeiro item faz uma suposição referente a um dos exemplos de requisito listados em TR, "O sistema de armazenamento deverá trabalhar com UoHs padrão de mercado", na seção anterior. Como o requisito não especifica o tipo de padrão de UoH

a ser utilizado, a premissa estabelece que deverá se tratar do tipo Pallet PBR, um dos muitos padrões de pallet utilizados na indústria logística.

Já o segundo item da lista pressupõe uma simplificação sobre o requisito "O sistema de armazenamento deverá levar em consideração a implementação de distribuição A-B-C no armazenamento mercadorias". Essas simplificações têm como objetivo tornar menos complexa a construção de *MDref* e, conseqüentemente, a aplicação de *MTcsl*, desde que o projeto *Px* em questão assim o permita.

Por fim, o terceiro item restringe as instâncias de *MDref* apenas àquelas que envolvam empilhadeiras de tração elétrica, possivelmente devido à limitações do local de instalação relacionadas a nível de ruído e operação em locais fechados.

3.1.2.3 Especificação caixa-preta

Estabelecidos os subconjuntos TRnf, TRf, TRif e TPR, derivados do *template* TR, inicia-se a construção de *MDref* partindo-se de sua especificação caixa-preta.

Usa-se o termo caixa-preta porque, a princípio, nada se sabe a respeito do funcionamento sistema sendo modelado, apenas seu bloco principal, ainda vazio, e seu conjunto de requisitos e premissas TRnf, TRf, TRif e TPR.

A especificação caixa-preta inicia-se com a criação de um novo BDD, no qual deverá ser inserido apenas o bloco principal de *MDref*, aqui denominado de *Storage*. Inicialmente, e da mesma forma como ocorre com o BDD de escopo *SCPs_{sys}*, *Storage* é mostrado como um bloco opaco, exibindo apenas seu nome. Especificar sua caixa-preta significa, portanto, definir quais serão suas propriedades de valor, operações e portas⁴ (figura 16).

Cada uma das três categorias de requisitos, TRnf, TRf e TRif, dará origem a pro-

⁴Os termos *propriedades de valor*, *operações* e *portas* não foram aqui definidos ao acaso. São termos formalmente empregados pela linguagem SysML para identificar os diferentes tipos de atributos, métodos e interfaces que um bloco qualquer pode conter.

(a) Fonte: (PEARCE; FRIEDENTHAL, 2013)

(b) Fonte: (FRIEDENTHAL; MOORE; STEINER, 2012)

Figura 16: Exemplos de especificação caixa-preta

priedades distintas no bloco *Storage*, enquanto as premissas listadas em TPR irão balizar decisões de *design* de *MDref*, como a formulação de suas equações nos Diagramas Paramétricos, ou valores a serem atribuídos às suas instâncias. As relações entre os re-

quisitos dos grupos TRnf, TRf e TRif e as propriedades de valor, operações e portas do bloco *Storage* serão formalizadas através de relações conhecidas dentro da linguagem SysML como de *refinamento*⁵.

A partir dos requisitos não-funcionais TRnf, refinam-se as propriedades de valor do bloco *Storage*, as quais irão representar suas MOEs.

Requisitos funcionais dão origem às operações do sistema, o equivalente a métodos de uma classe caso se estivesse utilizando a linguagem UML. Podem ser posteriormente detalhados em Diagramas de Atividades pertencentes ao bloco *Storage*.

Por fim, requisitos de interface definem, no jargão da linguagem SysML, as portas do bloco *Storage*, pelas quais se modelam fluxos de elementos que o adentram e deixam, bem como as operações por ele disponibilizadas para atores externos.

É fundamental que cada um dos requisitos, não importando a qual dos grupos TRnf, TRf ou TRif pertença, esteja em pelo menos uma relação de refinamento com uma das propriedades da especificação caixa-preta do sistema. Um requisito que não refina nenhuma propriedade não tem porquê existir, e é permitido a um mesmo requisito participar do refinamento de duas, ou mais, propriedades de categorias distintas.

Assim como ocorre com Diagramas de Requisitos (REQ) em SysML, as relações de refinamento podem não ser tão bem visualizadas em Diagramas de Blocos (BDD), por questões de poluição visual. A notação matricial é a mais utilizada e, dependendo da ferramenta *CASE* empregada, permite-se a execução de filtros importantes, como requisitos que não possuem nenhuma relação de refinamento com nenhuma propriedade.

Concluído o refinamento de requisitos e a especificação da caixa-preta do bloco *Storage*, a construção do modelo de referência *MDref* pode prosseguir então para a etapa seguinte, que vem a ser sua decomposição estrutural. A decomposição estrutural

⁵Uma descrição completa de todos os tipos de relacionamentos em SysML, bem como o contexto de sua utilização, podem ser vistos em (FRIEDENTHAL; MOORE; STEINER, 2012)

do bloco *Storage* tem por objetivo definir quais serão os demais blocos que irão compor sua estrutura. Para se chegar a decomposição estrutural do bloco *Storage*, é preciso primeiramente conhecer a lista de blocos candidatos a fazer parte de tal. Esta lista é elaborada através de dois procedimentos denominados de *análise funcional* e *análise causal*, detalhados nas próximas seções.

3.1.2.4 Análise funcional

Figura 17: Exemplo de análise funcional (FRIEDENTHAL; MOORE; STEINER, 2012).

A análise funcional decompõe e detalha as operações pertencentes ao bloco *Storage*, o qual, até esta etapa, ainda consiste apenas numa representação caixa-preta do sistema de armazenamento logístico. Tais operações, conforme exposto na seção anterior, foram refinadas a partir dos requisitos funcionais TRf. No processo de decomposição, operações são quebradas em suas atividades básicas e busca-se identificar quem são os sujeitos de suas ações, isto é, quem são os blocos responsáveis por executá-las,

bem como os blocos-alvo daquelas mesmas atividades. Este processo deve ser feito em conjunto com um especialista que tenha conhecimentos a respeito do funcionamento do sistema, cabendo ao engenheiro de sistemas a transformação do conhecimento tácito apreendido em conhecimentos explícitos, através do processo de modelagem. No caso das operações, esta apreensão se dá dentro dos Diagramas de Atividades (ACT) da SysML. A figura 17 exhibe um exemplo de análise funcional simplificada para uma operação de *log on* num sistema de segurança, da qual se pode extrair *user interface* e *controller* como blocos candidatos à decomposição estrutural de *surveillance system*.

3.1.2.5 Análise causal

A análise causal realiza o rastreamento de todas as MOEs do bloco *Storage*, a fim de identificar de quais outras propriedades dependem, e como são calculadas. Quando expressa graficamente, a análise causal é conhecida também como Diagrama de Ishikawa, ou Espinha de Peixe. Assim como na análise funcional, a análise causal também deve ser realizada com o auxílio de um profissional especialista no sistema em questão.

Mais do que identificar as propriedades das quais as MOEs derivam, a análise causal irá apontar os donos de tais propriedades, contribuindo para a listagem de blocos pertencentes a decomposição estrutural do sistema. Junto a isso, serão conhecidas também as equações que usam as referidas propriedades para o cálculo das MOEs. Essas equações são artefatos importantes ao método *MTcsl* proposto neste trabalho, pois serão posteriormente executadas em ferramentas resolvedoras, a fim de realizar a simulação de soluções com valores numéricos e satisfazer a condição C_4 imposta ao método, na seção 1.2 Objetivo.

A figura 18 mostra um exemplo deste tipo de análise, extraído de (FRIEDENTHAL; MOORE; STEINER, 2012). Nele, é feita a análise causal de uma MOE hipotética denominada 'queda nas vendas' (*lack of sales*) referente a um sistema de segurança residencial. Todos os parâmetros que, de alguma forma, a influenciam são exibidos no

Figura 18: Exemplo de análise causal (FRIEDENTHAL; MOORE; STEINER, 2012).

diagrama estilo espinha-de-peixe. As equações são omitidas e substituídas por setas de ligação, enquanto sugestões podem ser feitas a respeito de quais blocos poderiam deter os parâmetros ali exibidos. 'Sistema de segurança' e 'Mercado' são blocos candidatos que poderiam ser extraídos diretamente da análise causal do exemplo da figura 18.

Este exemplo exibiu apenas os procedimentos de análise causal para a MOE hipotética 'queda nas vendas', sendo que os mesmos devem ser repetidos para cada uma das MOEs listadas na especificação de caixa-preta do bloco *Storage*. O resultado final para o modelo *MDref* completo certamente incluirá dezenas, senão centenas, de propriedades e equações. Assim, é interessante que a ferramenta *CASE* escolhida disponha de meios para aplicar filtros sobre os diagramas, a fim de produzir relatórios mais compreensíveis do que análises diretas sobre o modelo *MDref*.

Em SysML, a apreensão das análises causais dentro da ferramenta *CASE* se dá através dos diagramas paramétricos (PAR). Será nesses diagramas que ocorrerá a conexão entre as MOEs do bloco principal do sistema sendo modelado, as propriedades

das quais dependem e as equações que as calculam. Os diagramas paramétricos, ao contrário dos BDDs, precisam necessariamente pertencer a um bloco em particular, e devem ser organizados dentro do pacote correspondente na árvore do modelo.

Figura 19: Exemplo de análise causal, expressa em diagramas paramétricos (LLC, 2014).

A figura 19 exibe um segundo exemplo de análise causal, desta vez referente a um sistema de monitoramento por VANTs⁶, denominado de *LittleEye* (LLC, 2014). Neste segundo exemplo, é mostrado como as MOEs 'tripulação disponível' (*numberAvailableCrews*) e 'combustível disponível' (*numberAvailableFuelLoads*) são calculadas. Entretanto, ao contrário do diagrama de Ishikawa da figura 18, a figura 19 exibe a análise causal de forma quantitativa, explicitando as equações⁷ responsáveis pelo cálculo das MOEs. Também torna-se explícita a listagem de blocos (*Aircraft*, *Fuel* e *Crew*) que formam a decomposição estrutural do sistema de interesse *LittleEye*, pois são os blocos desta lista que contém as propriedades das quais as MOEs são derivadas. Em termos práticos, diagramas de Ishikawa se prestam como esboços ou relatórios, sendo as análises causais efetivamente montadas através dos diagramas paramétricos.

Uma observação com relação aos procedimentos de análise causal diz respeito até onde as relações de dependência entre as propriedades e equações devem ser mapeadas. Qualquer sistema pode ser infinitamente modelado até os seus detalhes mais elementares. Entretanto, como a função de um modelo é transmitir informações relevantes a respeito do sistema sobre o qual foi feito, o critério de parada no detalhamento

⁶Veículos aéreos não-tripulados.

⁷Em SysML, equações são representadas por blocos com o estereótipo «constraint».

deve refletir os interesses do cliente no referido sistema. Em outras palavras, um cliente interessado na aquisição de um sistema de armazenamento logístico completo, pode não precisar de detalhes acerca do funcionamento do sistema de elevação de garfos das empilhadeiras utilizadas. Assim, não faz sentido seguir com o detalhamento muito além das MOEs da empilhadeira, a não ser que o objetivo fosse o projeto do próprio equipamento. Esses critérios de parada devem ser observados para todas as propriedades durante as análises causais, a fim de se evitar modelagem excessiva e desnecessária.

3.1.2.6 Decomposição estrutural

Concluídas as análises funcional e causal, processos que podem requerer algumas iterações, já se tem formada a lista de blocos que farão a composição estrutural do modelo de referência *MDref* para o sistema de armazenamento logístico.

A partir deste ponto, o bloco *Storage* não mais representa apenas uma caixa-preta do sistema de armazenamento, pois agora já são conhecidos os blocos que o compõem, as propriedades de valor dos mesmos e, sobretudo, as equações que relacionam as propriedades de valor dos blocos constituintes com as MOEs do bloco *Storage*.

Em SysML, os blocos que formam a composição estrutural de um outro bloco são definidos como sendo suas propriedades de parte, ou de referência. A diferença entre esses dois tipos de propriedades dependerá da relação entre o ciclo de vida do bloco principal e seus constituintes (ver próxima seção).

A figura 20 mostra, de forma simplificada, a decomposição estrutural para o mesmo exemplo do sistema de monitoramento por VANTs, *LittleEye*, adotado na seção anterior, 3.1.2.5 - Análise Causal, na figura 19 especificamente. No referido exemplo, é possível notar o bloco principal, *LittleEyeSystem* e seus blocos constituintes. As operações e portas dos blocos foram suprimidas nessa visualização.

Figura 20: Exemplo de decomposição estrutural de um sistema de interesse (LLC, 2014).

3.1.2.7 Classificação de parâmetros e relações

Após a definição da decomposição estrutural do bloco *Storage*, ainda ficam por determinar dois aspectos para que se possa dar como concluída a construção de *MDref*:

1. A classificação das propriedades de valor do bloco de *Storage* e dos seus blocos constituintes como: 'independentes', 'derivadas' ou 'somente-leitura';
2. O tipo de relação que os blocos listados terão entre si, e com o bloco de *Storage*.

A classificação das propriedades de valor dos blocos de *MDref*, tanto do bloco de *Storage* quanto de seus constituintes, tem por finalidade identificar as propriedades que, durante a execução de um estudo de viabilidade *TSx* qualquer:

- Poderão ser manipuladas livremente a fim de simular diversos cenários (propriedades 'independentes');
- Não poderão ser alteradas, devendo ser definidas apenas no momento da instânciação dos blocos de *MDref* (propriedades 'somente-leitura');
- Serão resultados de equações, tendo sua definição dependendo exclusivamente de outras propriedades (propriedades 'derivadas').

Através da atribuição de valores às propriedades ditas 'somente-leitura', e da variação das propriedades 'independentes' dentro de um intervalo pré-definido, pode-se executar o equacionamento contido nos diagramas paramétricos de *MDref* para se calcular as propriedades 'derivadas', e produzir assim o comparativo entre as possíveis soluções de armazenamento para um dado projeto *Px*. Ressalta-se que as propriedades de valor classificados como MOEs podem pertencer a qualquer uma das três classificações apresentadas, porém sendo usualmente classificadas como 'derivadas'.

A montagem dos estudos de viabilidade TSx, os quais dependem de como é feita a classificação das propriedades de *MDref* para produção dos comparativos entre soluções, é objeto de estudo da próxima seção, 3.2. Nesta, será dado foco a maneira como tais propriedades são efetivamente classificadas.

Formalmente, seja *CAmoe-x* uma análise causal completa para uma MOE *x* qualquer, construída conforme mostrado na seção 3.1.2.5, e como no exemplo na figura 18. Seja *CAmoe* o conjunto de todas as análises causais realizadas para um sistema de interesse. Assim, $CAmoe = \{CAmoe-1, CAmoe-2, \dots, CAmoe-n\}$.

O conjunto de análises causais *CAmoe* pode ser considerado um grafo direcionado acíclico (GDA). Um GDA é um tripla ordenada do tipo (N, A, g) onde:

- N = um conjunto de vértices
- A = um conjunto de arestas
- g = uma função que associe a cada aresta a um par ordenado (x, y) de vértices, onde x é o ponto inicial e y é o ponto final de A .

Ao fazer a transposição da definição de GDA para o domínio da SysML, mais especificamente para os elementos da referida linguagem que compõem os diagramas paramétricos, tem-se que:

- N representa o conjunto de propriedades de valor de *MDref*, MOEs inclusas;

- A representa o conjunto de blocos do tipo «*constraint*»;
- g representa as equações a serem escritas dentro dos blocos «*constraint*», e que irão relacionar os diversos pares (x, y) de propriedades de valor, onde x representará as propriedades de entrada das equações contidas nos blocos «*constraint*», e y será a propriedade calculada do mesmo bloco.

Seja N' o subconjunto de N para o qual não há funções g onde seus elementos são pontos finais das arestas contidas em A. Em outras palavras, N' representa o conjunto de todas as propriedades de valor de um modelo em SysML que não são da categoria 'derivadas', isto é, cujos valores não são calculados por nenhum bloco «*constraint*». Assim, N' conterá somente propriedades de valor das categorias 'somente-leitura' e 'independente', representadas pelos subconjuntos: $N'sl$ e $N'ind$. Para se preencher os conjuntos $N'sl$ e $N'ind$, deve-se percorrer cada um dos elementos de N' e avaliar se seus valores podem ser alterados por decisões de design, ou se são dependentes de fatores externos, além do alcance da equipe responsável pelo projeto em questão. Propriedades como, por exemplo, duração de uma bateria, velocidade máxima de avanço de uma empilhadeira ou o custo de iluminação por unidade de área são exemplos de propriedades que não podem ser modificadas, devendo ser marcados como 'somente-leitura' e colocadas em $N'sl$. Tais propriedades decorrem sobretudo do uso de COTS⁸. O restante das propriedades de N' que não se encaixa em $N'sl$ devem ser consideradas como da categoria 'independente', pertencentes ao conjunto $N'ind$. Uma vez montado o conjunto $N' = \{N'sl + N'ind\}$ as propriedades 'derivadas' podem ser obtidas da subtração $N - N'$. A figura 21 mostra um exemplo de grafo direcionado acíclico (GDA), onde as propriedades de valor A, B e H pertencem ao conjunto N' , e as restantes são categorizadas automaticamente como 'derivadas'.

Concluída a classificação das propriedades de valor de $MDref$, as mesmas devem

⁸COTS: sigla, em inglês, para componentes de prateleira (*Commercial off-the-shelf*), para os quais não há atividade de design envolvida, pois são adquiridos já prontos de fornecedores. Assim, seus parâmetros não podem ser modificados.

Figura 21: Exemplo de GDA representando categorias de propriedades de valor em SysML

ser devidamente identificadas dentro dos blocos as quais pertencem. Isso se deve não apenas para simplificar a análise visual de *MDref*, como também para permitir a geração automatizada de relatórios, dependendo das funcionalidades oferecidas pela ferramenta *CASE* empregada.

Propriedades de valor derivadas devem receber o prefixo *'/'* junto a seus nomes, dentro do bloco que as contém. Propriedades somente-leitura recebem, de forma análoga, o sufixo *'{ReadOnly}'*. Propriedades independentes não necessitam de identificação.

Por fim, as relações entre os blocos que compõem *MDref* podem ser de *composição* ou de simples *associação*. O bloco de *Storage* terá relações ditas de composição com todos os demais blocos cujo ciclo de vida dependerem deste, os quais se definirão como sendo suas propriedades de parte. Em contrapartida, relações de associação serão usadas quando o ciclo de vida dos blocos listados não tiver qualquer dependência com o ciclo de vida do bloco *Storage*, dos quais se definem as propriedades de

(a)

(b)

Figura 22: Exemplo de modelo de referência (*MDref*) finalizado (LLC, 2014).

referência.

3.1.2.8 Finalização do modelo de referência

Uma vez definidas as relações entre o bloco de *Storage* e seus blocos constituintes, e a categorização de suas propriedades de valor, o modelo de referência *MDref* para o

sistema de armazenamento logístico pode ser dado como concluído.

O modelo de referência *MDref* deve ser visualizado em um novo diagrama BDD, o qual recebe o nome de 'árvore do produto', ou, alternativamente, o nome do próprio sistema de interesse. A figura 22 mostra um modelo de referência simplificado para o exemplo do sistema de monitoramento por VANTs *LittleEyeSystem*, citado nas seções anteriores. Nela, podem ser identificados o bloco principal em relações de composição com seus blocos constituintes, junto com o diagrama paramétrico que relaciona todas as suas propriedades de valor. Porém, para fins de organização do diagrama e minimização da poluição visual, algumas das propriedades, operações e portas foram ocultadas. Os prefixos '/' e 'ReadOnly' das propriedades de valor também foram omitidos. Opcionalmente, se a complexidade do sistema assim demandar, pode-se estereotipar alguns dos blocos constituintes de acordo com o subsistema do qual fazem parte, como «eletrônico» ou «mecânico», a fim de facilitar a legibilidade do modelo.

3.2 Construção dos estudos de viabilidade

Estudos de viabilidade, também conhecidos pelo termo em língua inglesa *trade-studies*, abreviados pela sigla *TSx*, são estudos que procuram identificar, dentro de um conjunto de soluções candidatas, qual a solução mais adequada para um problema. Os estudos de viabilidade avaliam as soluções candidatas através de critérios tais como, e não limitados a, custo, performance, peso, dimensões e complexidade.

O objetivo de um estudo de viabilidade *TSx* será produzir um comparativo de soluções de armazenamento logístico, que sejam candidatas a atender a uma lista de requisitos LR, elicitada de um projeto *Px* qualquer.

O estudo de viabilidade *TSx* deverá produzir o comparativo entre soluções de armazenamento através de sucessivas execuções das equações contidas nos diagramas paramétricos de *MDref*. A cada execução, parâmetros de entrada de *TSx* deverão ser

gradativamente incrementados, ou decrementados, a fim de produzir alterações nos parâmetros de saída do mesmo estudo. O conjunto de tais parâmetros, de entrada e de saída, de *TSx* deverá ser idêntico às MOEs de *MDref*. A definição de quais MOEs deverão fazer o papel de parâmetros de entrada, e quais de saída, será um assunto a ser tratado nas seções 3.2.1.1 e 3.2.2.1.

De forma análoga ao modelo de referência, *MDref*, detalhado na seção anterior 3.1, os estudos de viabilidade *TSx* também consistem de um conjunto de diagramas expressos em SysML, construídos e gerenciados pela mesma ferramenta de software *CASE* selecionada para a elaboração de *MDref*. Tais estudos, por fins de praticidade e organização, podem ser armazenados no mesmo arquivo de computador da ferramenta *CASE* onde se encontra *MDref*, tornando assim mais práticas suas montagens e execuções.

As seções subsequentes irão descrever o processo de montagem dos estudos de viabilidade, e a maneira como os mesmos utilizam as equações de *MDref* para produzir o comparativo de soluções de armazenamento logístico.

3.2.1 Atividades introdutórias

3.2.1.1 Elicitação de requisitos do cliente

A montagem de um estudo de viabilidade *TSx* tem início na necessidade de se comparar possíveis soluções de armazenamento, que tenham condições satisfazer os requisitos de um projeto *Px* em questão.

Assim, a etapa inicial da montagem de *TSx* consiste na construção da lista de requisitos LR, elicitada do projeto *Px*. Ao contrário do *template* TR, que fora construído, na seção anterior, através de sucessivas entrevistas a especialistas, estudos de projetos concluídos e leituras bibliográficas da área, LR deverá ser montada com as informações provenientes de um único cliente, e deverá refletir os anseios do mesmo com relação

ao projeto Px do sistema de armazenamento logístico desejado.

Mesmo tendo sido montada com informações provenientes de uma única fonte, isto é, o cliente, LR também deverá passar pelo mesmo processo de TR, o qual consiste na derivação dos subconjuntos de requisitos não-funcionais, funcionais, de interface e premissas.

Uma vez feita a derivação de LR nos subconjuntos mencionados, deve ser verificado se é possível realizar o mapeamento dos seus requisitos para aqueles listados no *template* TR. Neste processo de verificação de mapeamento, que irá apontar a necessidades de se estender, ou não, *MDref*, três situações podem ocorrer:

1. Todos os requisitos de LR possuem equivalência no *template* TR, e há mais requisitos no mesmo do que em LR. Nesta situação, *MDref* não precisa ser estendido, e os requisitos de TR que não são endereçados por LR, caso pertençam ao subconjunto TRnf, são candidatos a serem os parâmetros variáveis do estudo de viabilidade TSx , para o projeto Px em questão;
2. Todos os requisitos de LR possuem equivalência no *template* TR, e todos os requisitos do mesmo são endereçados pelos requisitos listados em LR. Nesta situação, *MDref* não precisa ser estendido, porém deverão ser selecionados requisitos do subconjunto TRnf a servirem como os parâmetros variáveis do estudo de viabilidade TSx , para o projeto Px em questão. O cliente deverá ser consultado a respeito de quais requisitos não-funcionais possuem menor peso, de modo que, mesmo já tendo sido especificados, poderiam ter sua variação analisada a fim de tornar o projeto viável;
3. Nem todos os requisitos de LR possuem equivalência no *template* TR. Nesta situação, *MDref* precisa ser estendido, pois o cliente levantou necessidades no projeto Px em questão que não foram previstas no *template* TR original. A extensão de *MDref*, detalhada seção 3.2.1.2, deverá levar LR à situação de número

1 ou número 2 dessa listagem.

(a) Situação 1 - *MDref* não precisa ser estendido, com as propriedades D, E e F de TRnf candidatas naturais a parâmetros do estudo de viabilidade

(b) Situação 2 - *MDref* não precisa ser estendido, com pelo menos uma das propriedades A, B ou C de TRnf a ser escolhida como candidata a parâmetro do estudo de viabilidade

(c) Situação 3 - *MDref* precisa ser estendido, pois LR não abrange TR em sua totalidade.

Figura 23: Correspondência entre *template* e requisitos de projeto.

Em resumo, as situações listadas afirmam que não podem existir requisitos em LR que não sejam mapeáveis em TR, qualquer que seja o subconjunto considerado. Caso esta condição não se verifique, *MDref* deve ser estendido. Em contrapartida, TR poderá ser mais abrangente do que LR sem que *MDref* tenha que ser modificado.

Formalmente, sejam $T = \{TRnf, TRf, TRif, TPR\}$ e $L = \{LRnf, LRf, LRif, LPR\}$ os conjuntos de requisitos não-funcionais, funcionais, de interface e premissas derivados de TR e de LR, respectivamente. E seja f uma função hipotética que faça a correspondência semântica entre os conjuntos de requisitos de L e T, ou seja, $f: L \rightarrow T$ (figura 23).

Se, para pelo menos um requisito de L, qualquer que seja o subconjunto ao qual pertença, f for uma *função parcial*, então *MDref* deverá ser estendido. Tal procedi-

mento deverá ser feito antes da montagem de TSx, conforme descrito na seção seguinte, 3.2.1.2.

A listagem também explica que os estudos de viabilidade TSx serão conduzidos através de variações em parâmetros extraídos do subconjunto de requisitos não-funcionais TRnf. Isto se deve ao fato desses requisitos representarem as MOEs, ou medidas de efetividade, do sistema de armazenamento logístico, cujos valores a elas atribuídos num dado cenário irão informar o quão bem, o mal, o referido sistema está desempenhando em um dado projeto *Px*. Tais requisitos foram refinados em propriedades de valor no bloco principal, *Storage*, conforme descrito na seção 3.1.2.3.

3.2.1.2 A extensão do modelo de referência

Na seção 3.1.2, mais especificamente na subseção 3.1.2.1, fora afirmado que não se pode ter um único modelo de sistema de armazenamento logístico que seja o suficiente para abranger todas as realidades de projeto. Tal como ocorre com qualquer sistema de engenharia. Por mais que se mapeie os requisitos de projeto mais frequentes em *templates* pré-definidos, sempre haverá situações em que clientes irão demandar do sistema características não previstas inicialmente. Em outras palavras, é alta a probabilidade de que a listagem de requisitos LR para um projeto *Px* qualquer contenha, pelo menos, um requisito que não conste num *template* TR tido como padrão.

Nas seções acima citadas, também foi mencionado que estas situações são resolvidas com *extensões* feitas sobre o modelo de referência *MDref*.

Estender um modelo de referência *MDref* implica em alterar o *template* de requisitos TR que lhe deu origem, incluindo no mesmo os novos requisitos elicitados do projeto *Px* em questão. Na sequência, todas as etapas de elaboração de *MDref* (seção 3.1.2), desde a montagem do seu novo *template* TR' contendo os novos requisitos, até a sua decomposição estrutural e classificação de novos parâmetros e relações, devem ser repetidas.

Porém, não seria prático ter de refazer inteiramente um *MDref* apenas para abarcar, por vezes, uns poucos novos requisitos. Fato ainda agravado pela frequência, não tão baixa, com que esses novos requisitos surgem, e devem ser incorporados ao *template* TR.

Felizmente, o retrabalho que seria necessário para se criar um *MDref* completamente novo pode ser minimizado através do emprego de um tipo especial de relação, suportado pela linguagem SysML, conhecido como *generalização*⁹.

As relações de generalização ocorrem entre pares de blocos em SysML, onde um deles, usualmente denominado como bloco-filho, herda todas as características do outro, identificado como bloco-pai. A relação de generalização é, portanto, uma relação direcional, apontando do bloco-filho para o bloco-pai. Por herdar todas as características, entende-se que o bloco-filho irá receber, automaticamente, todas as propriedades de valor, parte e referência do bloco-pai, bem como suas portas, operações e equações (*constraints*).

Ao bloco-filho também é permitido realizar alterações nas propriedades herdadas, bem como criar novas que antes inexistiam no bloco-pai. E é precisamente esta característica, inerente às relações de generalização, que permite a extensão de *MDref* em um novo modelo de referência, *MDref'*.

Uma vez definido o novo *template* TR', cria-se um novo bloco para representar a caixa-preta do modelo estendido *MDref'*. Este novo bloco, *Storage'*, deverá entrar numa relação de generalização com o bloco original *Storage*, herdando todas as suas propriedades, operações e portas.

O restante das etapas de elaboração de *MDref'* são análogas às descritas para *MDref*. Os requisitos elicitados em TR' devem passar pelo mesmo processo de derivação, descrito na seção 3.1.2.2, em conjuntos de não-funcionais, funcionais, de interface

⁹Em linguagem UML também conhecido como *herança*.

e premissas. Entretanto, não é necessário analisar TR' em sua totalidade, mas apenas nos novos requisitos que o diferem de TR . Assim, a especificação caixa-preta (seção 3.1.2.3) para $MDref'$ implicará no refinamento apenas desses novos requisitos, resultando em novas propriedades de valor (novas MOEs), novas operações e novas portas em $Storage'$, ou mesmo na alteração das já existentes em $Storage$ através de operações de *redefine*.

As etapas de análise funcional e causal deverão ser feitas para as novas operações e propriedades de valor, a fim de identificar mudanças na maneira de se calcular as MOEs do sistema, bem como o surgimento de novos blocos candidatos a serem acrescentados na sua composição estrutural.

Por fim, e se necessário, uma nova decomposição estrutural para $MDref'$ deve ser especificada a partir da nova lista, caso a mesma exista, de blocos constituintes candidatos, seguido também de uma nova classificação de parâmetros e relações.

Assim, concluídas as etapas de elaboração de $MDref'$, tem-se no mesmo um novo modelo de referência para sistemas de armazenamento logístico, estendido de $MDref$, porém capaz de ser utilizado em estudos de viabilidade TSx para o projeto Px em questão, cuja lista de requisitos LR agora pode ser completamente mapeada no *template* de requisitos TR' .

A figura 24 mostra exemplos do emprego da relação de generalização em modelo SysML, com novas propriedades sendo criadas, e antigas sendo redefinidas nos blocos-filho.

Um ponto interessante referente ao uso desse tipo de relação está na capacidade de se criar famílias de modelos estendidos $MDref'$, também chamadas de arquiteturas variantes (figura 24b), especializadas em determinados cenários de projeto que têm ocorrência comum. Um exemplo seria uma arquitetura variante, $MDref'HVAC$, estendida de $MDref$ a partir de um *template* de requisitos TR' para sistemas de armazenamento

(a) Fonte: (PEARCE; FRIEDENTHAL, 2013)

(b) Fonte: (FRIEDENTHAL; MOORE; STEINER, 2012)

Figura 24: Extensões de modelos em SysML

frigorificados. Dessa forma, sempre que o projeto P_x em questão envolver esse tipo de sistema logístico, a equipe responsável poderá recorrer diretamente a $MDref'HVAC$ para a seleção de soluções, incorrendo em menos esforços de extensão do que em $MDref$, pois a maior parte dos requisitos relacionados a ambientes com temperatura

controlada em sistemas logísticos já terá sido previamente considerada no processo de extensão para $MDref'HVAC$.

3.2.2 Elaboração do estudo de viabilidade

Concluídas as atividades iniciais de elicitação dos requisitos para o projeto Px em questão, e, se necessário, de extensão de $MDref$ para se adequar a referida listagem, o estudo de viabilidade TSx poderá ter sua montagem iniciada.

3.2.2.1 Parâmetros de entrada e de saída

A montagem propriamente dita do estudo de viabilidade TSx se inicia com a definição de quais MOEs servirão como parâmetros de entrada, e quais como parâmetros de saída. Os estudos de viabilidade têm como foco as MOEs, propriedades de valor refinadas dos requisitos não-funcionais (TRnf), pois são elas que determinam o quão bom está sendo o desempenho de um dado sistema em análise. Os demais requisitos, refinados em operações e interfaces do sistema em questão, não participam dos estudos de viabilidade pois não há como incrementá-los ou decrementá-los, visto que não representam valores numéricos. Alterá-los implicaria basicamente na realização de uma extensão de $MDref$.

Na seção 3.2.1.1, foi mencionado que os parâmetros, tanto de entrada quanto de saída, de TSx são extraídos dos requisitos pertencentes ao conjunto dos não-funcionais (TRnf, ou, simplesmente, MOEs), do *template* TR. Assim, uma parte dos requisitos deste conjunto deverá ser separada como parâmetros de entrada, e a outra parte, como de saída.

Tal separação será determinada pelo subconjunto LRnf derivado de LR para o projeto Px em questão. Conforme as figuras 23a e 23b, expostas na subseção 3.2.1.1, poderão ocorrer duas situações: a primeira onde LRnf não endereça completamente TRnf; e a segunda quando TRnf é completamente endereçada por LRnf.

Na primeira situação, o cliente deixa em aberto certas MOEs para o sistema de armazenamento, não definindo valor algum para as mesmas. Isto se deve a diversos fatores, como a não obrigatoriedade a valores específicos de tais MOEs para o projeto de seu interesse, ou ao simples desconhecimento do cliente sobre a existência dessas MOEs.

Porém, na segunda situação o cliente define, em seu projeto Px , valores específicos de performance para todas as MOEs previstas no subconjunto TR_{nf} do *template* TR . Dessa forma, não seria realizável o estudo de viabilidade TSx , pois não haveria parâmetros de entrada que pudessem ser variados. Assim, o cliente deve ser consultado a fim de apontar quais dos requisitos por ele definidos em LR_{nf} teriam menor peso ou relevância, e cuja variação poderia ser considerada num estudo TSx , a fim de se possa produzir um comparativo de soluções para o projeto Px como um todo.

3.2.2.2 Modelagem dos requisitos

Uma vez feita a divisão das MOEs de MD_{ref} entre parâmetros de entrada e de saída, ambos devem ser modelados dentro de TSx , e isso se dá de formas distintas para cada um. Os parâmetros de entrada serão apreendidos como propriedades de valor dentro do bloco que representará o estudo de viabilidade TSx , cujos detalhes de construção serão vistos na seção 3.2.2.3. Já os parâmetros de saída, os quais representam os requisitos LR_{nf} do projeto Px em questão, serão transferidos para dentro de um bloco separado do bloco principal de TSx , onde cada um de seus requisitos também irá ser representado por uma propriedade de valor.

Esta etapa de modelagem de requisitos se faz necessária em blocos pois, em SysML¹⁰, os elementos de requisitos guardam apenas informações textuais, e não valores numéricos. Adicionalmente, ter os requisitos do projeto Px em questão apreendidos em um bloco separado da estrutura de TSx torna mais simples sua substituição por outro *set*

¹⁰Versão 1.4 da linguagem SysML, utilizada neste trabalho.

de requisitos, bastando substituir o bloco por completo, bem como sua reutilização em outros estudos de viabilidade.

A figura 25 mostra um exemplo de uma lista de requisitos LRnf apreendida dentro de um bloco em SysML. O elemento da direita representa uma instância do referido bloco, para a qual se pode efetivamente atribuir valores aos requisitos especificados pelo cliente. As etapas de instanciação, tanto dos blocos constituintes de *MDref* quanto de *TSx* serão vistas na seção 3.2.2.5.

Figura 25: Modelagem de requisitos utilizando blocos.

3.2.2.3 Diagrama de blocos

Conforme mostrado nos parágrafos iniciais desta seção, os estudos de viabilidade *TSx*, similarmente ao modelo de referência *MDref*, também consistem num conjunto de diagramas SysML, tratados pela mesma ferramenta *CASE*. E tal como ocorre com *MDref*, os estudos de viabilidade *TSx* também possuem uma composição estrutural, isto é, têm sua estrutura formada por blocos constituintes, ou, melhor dizendo, propriedades de parte e referência para se ater ao jargão da linguagem SysML. Tais blocos devem ser inseridos num diagrama BDD específico para o *TSx* em questão, e colocado dentro da estrutura de pacotes do projeto.

A estrutura de blocos de um estudo de viabilidade deverá conter, ao menos, três blocos distintos:

- Um bloco para representar o próprio estudo *TSx* em questão, que deverá receber

seu nome;

- Um bloco para representar o sistema sendo analisado. Neste caso, trata-se do bloco principal de *MDref*, *Storage*;
- Um bloco para representar os requisitos não-funcionais que serão avaliados no estudo de viabilidade *TSx* (seção anterior, 3.2.2.2, figura 25).

Além dos três blocos estruturais descritos na listagem anterior, os estudos de viabilidade *TSx* possuirão também blocos de equação (*constraint blocks*) exclusivos, responsáveis por funções como relacionar seus parâmetros de entrada com os de saída, e calcular margens de segurança para cada um dos requisitos testados ('valor requerido' - 'valor calculado'). Tais blocos usualmente são mostrados apenas na árvore de pastas do projeto, e em seus respectivos diagramas paramétricos, sendo sua exibição no BDD do estudo de viabilidade algo opcional. Serão abordados na seção 3.2.2.4.

O bloco principal de *TSx* deverá conter dois grupos de propriedades de valor: o primeiro será de propriedades do tipo 'independente', e o segundo do tipo 'derivadas'

¹¹.

As propriedades independentes, no primeiro grupo, corresponderão aos parâmetros de entrada do estudo de viabilidade *TSx*, conforme explicado na seção 3.2.2.1. São apreendidos como propriedades independentes pois esses parâmetros de entrada terão de ser gradualmente incrementados, ou decrementados, a fim de que *MDref* possa produzir o comparativo entre as soluções.

Já as propriedades derivadas, no segundo grupo, corresponderão aos parâmetros de saída do estudo, recebendo o resultado da comparação entre os valores apreendidos no bloco de requisitos (valores requeridos), contra os valores das MOEs calculadas por *MDreq* (valores calculados). Essas propriedades derivadas são normalmente expressas em termos percentuais, indicando, para uma dada iteração do estudo em questão, o

¹¹ver seção 3.1.2.7 sobre classificação de propriedades de valor.

quão bem, ou mal, um requisito está sendo atendido. Esse segundo grupo de propriedades, naturalmente, deverá espelhar as propriedades do bloco de requisitos.

Uma vez inseridas todas as propriedades de valor necessárias ao bloco principal do estudo de viabilidade *TSx*, o mesmo deverá entrar em relações de agregação com os blocos de requisitos, e com o bloco principal de *MDref*. A relação utilizada é do tipo *agregação* pois os ciclos de vida desses blocos não dependem uns dos outros, podendo o bloco de requisitos, bem como o principal de *MDref*, serem usados em outros estudos de viabilidade.

A figura 26 mostra duas imagens de diagramas de blocos para estudos de viabilidade. Estes exemplos não seguem por completo a descrição feita neste trabalho, porém ilustram com clareza como tais estudos devem ser montados.

A primeira imagem, 26a, mostra um estudo de viabilidade construído para o sistema de monitoramento por VANTs, já usado como exemplo nas seções 3.1.2.6 e 3.1.2.8. Nele, é possível identificar o bloco correspondente ao estudo de viabilidade, *CoverageAnalysis*, bem como o bloco principal do *MDref* do sistema de monitoramento, *LittleEyeSystem*, onde optou-se por exibir suas propriedades de parte. Em *CoverageAnalysis*, há um parâmetro de saída, marcado como derivado ('/'), e dois parâmetros de entrada, cuja ausência de marcação indica serem independentes. Neste primeiro exemplo ocultou-se o bloco de requisitos.

A segunda imagem, 26b, mostra um estudo de viabilidade feito para comparar dois modelos de solução de monitoramento noturno por câmeras. O estudo deste exemplo utiliza dois modelos de referência simultaneamente (blocos *Low-Light Camera* e *Camera with Light*), bem como duas instâncias do mesmo bloco de requisitos, aqui este-reotipado como «objectiveFunction». Neste caso, há apenas dois parâmetros de saída, e nenhum de entrada, sendo este estudo desenhado para ser executado apenas uma vez, a produzir um comparativo para um único cenário de utilização das câmeras.

(a) Fonte: (LLC, 2014)

(b) Fonte: (FRIEDENTHAL; MOORE; STEINER, 2012)

Figura 26: Exemplos de montagens de estudos de viabilidade

3.2.2.4 Equacionamento

Na seção anterior, 3.2.2.3, fora mencionado que os estudos de viabilidade, TS_x , devem possuir seus próprios blocos de equações, além daqueles que já estão contidos no modelo de referência $MDref$ analisado no estudo. Conforme foi apontado, tais blocos de equação são responsáveis por funções como relacionar os parâmetros de entrada de TS_x com os de saída, e calcular margens de segurança para cada um dos

requisitos testados de *MDref*.

(a) Fonte: (FRIEDENTHAL; MOORE; STEINER, 2012)

(b) Fonte: (FRIEDENTHAL; MOORE; STEINER, 2012)

Figura 27: Equacionamento dos estudos de viabilidade

O relacionamento dos parâmetros de entrada com os de saída é feito através de equações que invertem a ordem dos cálculos das MOEs de *MDref*. Melhor detalhando, *MDref* conta com propriedades de valor classificadas como 'independentes', conforme já visto. Tais propriedades (figura 21, vértices A, B e H), quando gradualmente ajustadas, podem fazer com que uma instância de *MDref* represente todas as possíveis

variações dentro de uma dada solução. Lembrando que a solução em si é caracterizada pelos valores atribuídos às variáveis 'somente-leitura'. Assim, será função do estudo de viabilidade TSx calcular as propriedades independentes de $MDref$, em função dos seus parâmetros de entrada. Esta inversão de cálculos é feita através de blocos de equação especializados para cada estudo de viabilidade.

A figura 27 mostra, em duas imagens, um exemplo de equacionamento para um estudo de viabilidade TSx de latência de rede, referente a um sistema de segurança composto por câmeras. Na primeira imagem, tem-se o diagrama de blocos de TSx montado de forma semelhante aos exemplos da seção anterior, porem aqui exhibe-se também os blocos de equação dentro do BDD de TSx . A segunda imagem mostra o equacionamento propriamente dito, dentro de um diagrama paramétrico do bloco principal de TSx , aqui chamado de *Network Latency*. O diagrama paramétrico mostra como os blocos de equação estão relacionados, e como o único parâmetro de saída, *analysis result*, é calculado.

3.2.2.5 Instanciação e execução

A partir desta etapa, o estudo de viabilidade TSx , para o projeto Px em questão, pode ser dado como concluído e está em condições de ser executado.

A execução de um estudo de viabilidade TSx tem início com a criação de instâncias dos blocos que o compõem. Uma instância de um bloco em SysML vem a ser a sua real utilização, ou seja, enquanto o bloco por si só consiste apenas numa definição, uma instância do mesmo representa uma manifestação física sua, para a qual se podem atribuir valores, bem como realizar cálculos e operações. Um mesmo bloco pode ter várias instâncias.

Conforme já visto, um estudo de viabilidade TSx é composto por três blocos:

1. Bloco de requisitos do projeto Px em questão;

2. Bloco do modelo de referência *MDref*;
3. Bloco principal do próprio estudo de viabilidade *TSx*;

Assim, deverão ser criadas instâncias para cada um dos três blocos citados.

Primeiramente, e como já mostrado na figura 25, deve ser criada uma instância para o bloco de requisitos. A instância deverá receber, em cada uma de suas propriedades de valor, os valores desejados pelo cliente para cada um dos requisitos do projeto *Px* em questão. Caso seja necessário realizar o mesmo estudo de viabilidade *TSx* considerando-se outros valores para os requisitos, pode-se criar uma segunda instância do mesmo bloco, e lhe atribuir os novos valores, preservando-se a primeira.

Em seguida, deve-se instanciar o modelo de referência do sistema de armazenamento logístico, *MDref*. Instanciar *MDref* implica em criar instâncias tanto de seu bloco principal, *Storage*, quanto de suas propriedades de parte e referência, isto é, de seus blocos constituintes. Deverão ser criadas tantas instâncias de *MDref* quantas forem as soluções de armazenamento que se quiser testar para um dado conjunto de requisitos LR de um projeto *Px* em questão. Em cada uma das instância de *MDref*, valores deverão ser atribuídos à todas as propriedades de valor que forem do tipo 'somente-leitura'. Ao atribuir valores específicos a essas propriedades, faz-se com que cada uma das instâncias passe a representar uma solução distinta do sistema de armazenamento logístico, como um AS/RS, ou um sistema de empilhadeiras contra-balanceadas por exemplo. As propriedades do tipo 'independentes' e 'derivadas' não precisam ter valores atribuídos no ato da instanciação, pois o estudo de viabilidade a ser executado se encarregará de fornecer os valores às primeiras, enquanto as segundas são resultados de equações.

Por fim, deve ser criada a instancia do próprio bloco principal do estudo de viabilidade *TSx*. Esta instanciação é intencionalmente deixada por último, pois no ato da mesma terão de ser definidas quais as instâncias de *MDref*, bem como do bloco de

requisitos, que deverão constituí-la.

Feitos os processos de instanciação, o estudo de viabilidade TSx é conduzido através da execução de suas equações para cada uma das instâncias de $MDref$. Seja $Cinst = \{Inst1, Inst2, \dots, Instn\}$ o conjunto de instâncias de $MDref$, ou seja, de soluções distintas de armazenamento logístico, que se deseja avaliar em um determinado conjunto de cenários $Ccen = \{Cen1, Cen2, \dots, Cenn\}$ em um estudo de viabilidade TSx . A primeira instância de $MDref$, $Inst1$, deverá ser atribuída à instância do bloco principal de TSx , e deverão ser executados os cálculos para todos os cenários contidos em $Ccen$, onde a transição de um cenário para o seguinte implicará em incrementos, ou decrementos, dos parâmetros de entrada de TSx . O processo deverá ser repetidos para todas as instâncias de $Cinst$, produzindo assim, o comparativo de soluções de armazenamento, que vem a ser o objetivo do método $MTcsl$, proposto neste trabalho.

A execução dos estudos de viabilidade TSx é operacionalizada dentro da mesma ferramenta *CASE* utilizada para a sua construção, e de $MDref$. Entretanto, deve-se fazer uso também de resolvedores paramétricos para permitir a execução das equações contidas nos diagramas de mesmo nome, bem como softwares auxiliares para fornecer, e armazenar, os valores dos parâmetros de entrada e de saída de TSx , sendo que esses últimos consistem no próprio comparativo. Bancos de dados, arquivos de texto ou softwares de planilhas podem se prestar a esse fim, de acordo com as funcionalidades oferecidas pela ferramenta *CASE* empregada.

A figura 28 exibe um exemplo de instanciação de blocos (28a) para o estudo de viabilidade *CoverageAnalysis*, apresentado na seção 3.2.2.3, imagem 26a. Não apenas o bloco principal de $MDref$, *LittleEyeSystem*, é instanciado como também suas propriedades de parte. E aqui, novamente, o bloco de requisitos foi omitido do diagrama. Apesar das similaridades com blocos em SysML, é possível notar nas instâncias valores reais sendo atribuídos às suas propriedades. Aquelas deixadas sem valor algum pertencem à categoria 'derivadas'. Na mesma figura, porém na imagem 28b, observa-

se a tela de um resolvidor paramétrico configurado para resolver a mesma instância.

Figura 28: Instanciação e execução

A figura 29, ainda considerando o mesmo exemplo de estudo de viabilidade *CoverageAnalysis*, revela como a execução de cenários é operacionalizada com o auxílio de um software de planilhas. O número de cenários é dado pela quantidade de linhas de dados a servir como parâmetros de entrada. No exemplo, o estudo fora montado de modo a usar os parâmetros de entrada 'número de aviões' e 'número de tripulantes', a fim de analisar como o parâmetro de saída, 'milhas percorridas em 24h' é alterado em cada cenário. Ressalta-se que o estudo deste exemplo avalia uma única solução. Um comparativo de soluções poderia ser concebido, por exemplo, trocando-se o modelo da aeronave, ou as características da tripulação, e repetindo-se os cálculos para cada um dos vinte cenários ali exibidos.

(a) Fonte: (LLC, 2014)

(b) Fonte: (LLC, 2014)

Figura 29: Resultado do estudo de viabilidade

4 RESULTADOS

O capítulo de Resultados deste trabalho terá como objetivo demonstrar a validade do método proposto, *MTcsl*, descrito em detalhes na seção anterior 3 - Desenvolvimento, através de sua aplicação em dois estudos de caso distintos. Tais estudos buscarão simular situações reais de projetos de sistemas de armazenamento logístico, nos quais decisões a respeito de qual solução de armazenamento adotar devem ser tomadas em algum momento dos trabalhos. O método proposto *MTcsl*, por produzir comparativos dessas soluções, tem por objetivo diminuir o risco dessas decisões.

O primeiro estudo de caso, seção 4.1, utilizará o método *MTcsl* para produzir um comparativo de soluções de armazenamento em função do número de endereços de armazenagem oferecidos. Assim, a partir do comparativo produzido por *MTcsl*, espera-se que seja possível apontar qual a solução que atende a todos os requisitos LR do projeto *Px* em questão, ao mesmo tempo em que maximiza o número de endereços disponíveis.

O segundo estudo de caso, seção 4.2, terá estrutura semelhante ao primeiro, porém produzirá o comparativo de soluções não em função do número de endereços de armazenagem, mas sim da área total ocupada pela solução. Neste estudo, assim como no anterior, espera-se encontrar uma solução que satisfaça a todos os requisitos LR, mas que minimize a área necessária para sua instalação.

Os estudos apresentados neste capítulo serão conduzidos com o auxílio da ferramenta CASE MagicDrawTMv18.3, junto com o emprego do *plug-in* SysML v18.3,

ambos produzidos pela empresa NoMagic Inc. Serão utilizados também o *plug-in* ParaMagicTMv18.0, da empresa InterCAX, como resolvidor das equações dos diagramas paramétricos do modelo, e também o software de planilhas MS ExcelTM2010, como repositório de dados para os parâmetros de entrada e de saída dos referidos estudos.

4.1 Comparativo de soluções em função do número de posições de armazenamento

O método *MTcsl*, conforme apresentado no objetivo deste trabalho, e detalhado no capítulo 3 - Desenvolvimento, é composto por duas etapas que, no contexto deste estudo de caso, se definem da seguinte forma:

1. Construção de um modelo de referência *MDref* para sistemas de armazenamento logísticos;
2. Construção de um estudo de viabilidade *TSpa* para a produção do comparativo de soluções, utilizando *MDref*, em função das posições de armazenagem disponibilizadas.

4.1.1 Construção do modelo de referência

A construção do modelo de referência *MDref* tem início com as atividades introdutórias de definição de objetivos, tanto do modelo (*OBJmd*) quanto do sistema (*OBJsys*) sendo modelado, bem como o modo que se dará a organização em pacotes do mesmo.

A organização em pacotes para *MDref* pode ser visualizada na figura 30, onde se utilizou a já mencionada estrutura organizacional recursiva (capítulo 3, seção 3.1.1.1), baseada nos quatro pilares da linguagem SysML: requisitos, comportamento, estrutura e parâmetros. Na figura, é possível notar pacotes definidos para cada um dos pilares. Esta estrutura é dita recursiva pois caso seja necessário detalhar os modelos também

(a) Estrutura de pacotes do modelo de referência.

(b) Árvore do modelo de referência.

Figura 30: Organização do modelo de referência do sistema de armazenamento logístico.

das propriedades de parte, i.e., blocos constituintes de *MDref*, a mesma estrutura de pacotes apresentada na figura 30 pode se replicada para cada um dos blocos, que deverão estar localizados dentro do pacote intitulado 'Estrutura' (*Structure*). Adicionalmente, além dos quatro pacotes principais, este trabalho insere pacotes auxiliares na estrutura do modelo, a fim de tornar mais eficaz sua organização. A utilização de tais pacotes fora extraída de boas práticas de modelagem em SysML listadas em (R.KARBAN et al., 2011), (PEARCE; FRIEDENTHAL, 2013), (PLEGT, 2011) e (FRIEDENTHAL; MOORE; STEINER, 2012). Seguem suas definições:

- *Goals* (Objetivos). Pacote que irá armazenar os objetivos tanto do modelo quanto do sistema. Seu conteúdo deverá ser o primeiro a ser estudado pela equipe responsável pelo projeto, para que não se tenha dúvidas do que se trata o sistema, de como será utilizado, e que tipo de informações conterà seu modelo.
- *Analysis* (Estudos de Viabilidade). Pacote que irá armazenar todos os estudos de viabilidade que façam uso do modelo de referência *MDref*. Neste estudo de caso, armazenará os elementos do modelo de *TSpa*. Do ponto de vista técnico, tal pacote não tem a obrigatoriedade de constar dentro da estrutura organizacional de *MDref*, sequer dentro do mesmo arquivo da ferramenta *CASE*, só o sendo para fins de praticidade.
- *Glossary* (Glossário). Pacote que contém as principais definições e termos utilizados na construção do modelo, podendo documentar, por exemplo, nomes de blocos, operações e propriedades.
- *Traceability* (Relatórios). Pacote que irá armazenar todos os relatórios que se julguem necessários para a compreensão de *MDref*. Poderá conter relatórios como, por exemplo, matrizes de rastreabilidade de requisitos, e mapas de relacionamentos que expressem análises causais.
- *ValueTypes* (Tipos de valores). Pacote que irá armazenar as unidades utilizadas

para tipificar as propriedades de valor. Neste trabalho, consiste nas unidades do Sistema Internacional, SI.

Uma vez estabelecida a estrutura organizacional em pacotes para o modelo *MDref*, pode-se concluir suas atividades introdutórias definindo-se seu objetivo, *OBJmd*, bem como o objetivo do sistema como um todo, *OBJsys*. Ambos são apresentados na figura 31, e são inseridos dentro do pacote *Goals* (Objetivos).

(a) Objetivo do sistema de armazenamento.

(b) Objetivo do modelo de referência .

Figura 31: Definição dos objetivos do sistema e do modelo de referência.

Feitas as atividades introdutórias, pode-se iniciar a elaboração de *MDref* seguindo-se as etapas subsequentes, apresentadas no capítulo 3, seção 3.1.2:

1. Escopo do modelo (*SCPsys*);
2. *Template* de requisitos (TR);

3. Especificação caixa-preta;
4. Análises funcional e causal;
5. Decomposição estrutural;
6. Classificação de parâmetros e relações.

O escopo do modelo, *SCPsys*, que tem por função definir a fronteira que separa o sistema de armazenamento logístico do ambiente ao seu redor, pode ser visualizado na figura 32, onde se tem a primeira ocorrência do bloco principal do sistema de armazenamento logístico, estereotipado como «system». Neste trabalho, o bloco principal do sistema de armazenamento recebe o nome *Storage*, enquanto o bloco de seu contexto define-se como *Warehouse*. O bloco *Storage* é o componente inicial de *MDref*, marcando o início de sua construção propriamente dita.

Figura 32: Escopo do sistema de armazenamento logístico.

O *template* de requisitos TR, que vem a ser uma lista dos requisitos mais frequentes em projetos de sistemas de armazenamento logístico, pode ser traduzido como as principais preocupações que clientes manifestam ao adquirirem tais sistemas. Neste trabalho, a montagem de TR, conforme já mencionado, seguiu as etapas da metodologia exposta no capítulo 1, seção 1.3, e seu resultado pode ser visto na figura 33.

Criteria			
Scope (optional):			Drag elements from the Model Browser { } ... Filter: Q
#	Id	Name	Text
1	36	Armazenamento e retirada	O sistema de armazenamento logístico deverá dispor de mecanismos próprios para realizar as operações de armazenagem, e retirada, de mercadorias.
2	31	Capacidade de armazenamento	O sistema de armazenamento logístico deverá prover quantidade adequada de endereços de armazenamento ao cliente.
3	32	Custos	O sistema de armazenamento logístico deverá respeitar as restrições de custos feitas por parte do cliente.
4	33	Dimensões	O sistema de armazenamento logístico deverá ter uma pegada dimensional compatível com o local em que será instalado.
5	34	Giro de mercadorias	O sistema de armazenamento logístico deverá prover agilidade tanto no armazenamento quanto na retirada de mercadorias.
6	35	Peso	O sistema de armazenamento logístico deverá respeitar as restrições de carga imposta ao piso do local em que será instalado.
7	38	Tipo de UoH utilizado	O sistema de armazenamento logístico deverá trabalhar com UoHs padrão de mercado.
8	39	Turnos	O sistema de armazenamento logístico deverá operar de acordo com os horários de trabalho do local em que será instalado.

Figura 33: *Template* TR de requisitos para o sistema de armazenamento logístico.

A partir de TR, derivam-se os quatro conjuntos especializados de requisitos, que, na etapa seguinte, irão definir a especificação caixa-preta do bloco principal de *MDref*, *Storage*. Os quatro conjuntos, já apresentados no capítulo 3, são os seguintes:

1. Requisitos não-funcionais (*TRnf*);
2. Requisitos funcionais (*TRf*);
3. Requisitos de interface (*TRif*);
4. Premissas (*TPR*).

As figuras de 34 a 37 mostram imagens da listagem de cada um dos quatro conjuntos para o modelo de referência *MDref*, alguns em formato tabular, outros em diagrama, sendo ambos equivalentes. Já as figuras seguintes, 38a e 38b, exibem relatórios em formato matricial, mostrando relações de derivação existentes para os requisitos de TR, e para o conjunto de premissas *TPR*, respectivamente. Todo e qualquer requisito

pertencente a {TRnf, TRf, TRif} deve ser derivado de, ao menos, um requisito de TR, enquanto premissas listadas em *TPR* podem ser derivados tanto do *template* quanto de seus outros três conjuntos. Tanto TR quanto {TRnf, TRf, TRif, TPR}, devem ser colocados dentro do pacote intitulado 'Requisitos' (figura 30).

Criteria			
Scope (optional):		Drag elements from the Model Browser	... Filter: <input type="text"/>
#	Id	Name	Text
1	6	Performance do sistema de armazenamento	O sistema de armazenamento logístico deverá atender a todos os requisitos de performance
2	6.10	Nº de movimentações por unidade de tempo	O sistema de armazenamento deverá poder realizar um nº de movimentos de entrada e de saída de UoHs, por intervalo de tempo, igual ou superior ao especificado pelo cliente.
3	6.11	Nº de endereços disponíveis	O sistema de armazenamento deverá disponibilizar um total de endereços igual, ou superior, ao especificado pelo cliente.
4	6.14	Custos operacionais	Os custos operacionais deverão ser iguais ou inferiores aos especificados pelo cliente.
5	6.15	Custos de instalação	Os custos de instalação deverão ser iguais ou inferiores aos especificados pelo cliente.
6	6.16	Área	A área utilizada pelo sistema de armazenamento deverá ser igual ou inferior à especificada pelo cliente.
7	6.17	Altura, largura e comprimento	As dimensões de comprimento, largura e altura do sistema de armazenamento deverão ser iguais ou inferiores as especificadas pelo cliente.
8	6.18	Carga sobre piso	A força por m ² exercida sobre o piso, pelo sistema de armazenamento, deverá ser inferior à especificada pelo cliente.
9	20.13	Especificações de UoH	A UoH com a qual o sistema de armazenamento deverá trabalhar será do tipo Pallet PBR

Figura 34: Requisitos não-funcionais para o sistema de armazenamento logístico, exibidos em formato tabular.

Figura 35: Requisitos funcionais para o sistema de armazenamento logístico, exibidos em diagrama.

Figura 36: Requisitos de interface para o sistema de armazenamento logístico, exibidos em diagrama.

Criteria			
Scope (optional):		4 - Model Assumptions	Filter: <input type="text"/>
#	Id	Name	Text
1	20	☐ Premissas	A construção do modelo de referência para o sistema de armazenamento logístico partirá dos seguintes pressupostos
2	20.7	☐ Fluxo de UoHs	O sistema de armazenamento deverá ser instalado num Centro de Distribuição, ou estrutura equivalente, onde a entrada e a saída de pallets ocorre no mesmo lado, com corredores paralelos ao fluxo dos mesmos.
3	20.8	☐ Pontos de entrada e de saída de UoHs	Os transportadores buscarão e deixarão as UoHs, após as operações de armazenagem e retirada, no início dos corredores. A partir dali, a UoH não está mais dentro das fronteiras do sistema de armazenamento
4	20.9	☐ Altura máxima de empilhamento de UoHs	O nível de empilhamento das estruturas de racks, onde são armazenadas as UoHs, deverá ser equivalente a máxima altura que o transportador puder atingir.
5	20.10	☐ Ciclo Duplo	As operações de armazenagem e retirada serão realizadas sempre em ciclo duplo
6	20.11	☐ Non-ABC	Não será considerada a implementação da curva ABC
7	20.12	☐ Turnos	Será presumido turno único, com 24h de operação, e 100% de eficiência
8	20.13	☐ Especificações de UoH	A UoH com a qual o sistema de armazenamento deverá trabalhar será do tipo Pallet PBR

Figura 37: Premissas e restrições para o sistema de armazenamento logístico, exibidos em formato tabular.

Dando continuidade à montagem do modelo de referência *MDref*, para sistemas de armazenamento logístico, a etapa subsequente vem a ser a especificação de sua caixa-preta a partir dos conjuntos de requisitos derivados *TRnf*, *TRf* e *TRif*. Conforme visto no capítulo 3, cada um desses conjuntos dará origem a propriedades de valor, operações e portas, respectivamente, no bloco principal *Storage*. Já as premissas listadas em *TPR* (figura 37) irão balizar decisões de design, como formulações de equações, ou valores a se atribuir às instâncias de *MDref*.

A figura 39 exibe a especificação caixa-preta já concluída para o bloco *Storage*, que fora mostrado pela primeira vez no diagrama de escopo da figura 32 apenas como um bloco opaco. Na matriz de relações da figura subsequente, 40, são exibidas todas as relações que refinaram os requisitos contidos em *TRnf*, *TRf* e *TRif* na referida especificação caixa-preta.

Legend		0 - Template Requirements									
DeriveReq		31 Capacidade de armazenamento	32 Custos	33 Dimensões	34 Giro de mercadorias	35 Peso	36 Armazenamento e retirada	38 Tipo de UoH utilizado	39 Turnos		
1 - Non Functional Requirements		1	2	2	1	1					
6 Performance do sistema de armazenamento											
6.10 Nº de movimentações por unidade de tempo		1									
6.11 Nº de endereços disponíveis		1									
6.14 Custos operacionais		1									
6.15 Custos de instalação		1									
6.16 Área		1									
6.17 Altura, largura e comprimento		1									
6.18 Carga sobre piso		1									
2 - Functional Requirements							2				
18 Operações do sistema de armazenamento logístico											
18.3 Armazenar		1									
18.4 Retirar		1									
3 - Interface Requirements							2				
19 Interfaces do sistema de armazenamento logístico											
19.3 Entrada / saída de UoHs		1									
19.4 Entrada / saída informações sobre pedidos		1									

(a) Requisitos derivados do *template* TR.

Legend		4 - Model Assumptions									
DeriveReq		20.7 Fluxo de UoHs	20.8 Pontos de entrada e de ...	20.9 Altura máxima de empilh...	20.10 Ciclo Duplo	20.11 Non-ABC	20.12 Turnos	20.13 Especificações de UoH			
0 - Template Requirements							1	1			
31 Capacidade de armazenamento											
32 Custos											
33 Dimensões											
34 Giro de mercadorias											
35 Peso											
36 Armazenamento e retirada											
38 Tipo de UoH utilizado		1									
39 Turnos		1									
1 - Non Functional Requirements			1	1	1	1	1				
6 Performance do sistema de armazenamento											
6.10 Nº de movimentações por unidade de tempo		4									
6.11 Nº de endereços disponíveis											
6.14 Custos operacionais											
6.15 Custos de instalação											
6.16 Área											
6.17 Altura, largura e comprimento		1									
6.18 Carga sobre piso											
2 - Functional Requirements											
3 - Interface Requirements											

(b) Relações de derivação entre premissas e demais requisitos.

Figura 38: Relatórios com relações de derivação entre requisitos.

Figura 39: Especificação caixa-preta para o sistema de armazenamento logístico.

Figura 40: Relatório com relações de refinamento entre requisitos e especificação caixa-preta.

A etapa construtiva seguinte à especificação caixa-preta de *MDref* implica na definição de como suas propriedades de valor, estereotipadas como MOEs, são calculadas. Determinar as linhas de cálculo de cada uma de suas MOEs requer a execução das

análises funcional e causal de *MDref*, seguida de sua decomposição estrutural, a fim de identificar seus blocos constituintes (propriedades de parte e referência, no jargão da linguagem SysML) e respectivas propriedades.

Figura 41: Análise funcional para a operação de armazenagem.

As figuras 41 e 42 exibem análises funcionais para ambas as operações listadas na especificação caixa-preta do bloco *Storage*, *store* (armazenar) e *retrieve* (retirar). Em tais análises, performadas através de diagramas de atividades (ACT), tem-se a decomposição das operações mencionadas em sequências de atividades atomizadas, junto com a identificação dos sujeitos, e dos objetos, envolvidos nas mesmas.

Figura 42: Análise funcional para a operação de retirada.

Os sujeitos e objetos, identificados na etapa de análise funcional das operações do bloco *Storage*, darão origem à lista de blocos candidatos a fazer parte de sua estrutura. Tal listagem deverá ser complementada com a execução da etapa seguinte, a análise causal. Nas análises funcionais realizadas neste estudo de caso, foram identificados os seguintes blocos candidatos a constituintes de *Storage*:

- *Shuttle* (Transportador);
- *RackStructure* (Estrutura de Prateleiras);

- *UnitOfHandling* (Unidade de manuseio);
- *OrderInfo* (Pedido).

As análises causais, além de complementarem a lista de blocos candidatos à composição estrutural do bloco *Storage*, identificam as propriedades de valor de tais blocos, bem como as equações que as relacionam com as MOEs do sistema. Isto implica que para cada MOE listada no bloco *Storage*, uma respectiva análise causal deverá ser realizada, sempre respeitando, naturalmente, os critérios de parada de modelagem conforme explicado no capítulo 3, seção 3.1.2.5.

No transcorrer das análises causais para as MOEs do sistema, novas propriedades e equações que forem sendo identificadas deverão ser inseridas nos seus blocos correspondentes, sendo as primeiras como propriedades de valor, e as segundas como blocos de equação (*constraint blocks*). Caso os blocos a quais pertençam ainda não façam parte da listagem de blocos candidatos, os mesmos deverão ser criados, e inseridos, dentro da pasta 'Structure' do modelo. As novas propriedades de valor e equações deverão estar relacionadas entre si, e com as MOEs que delas dependem, dentro de diagramas paramétricos pertencentes aos mesmos blocos que as possuem. Para fins de organização, deve-se colocar todos os blocos de equação dentro da pasta '3- Parametrics', sendo que deverão se manifestar dentro de seus blocos como instâncias dessas equações.

As figuras 43 e 44 exibem exemplos de análises causais realizadas para as MOEs 'Area' (Área), 'OperationalCost' e 'InsalationCost' (Custos de instalação e operacionais). Os exemplos são exibidos em diagramas paramétricos, onde os blocos estruturais são mostrados como retângulos de cantos retos, e os blocos de equação exibem cantos arredondados. Enquanto a análise para a MOE de área confirma a listagem já existente de blocos estruturais, nos cálculos das MOEs de custos nota-se a ocorrência de um novo bloco, 'IndustrialServices' (Serviços Industriais), que não fora previamente

(a) Análise causal expressa em Diagrama Espinha-de-peixe

(b) Análise causal expressa em Diagrama Paramétrico

Figura 43: Análise causal para a MOE Área.

identificado nas análises funcionais, e que passa agora constar na listagem de blocos candidatos a composição estrutural. Nos mesmos cálculos, também é possível reparar que a MOE de custo operacional dependem diretamente da MOE 'Área'.

Repetindo-se a estrutura dos exemplos anteriores para todas as MOEs do bloco

Figura 44: Análise causal para as MOEs de custos.

Storage, determina-se como cada uma delas é calculada, junto com a listagem completa de blocos candidatos à composição estrutural. Tal lista, para este estudo de caso, pode ser vista na figura 45 contendo um total de seis blocos. Desse total, quatro foram identificados nas análises funcionais, e os outros dois, 'Floor' (Piso) e 'IndustrialServices' (Serviços Industriais), nas análises causais.

Criteria	
Element Type: Block	
#	Name
1	Floor
2	IndustrialServices
3	OrderInfo
4	RackStructure
5	Shuttle
6	UnitOfHandling

Figura 45: Listagem de blocos para composição estrutural.

Para a conclusão da montagem de *MDref* a partir do bloco *Storage*, ficam por

fazem as classificações de suas relações entre blocos, e de suas propriedades de valor, conforme descrito no capítulo 3, seção 3.1.2.7.

As relações entre blocos constituintes pode ser, como já visto, de simples associação ou composição, sendo a primeira utilizada quando os ciclos de vida dos blocos são independentes entre si mas necessitam uns dos outros para os cálculos de suas equações, e a segunda quando não são independentes, caso típico de arranjos do tipo 'todo->parte'.

Já propriedades de valor, tanto as pertencentes ao bloco *Storage* quanto às suas propriedades de parte e referência, devem receber a classificação 'independentes', 'somente-leitura' ou 'derivadas'. Conforme explicado, são classificadas como do primeiro tipo ('independentes') aquelas propriedades cujo valor pode variar livremente durante o processo de montagem dos comparativos. As propriedades do segundo tipo ('somente-leitura') têm seus valores atribuídos apenas no momento da instanciação, e estão relacionadas ao uso de componentes de prateleira (COTs), ou decorrem da incidência de normas técnicas e de segurança. Por fim, propriedades do terceiro tipo ('derivadas'), são resultados de equações, e não podem ter valores diretamente atribuídos.

A figura 46 exhibe o modelo de referência para sistemas de armazenamento logístico, *MDref*, finalizado. Nele já foram feitas as classificações das relações entre seus blocos constituintes, junto com as de suas propriedades de valor. Os blocos em relações de associação são mostrados como blocos opacos apenas para não sobrecarregar a visualização do diagrama BDD.

Primeiro ponto de interesse a respeito de *MDref*, na figura 46, está no fato de que, neste modelo, há apenas três propriedades da categoria 'independentes', e todas pertencentes à propriedade de parte *RackStructure*:

- *AisleLengthInUoH*. Comprimento dos corredores, em termos da contagem de

Figura 46: O modelo de referência para sistemas de armazenamento logístico, *MDref*.

UoHs;

- *AisleCount*. Número de corredores.
- *ShuttlePerAisle*. Quantidade de transportadores utilizados por corredor.

Estas serão as propriedades que sofrerão variações, de forma indireta através dos estudos de viabilidade, para a produção dos comparativos de soluções.

Segunda observação referente a *MDref* está na presença de relações de associação não apenas entre seu bloco principal, *Storage*, e suas propriedades de referência, mas também entre seus blocos constituintes. Pelo diagrama, percebe-se que o bloco *RackStructure* utiliza como parâmetros de entrada propriedades tanto do bloco *Shuttle* quanto do bloco *UnitOfHandling*.

Assim, encerra-se a aplicação da primeira etapa do método *MTcsl*, que consistiu na construção do modelo de referência para sistemas de armazenamento logístico, *MDref*. Tal modelo, podendo também ser chamado de arquitetura de referência, irá servir como artefato para a execução da segunda etapa do método, que consistirá na montagem do estudo de viabilidade *TSpa*, visando o comparativo de soluções em função do número de endereços de armazenamento oferecidos.

4.1.2 Construção do primeiro estudo de viabilidade

Conforme apresentado no início desta seção, este estudo de viabilidade, *TSpa*, irá utilizar o modelo de referência *MDref* para a produção de um comparativo de soluções em função do número de endereços de armazenamento oferecido por cada uma. O referido estudo busca simular uma situação real de projeto de sistemas logísticos de armazenamento. Tal projeto será denominado *Ppa* durante a descrição deste estudo.

Assim, deve-se iniciar a construção de *TSpa* supondo um primeiro contato com o cliente, do qual se obtém um documento contendo a lista de requisitos *LRpa* que o sistema de armazenamento logístico deverá cumprir. Tal documento, supõem-se, é apresentado de forma não estruturada, devendo os subconjuntos de *LRpa*, {*LRnf*, *LRf*, *LRif*, *LPR*}, ser derivados diretamente de seu conteúdo.

Seja o seguinte bloco de texto extraído de uma ata de reunião com o cliente:

- "Fez-se a compra de um terreno de dimensões 50m x 150m, no qual se deseja instalar um condomínio logístico a operar com pallets no padrão PBR. A fim de tornar tal condomínio competitivo, a capacidade mínima oferecida de giro diário de unidades deverá ser de 4000 pallets/dia, sendo 2000 entrando e 2000 saindo. Os custos mensais não devem exceder os R\$800.000,00 a fim de não encarecer demais a operação. Dispõe-se de um teto de R\$15.000.000,00 para a instalação dos sistemas de armazenamento (equipamentos de movimentação,

estruturas metálicas), e se deseja saber qual das soluções disponíveis comercialmente ofereceria o maior número possível de posições de armazenamento".

Inicialmente deve-se, como boa prática, formalizar o objetivo de *Tpa*, e o que se pretende analisar através do mesmo. Em seguida, a partir do texto apresentado pelo cliente, e que representa *LRpa* para o projeto *Ppa* em questão, é possível derivar requisitos não-funcionais (LRnf) e premissas (LPR), os quais podem ser visualizados nas figuras 47 e 48, respectivamente.

Figura 47: Requisitos não funcionais para o estudo *TSpa*.

Assim como *MDref*, os estudos de viabilidade, *Tpa* inclusive, também são formados por diagramas SysML, e devem ser armazenados na estrutura de pacotes previamente estabelecida. Neste trabalho, os estudos de viabilidade (seus elementos e

Figura 48: Premissas para o estudo *TSpa*. Figura 49: Objetivo do estudo *TSpa*.

diagramas) serão armazenados em seus sub-pacotes correspondentes, dentro do pacote intitulado 'Analysis' (figura 50).

Figura 50: Estudo *TSpa* inserido no diagrama de pacotes.

Figura 51: Mapeamento de requisitos do projeto *Ppa* (linhas) para o *template* TR (colunas)

A figura seguinte, 51, mostra como foi realizado o mapeamento dos conjuntos de *LRpa* para os respectivos conjuntos do *template* TR. Tal mapeamento irá indicar a necessidade de se estender, ou não, *MDref* para permitir que o mesmo possa ser usado na realização do estudo *TSpa*. Conforme explicado no capítulo 3, subseção 3.2.1.1, para

que *MDref* não tenha de ser modificado, i.e., estendido, todos os requisitos derivados de *LRpa* devem ser mapeáveis nos conjuntos do *template* TR. Tal mapeamento é operacionalizado na linguagem SysML através de relacionamentos do tipo «trace» entre os requisitos em questão.

Dado que todos os requisitos derivados de *LRpa* puderam ser mapeados de modo satisfatório, o modelo de referência *MDref* não terá de ser estendido para que possa ser usado no estudo *TSpa*. Como de *LRpa* não se derivaram nenhum requisito dos conjuntos *LRf* ou *LRif*, será presumida a não-alteração das operações e portas de *MDref*.

	41	42	43	44	45	46
1 - Non Functional Requirements	1					
6 Performance do sistema	1					
6.10 Nº de movimentações por unidade	1					
6.11 Nº de endereços disponíveis		1				
6.14 Custos operacionais					1	
6.15 Custos de instalação						1
6.16 Área		1				
6.17 Altura, largura e comprimento			1			
6.18 Carga sobre piso				1		

Figura 52: Requisitos não-funcionais do *template* candidatos a parâmetros de entrada.

Conforme a listagem de situações também exposta no capítulo 3, subseção 3.2.1.1, este estudo de viabilidade *Tpa* recai primeira na situação, onde os parâmetros de entrada possíveis para *Tpa* consistem em todos os requisitos do subconjunto *TRnf* que não foram mapeados por *LRnf*. Tais parâmetros podem ser visualizados na figura 52, a qual vem a ser um detalhamento da matriz exibida na figura anterior, 51, porém apenas para os requisitos da categoria não-funcionais e exibindo todos os elementos de requisitos, inclusive os que não tem relações. Tem-se, assim, que o estudo *TSpa* poderia

ter como parâmetros de entrada tanto as MOEs 'Número de endereços', quanto 'Carga sobre o piso', não tivesse o cliente já explicitado sua vontade de que o estudo fosse realizado especificamente em função da primeira.

Feita a definição do parâmetro de entrada ('Número de endereços') e dos parâmetros de saída (todas as MOEs que foram mapeadas pelos requisitos não-funcionais de *Ppa*), o estudo *TSpa* já pode ter seu diagrama de blocos montado e equações formalizadas. Seguindo a sequência de etapas expostas na subseção 3.2, deve-se transferir os valores dos requisitos para um bloco criado especialmente para armazená-los, uma vez que na linguagem SysML os elementos de requisitos armazenam apenas valores textuais. Tal bloco, visualizado na figura 53, deve ser colocado junto com os demais blocos do estudo de viabilidade no pacote intitulado '*Structure*' (Estrutura).

Figura 53: Requisitos apreendidos em bloco para o estudo *TSpa*.

O diagrama de blocos de *TSpa*, de acordo com as etapas da subseção 3.2, deverá conter além do bloco de requisitos, o bloco do modelo de referência *MDref*, e um bloco para representar o próprio estudo sendo conduzido. Tal arranjo pode ser conferido na figura 54a. Na mesma imagem, também são mostradas as equações («constraints») utilizadas exclusivamente para o estudo *TSpa* em questão. Tais equações ali se encontram para cumprir duas finalidades:

1. Calcular as margens de segurança dos requisitos;
2. Expressar os parâmetros de saída em função dos parâmetros de entrada, através da inversão da ordem de cálculo das MOEs de *MDref*.

(a) Diagrama de blocos do estudo de viabilidade *TSpa* (b) Bloco principal do estudo de viabilidade *TSpa*, com detalhes para os parâmetros de entrada e de saída.Figura 54: Os blocos constituintes do estudo de viabilidade *TSpa*.

As primeiras, que recebem o sufixo MOS¹ em seus nomes, irão computar em termos percentuais se um requisito está, ou não, sendo atendido. As segundas deverão receber os parâmetros de entrada e convertê-los em termos dos parâmetros independentes de *MDref*. A figura 55b mostra o cálculo da propriedade de valor independente

¹MOS: *margin of safety* (margem de segurança).

'*AisleLengthInUoH*', sendo feito em função do parâmetro de entrada '*PalletPosition*'. Às demais propriedades independentes, '*AisleCount*' e '*ShuttlePerAisle*', serão presumidos valores como se fossem parâmetros de entrada de *TSpa*, porém auxiliares. Tal manobra se faz necessária sempre que não se puder calcular todos as propriedades independentes de *MDref* apenas a partir dos parâmetros de entrada originais de *TSpa*, em casos onde se tem mais incógnitas do que equações.

Com o equacionamento concluído, o estudo de viabilidade *TSpa* está pronto para ser instanciado e executado. Conforme mencionado nas etapas de elaboração dos estudos de viabilidade, no capítulo anterior, instanciar *TSpa* implicará na criação de uma instância para seu bloco principal, *PalletPosition Analysis*, em uma instância para seu bloco de requisitos, *ppAnalysisReqs*, e tantas instâncias de *Storage*, bloco principal de *MDref*, quantas forem as soluções de armazenamento que se quiser comparar para o projeto *Ppa* em questão.

Neste estudo de caso, produziu-se o comparativo utilizando três instâncias de *MDref*, simbolizando três soluções distintas de armazenamento logístico. Em ordem crescente de complexidade:

1. Solução com empilhadeiras contrabalanceadas;
2. Solução com empilhadeiras trilaterais para corredores estreitos;
3. Solução totalmente automatizada, com uso de trans-elevadores (AS/RS).

A figura 56 mostra um exemplo de instanciação de *MDref* para uma solução do tipo AS/RS, seguido do seu uso na instanciação completa do estudo *TSpa*, figura 57. As propriedades de valor que apontam para aspas simples são do tipo 'derivadas', e não possuem valor algum atribuído pois, no ato da instanciação, o estudo *TSpa* ainda não pôde ser executado, de modo que nenhum valor foi ainda produzido.

(a) Equacionamento das margens de segurança (MOS) do estudo de viabilidade *TSpa*.

(b) Equacionamentos de conversão de parâmetros de entrada → propriedades independentes de *MDref*.

Figura 55: Equacionamento do estudo de viabilidade *TSpa*.

As instâncias, assim como demais elementos, também devem ser organizadas dentro da estrutura de pacotes do modelo, sendo usualmente armazenadas na sub-pasta '*Physical*', dentro da pasta '*Structure*'. A figura 58 mostra a estrutura de pacotes tanto

Figura 56: Instanciação de *MDref* para a solução AS/RS

de *MDref*, pasta *Storage System Reference Model*, quanto dos estudos de viabilidade, pasta *Analysis*. Nela, recebem destaque uma das instâncias de *MDref*, representando a solução AS/RS, junto com as instâncias dos blocos de requisitos e do bloco principal de *TSpa*.

A execução das instâncias do modelo de referência *MDref* e do estudo de viabili-

Figura 57: Instanciação do estudo de viabilidade *TSpa*

dade *TSpa* fica a cargo dos resolvedores paramétricos, acessórios normalmente disponibilizados em conjunto com as ferramentas de software *CASE*. Apesar de variarem em termos de interface e funcionalidades, tais resolvedores exercem basicamente a função de interpretar as equações inseridas nos diagramas paramétricos do modelo SysML, resolvê-las para um dado conjunto de valores de entrada, e exportar os resultados produzidos para softwares auxiliares, como bancos de dados, arquivos-texto ou planilhas.

As figuras 59 e 60 mostram cópias de telas do resolvedor paramétrico ParaMagic imediatamente antes, e após, a execução das equações do estudo de viabilidade *TSpa*, para a instância de *MDref* representando a solução AS/RS. A coluna de causalidades (*Causality*) indica, através da atribuição 'dado' e 'calculado' (*given* e *target*, respectivamente), quais os parâmetros de entrada, e quais os de saída.

Figura 58: Organização das instâncias de *TSpa* na estrutura de pacotes.

Name	Values	Type	Causality	Qualified Name
PalletPosition Analysis		PalletPosition Analysis		1- Storage - Reference Ar...
AisleCount	2	Real	given	
Area	?????	m ²	target	
Area_Mos	?????	Real	target	
Area_Req	?????	m ²	target	
InstallCost	?????	R\$	target	
InstallCost_Mos	?????	Real	target	
InstallCost_Req	?????	R\$	target	
Lenght	?????	m	target	
Length_Mos	?????	Real	target	
Length_Req	?????	m	target	
OpCost	?????	R\$/month	target	
OpCost_Mos	?????	R\$/month	target	
OpCost_Req	?????	R\$/month	target	
PalletPosition	2.000	Real	given	
ShuttlePerAisle	1	Real	given	
Throughput_Mos	?????	Real	target	
Throughput_Req	?????	UoH/day	target	
Throuhput	?????	UoH/day	target	
Width	?????	m	target	
Width_Mos	?????	Real	target	
Width_Req	?????	m	target	
ppAnalysisReqs		ppAnalysisReqs		1- Storage - Reference Ar...
Area	7.500	m ²	given	
InstallCost	15.000.000	R\$	given	
Lenght	150	m	given	
OpCost	800.000	R\$/month	given	
Throughput	4.000	UoH/day	given	
Widht	50	m	given	
storage		Storage		1- Storage - Reference Ar...

Figura 59: Resolvedor paramétrico configurado para as instâncias do estudo *TSpa*, ainda sem valores.

Entretanto, as telas dos resolvedores paramétricos podem exibir apenas um cenário por vez, enquanto o comparativo de soluções requer a visualização dos resultados de diversos cenários, para diversas soluções, simultaneamente. Assim, é necessária a configuração de repositórios de dados auxiliares para permitir a execução, em lote, do estudo de viabilidade por parte dos resolvedores paramétricos. O software de planilhas MS ExcelTM se prestará a este fim. Nele, foram inseridos valores para o parâmetro de entrada 'Número de endereços' num intervalo variando de dois a trinta mil endereços, totalizando vinte e nove cenários.

Uma vez executados todos os cenários programados no resolvedor paramétrico, o

Name	Values	Type	Causality	Qualified Name
PalletPosition Analysis		PalletPosition Analysis		1- Storage - Reference Ar...
AisleCount	2	Real	given	
Area	245,7m ²		target	
Area_Mos	0,967Real		target	
Area_Req	7.500m ²		target	
InstallCost	3.408.000R\$		target	
InstallCost_Mos	0,773Real		target	
InstallCost_Req	15.000.000R\$		target	
Lenght	18,9m		target	
Length_Mos	0,874Real		target	
Length_Req	150m		target	
OpCost	118.650,6R\$/month		target	
OpCost_Mos	0,852R\$/month		target	
OpCost_Req	800.000R\$/month		target	
PalletPosition	2.000	Real	given	
ShuttlePerAisle	1	Real	given	
Throughput_Mos	-0,3Real		target	
Throughput_Req	4.000UoH/day		target	
Throuhput	2.801,92UoH/day		target	
Width	13m		target	
Width_Mos	0,74Real		target	
Width_Req	50m		target	
ppAnalysisReqs		ppAnalysisReqs		1- Storage - Reference Ar...
Area	7.500m ²		given	
InstallCost	15.000.000R\$		given	
Lenght	150m		given	
OpCost	800.000R\$/month		given	
Throuhput	4.000UoH/day		given	
Widht	50m		given	
storage		Storage		1- Storage - Reference Ar...

Figura 60: Resolvedor paramétrico configurado para as instâncias do estudo *TSpa*, já com valores calculados.

comparativo entre soluções pode ser visto, concluído, nos repositórios de dados previamente configurados. Através da análise dos dados produzidos no estudo de viabilidade, e exportado para as planilhas, pode-se apontar qual a solução que melhor atende aos requisitos do projeto *Ppa*. As figuras 63, 64 e 65 exibem os resultados das execuções de vinte e nove cenários de *TSpa*, para o projeto *Ppa*, para cada uma das três soluções citadas no início da desta seção. Observa-se que apenas a instância de solução AS/RS consegue atender a todos os requisitos, para uma faixa de oito a quinze mil endereços oferecidos.

Figura 61: Resolvedor paramétrico configurado para a execução em lote de múltiplos cenários, com conexão a repositório de dados externo em planilhas.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1			ASRS																	
2	Endereços	Corredores	Area (m²)			Throughput (UoH/Day)			Comprimento (m)			Largura (m)			Custo Instalação (R\$)			Custo Operacional (R\$)		
3			Real	Req. (max)	% safety	Real	Req.(min)	% safety	Real	Req. (max)	% safety	Real	Req. (max)	% safety	Real	Req. (max)	% safety	Real	Req. (max)	% safety
4	2000	2																		
5	3000	2																		
6	4000	2																		
7	5000	2																		
8	6000	2																		
9	7000	2																		
10	8000	4																		
11	9000	4																		
12	10000	4																		
13	11000	4																		
14	12000	4																		
15	13000	4																		
16	14000	4																		
17	15000	4																		
18	16000	8																		
19	17000	8																		
20	18000	8																		
21	19000	8																		
22	20000	8																		
23	21000	8																		
24	22000	8																		
25	23000	8																		
26	24000	8																		
27	25000	8																		
28	26000	8																		
29	27000	8																		
30	28000	8																		
31	29000	8																		
32	30000	8																		
33																				
34																				
35																				
36																				
37																				
38																				

Figura 62: Repositório de dados externo, em planilha, antes da execução dos cenários do estudo de viabilidade.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
Endereços	Corredores	Área (m²)		Throughput (UoH/Day)		Comprimento (m)		Largura (m)		Custo Instalação (R\$)		Custo Operacional (R\$)		% safety	Req. (max)	% safety	Req. (max)	Real	Req. (max)	% safety
		Real	Req. (max)	Real	Req. (min)	% safety	Real	Req. (max)	Real	Req. (max)	Real	Req. (max)	Real							
1																				
2	4 2000	2	246	7500	97%	2802	4000	-30%	18,9	150	87%	13	50	74%	R\$ 3.408.000,00	R\$ 15.000.000,00	74%	R\$ 118.650,60	R\$ 800.000,00	85%
3	5 3000	2	369	7500	95%	2736	4000	-32%	28,35	150	81%	13	50	74%	R\$ 3.912.000,00	R\$ 15.000.000,00	74%	R\$ 150.975,90	R\$ 800.000,00	81%
4	6 4000	2	491	7500	93%	2664	4000	-33%	37,8	150	75%	13	50	74%	R\$ 4.416.000,00	R\$ 15.000.000,00	71%	R\$ 183.301,20	R\$ 800.000,00	77%
5	7 5000	2	614	7500	92%	2596	4000	-35%	47,25	150	69%	13	50	74%	R\$ 4.920.000,00	R\$ 15.000.000,00	67%	R\$ 215.626,50	R\$ 800.000,00	73%
6	8 6000	2	737	7500	90%	2531	4000	-37%	56,7	150	62%	13	50	74%	R\$ 5.424.000,00	R\$ 15.000.000,00	64%	R\$ 247.951,80	R\$ 800.000,00	69%
7	9 7000	2	860	7500	89%	2445	4000	-39%	66,15	150	56%	13	50	74%	R\$ 5.928.000,00	R\$ 15.000.000,00	60%	R\$ 280.277,10	R\$ 800.000,00	65%
8	10 8000	4	983	7500	87%	5328	4000	33%	37,8	150	75%	26	50	48%	R\$ 8.832.000,00	R\$ 15.000.000,00	41%	R\$ 366.602,40	R\$ 800.000,00	54%
9	11 9000	4	1119	7500	85%	5252	4000	31%	43,05	150	71%	26	50	48%	R\$ 9.392.000,00	R\$ 15.000.000,00	37%	R\$ 402.519,40	R\$ 800.000,00	50%
10	12 10000	4	1229	7500	84%	5192	4000	30%	47,25	150	69%	26	50	48%	R\$ 9.840.000,00	R\$ 15.000.000,00	34%	R\$ 431.525,00	R\$ 800.000,00	46%
11	13 11000	4	1365	7500	82%	5119	4000	28%	52,5	150	65%	26	50	48%	R\$ 10.400.000,00	R\$ 15.000.000,00	31%	R\$ 467.170,00	R\$ 800.000,00	42%
12	14 12000	4	1474	7500	80%	5063	4000	27%	56,7	150	62%	26	50	48%	R\$ 10.848.000,00	R\$ 15.000.000,00	28%	R\$ 495.903,60	R\$ 800.000,00	38%
13	15 13000	4	1611	7500	79%	4978	4000	24%	61,95	150	59%	26	50	48%	R\$ 11.408.000,00	R\$ 15.000.000,00	24%	R\$ 531.820,60	R\$ 800.000,00	34%
14	16 14000	4	1720	7500	77%	4890	4000	22%	66,15	150	56%	26	50	48%	R\$ 11.856.000,00	R\$ 15.000.000,00	21%	R\$ 560.554,20	R\$ 800.000,00	30%
15	17 15000	4	1829	7500	76%	4804	4000	20%	70,35	150	53%	26	50	48%	R\$ 12.304.000,00	R\$ 15.000.000,00	18%	R\$ 589.287,80	R\$ 800.000,00	26%
16	18 16000	8	1966	7500	74%	10656	4000	166%	37,8	150	75%	52	50	-4%	R\$ 17.664.000,00	R\$ 15.000.000,00	-18%	R\$ 733.204,80	R\$ 800.000,00	8%
17	19 17000	8	2075	7500	72%	10594	4000	165%	39,9	150	73%	52	50	-4%	R\$ 18.112.000,00	R\$ 15.000.000,00	-21%	R\$ 761.938,40	R\$ 800.000,00	5%
18	20 18000	8	2239	7500	70%	10503	4000	163%	43,05	150	71%	52	50	-4%	R\$ 18.784.000,00	R\$ 15.000.000,00	-25%	R\$ 805.038,80	R\$ 800.000,00	-1%
19	21 19000	8	2348	7500	69%	10443	4000	161%	45,15	150	70%	52	50	-4%	R\$ 19.232.000,00	R\$ 15.000.000,00	-28%	R\$ 833.772,40	R\$ 800.000,00	-4%
20	22 20000	8	2457	7500	67%	10384	4000	160%	47,25	150	69%	52	50	-4%	R\$ 19.680.000,00	R\$ 15.000.000,00	-31%	R\$ 862.506,00	R\$ 800.000,00	-8%
21	23 21000	8	2566	7500	66%	10325	4000	158%	49,35	150	67%	52	50	-4%	R\$ 20.128.000,00	R\$ 15.000.000,00	-34%	R\$ 891.339,60	R\$ 800.000,00	-11%
22	24 22000	8	2730	7500	64%	10238	4000	156%	52,5	150	65%	52	50	-4%	R\$ 20.800.000,00	R\$ 15.000.000,00	-39%	R\$ 934.340,00	R\$ 800.000,00	-17%
23	25 23000	8	2839	7500	62%	10181	4000	155%	54,6	150	64%	52	50	-4%	R\$ 21.248.000,00	R\$ 15.000.000,00	-42%	R\$ 963.073,60	R\$ 800.000,00	-20%
24	26 24000	8	2948	7500	61%	10125	4000	153%	56,7	150	62%	52	50	-4%	R\$ 21.696.000,00	R\$ 15.000.000,00	-45%	R\$ 991.807,20	R\$ 800.000,00	-24%
25	27 25000	8	3058	7500	59%	10069	4000	152%	58,8	150	61%	52	50	-4%	R\$ 22.144.000,00	R\$ 15.000.000,00	-48%	R\$ 1.020.540,80	R\$ 800.000,00	-28%
26	28 26000	8	3221	7500	57%	9957	4000	149%	61,95	150	59%	52	50	-4%	R\$ 22.816.000,00	R\$ 15.000.000,00	-52%	R\$ 1.063.641,20	R\$ 800.000,00	-33%
27	29 27000	8	3331	7500	56%	9867	4000	147%	64,05	150	57%	52	50	-4%	R\$ 23.264.000,00	R\$ 15.000.000,00	-55%	R\$ 1.092.374,80	R\$ 800.000,00	-37%
28	30 28000	8	3440	7500	54%	9779	4000	144%	66,15	150	56%	52	50	-4%	R\$ 23.712.000,00	R\$ 15.000.000,00	-58%	R\$ 1.121.108,40	R\$ 800.000,00	-40%
29	31 29000	8	3549	7500	53%	9693	4000	142%	68,25	150	55%	52	50	-4%	R\$ 24.160.000,00	R\$ 15.000.000,00	-61%	R\$ 1.149.842,00	R\$ 800.000,00	-44%
30	32 30000	8	3658	7500	51%	9608	4000	140%	70,35	150	53%	52	50	-4%	R\$ 24.608.000,00	R\$ 15.000.000,00	-64%	R\$ 1.178.575,60	R\$ 800.000,00	-47%
31																				
32																				
33																				
34																				
35																				
36																				
37																				
38	TS EndereçosASRS / TS EndereçosCB / TS EndereçosNA																			

Figura 63: Resultados da execução dos cenários para a solução AS/RS.

ID	A	B	Counter Balanced Forklift				E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T										
			Corredores		Area (m²)																		Throughput (UoH/Day)		Comprimento (m)		Largura (m)		Custo Instalação (R\$)		Custo Operacional (R\$)	
			Real	Req. (max)	% safety	Req. (min)																	% safety	Real	Req. (max)	% safety	Real	Req. (max)	% safety	Real	Req. (max)	% safety
1	2	Endereços	2	1815	7500	76%	789	4000	-80%	137,5	150	8%	13,2	50	74%	RS	520.000,00	RS 15.000.000,00	97%	RS	141.270,00	RS 800.000,00	82%									
3	4	2000	2	2730	7500	64%	643	4000	-84%	206,8	150	-38%	13,2	50	74%	RS	721.600,00	RS 15.000.000,00	95%	RS	204.406,08	RS 800.000,00	74%									
5	3000	2	3630	7500	52%	543	4000	-86%	275	150	-83%	13,2	50	74%	RS	920.000,00	RS 15.000.000,00	94%	RS	266.540,00	RS 800.000,00	67%										
7	4000	2	4545	7500	39%	470	4000	-88%	344,3	150	-130%	13,2	50	74%	RS	1.121.600,00	RS 15.000.000,00	93%	RS	329.676,08	RS 800.000,00	59%										
8	6000	2	5445	7500	27%	414	4000	-90%	412,5	150	-175%	13,2	50	74%	RS	1.320.000,00	RS 15.000.000,00	91%	RS	391.810,00	RS 800.000,00	51%										
9	7000	2	6360	7500	15%	370	4000	-91%	481,8	150	-221%	13,2	50	74%	RS	1.521.600,00	RS 15.000.000,00	90%	RS	454.946,08	RS 800.000,00	43%										
10	8000	4	7260	7500	3%	1087	4000	-73%	275	150	-83%	26,4	50	47%	RS	1.840.000,00	RS 15.000.000,00	88%	RS	533.980,00	RS 800.000,00	33%										
11	9000	4	8189	7500	-9%	1007	4000	-75%	310,2	150	-107%	26,4	50	47%	RS	2.044.800,00	RS 15.000.000,00	86%	RS	597.218,24	RS 800.000,00	25%										
12	10000	4	9090	7500	-21%	939	4000	-77%	344,3	150	-130%	26,4	50	47%	RS	2.243.200,00	RS 15.000.000,00	85%	RS	659.352,16	RS 800.000,00	18%										
13	11000	4	9990	7500	-33%	881	4000	-78%	378,4	150	-152%	26,4	50	47%	RS	2.441.600,00	RS 15.000.000,00	84%	RS	721.486,08	RS 800.000,00	10%										
14	12000	4	10890	7500	-45%	829	4000	-79%	412,5	150	-175%	26,4	50	47%	RS	2.640.000,00	RS 15.000.000,00	82%	RS	783.620,00	RS 800.000,00	2%										
15	13000	4	11819	7500	-58%	781	4000	-80%	447,7	150	-198%	26,4	50	47%	RS	2.844.800,00	RS 15.000.000,00	81%	RS	847.758,24	RS 800.000,00	-6%										
16	14000	4	12720	7500	-70%	740	4000	-81%	481,8	150	-221%	26,4	50	47%	RS	3.043.200,00	RS 15.000.000,00	80%	RS	909.892,16	RS 800.000,00	-14%										
17	15000	4	13620	7500	-82%	703	4000	-82%	515,9	150	-244%	26,4	50	47%	RS	3.241.600,00	RS 15.000.000,00	78%	RS	972.026,08	RS 800.000,00	-22%										
18	16000	8	14520	7500	-94%	2174	4000	-46%	275	150	-83%	52,8	50	-6%	RS	3.680.000,00	RS 15.000.000,00	75%	RS	1.066.160,00	RS 800.000,00	-33%										
19	17000	8	15449	7500	-106%	2090	4000	-48%	292,6	150	-95%	52,8	50	-6%	RS	3.884.800,00	RS 15.000.000,00	74%	RS	1.130.298,24	RS 800.000,00	-41%										
20	18000	8	16379	7500	-118%	2013	4000	-50%	310,2	150	-107%	52,8	50	-6%	RS	4.089.600,00	RS 15.000.000,00	73%	RS	1.194.436,48	RS 800.000,00	-49%										
21	19000	8	17250	7500	-130%	1946	4000	-51%	326,7	150	-118%	52,8	50	-6%	RS	4.281.600,00	RS 15.000.000,00	71%	RS	1.254.566,08	RS 800.000,00	-57%										
22	20000	8	18179	7500	-142%	1879	4000	-53%	344,3	150	-130%	52,8	50	-6%	RS	4.486.400,00	RS 15.000.000,00	70%	RS	1.318.704,32	RS 800.000,00	-65%										
23	21000	8	19108	7500	-155%	1816	4000	-55%	361,9	150	-141%	52,8	50	-6%	RS	4.691.200,00	RS 15.000.000,00	69%	RS	1.382.842,56	RS 800.000,00	-73%										
24	22000	8	19980	7500	-166%	1761	4000	-56%	378,4	150	-152%	52,8	50	-6%	RS	4.883.200,00	RS 15.000.000,00	67%	RS	1.442.972,16	RS 800.000,00	-80%										
25	23000	8	20909	7500	-179%	1706	4000	-57%	396	150	-164%	52,8	50	-6%	RS	5.088.000,00	RS 15.000.000,00	66%	RS	1.507.110,40	RS 800.000,00	-88%										
26	24000	8	21780	7500	-190%	1658	4000	-59%	412,5	150	-175%	52,8	50	-6%	RS	5.280.000,00	RS 15.000.000,00	65%	RS	1.567.240,00	RS 800.000,00	-96%										
27	25000	8	22709	7500	-203%	1609	4000	-60%	430,1	150	-187%	52,8	50	-6%	RS	5.484.800,00	RS 15.000.000,00	63%	RS	1.631.378,24	RS 800.000,00	-104%										
28	26000	8	23639	7500	-215%	1563	4000	-61%	447,7	150	-198%	52,8	50	-6%	RS	5.689.600,00	RS 15.000.000,00	62%	RS	1.695.516,48	RS 800.000,00	-112%										
29	27000	8	24510	7500	-227%	1522	4000	-62%	464,2	150	-209%	52,8	50	-6%	RS	5.881.600,00	RS 15.000.000,00	61%	RS	1.755.646,08	RS 800.000,00	-119%										
30	28000	8	25439	7500	-239%	1480	4000	-63%	481,8	150	-221%	52,8	50	-6%	RS	6.086.400,00	RS 15.000.000,00	59%	RS	1.819.784,32	RS 800.000,00	-127%										
31	29000	8	26368	7500	-252%	1441	4000	-64%	499,4	150	-233%	52,8	50	-6%	RS	6.291.200,00	RS 15.000.000,00	58%	RS	1.883.922,56	RS 800.000,00	-135%										
32	30000	8	27240	7500	-265%	1406	4000	-65%	515,9	150	-244%	52,8	50	-6%	RS	6.483.200,00	RS 15.000.000,00	57%	RS	1.944.052,16	RS 800.000,00	-143%										
33																																
34																																
35																																
36																																
37																																

Figura 64: Resultados da execução dos cenários para a solução com empilhadeiras contrabalanceadas.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
	Endereços	Corredores	Narrow Aisled		Area (m²)		Throughput (UoH/Dwy)		Comprimento (m)		Largura (m)		Custo Instalação (R\$)		Custo Operacional (R\$)		% safety		Req. (max)	
			Real	Req. (max)	% safety	Real	Req. (min)	% safety	Real	Req. (max)	% safety	Real	Req. (max)	% safety	Real	Req. (max)	% safety	Real	Req. (max)	% safety
2	3000	2	850	7500	-68%	1276	4000	-68%	92,4	150	82%	9,2	50	82%	R\$ 100.544,64	R\$ 800.000,00	94%	R\$ 100.544,64	R\$ 800.000,00	87%
3	3000	2	1265	7500	-83%	1063	4000	-73%	137,5	150	8%	9,2	50	82%	R\$ 139.370,00	R\$ 800.000,00	92%	R\$ 139.370,00	R\$ 800.000,00	83%
5	3000	2	1690	7500	-77%	908	4000	-77%	183,7	150	-22%	9,2	50	82%	R\$ 179.142,32	R\$ 800.000,00	91%	R\$ 179.142,32	R\$ 800.000,00	78%
6	4000	2	2115	7500	-72%	793	4000	-80%	229,9	150	-55%	9,2	50	82%	R\$ 218.914,64	R\$ 800.000,00	87%	R\$ 218.914,64	R\$ 800.000,00	73%
7	6000	2	2530	7500	-66%	705	4000	-82%	275	150	-83%	9,2	50	82%	R\$ 1.900.000,00	R\$ 15.000.000,00	89%	R\$ 257.740,00	R\$ 800.000,00	68%
8	7000	2	2955	7500	-61%	634	4000	-84%	321,2	150	-114%	9,2	50	82%	R\$ 2.152.000,00	R\$ 15.000.000,00	86%	R\$ 297.512,32	R\$ 800.000,00	63%
9	8000	4	3380	7500	-55%	1817	4000	-55%	183,7	150	-22%	18,4	50	63%	R\$ 358.284,64	R\$ 800.000,00	81%	R\$ 358.284,64	R\$ 800.000,00	55%
10	9000	4	3805	7500	-49%	1693	4000	-58%	206,8	150	-38%	18,4	50	63%	R\$ 398.056,96	R\$ 800.000,00	80%	R\$ 398.056,96	R\$ 800.000,00	50%
11	10000	4	4230	7500	-44%	1586	4000	-60%	229,9	150	-53%	18,4	50	63%	R\$ 437.829,28	R\$ 800.000,00	78%	R\$ 437.829,28	R\$ 800.000,00	45%
12	11000	4	4655	7500	-38%	1491	4000	-63%	253	150	-69%	18,4	50	63%	R\$ 477.601,60	R\$ 800.000,00	76%	R\$ 477.601,60	R\$ 800.000,00	40%
13	12000	4	5060	7500	-33%	1411	4000	-65%	275	150	-83%	18,4	50	63%	R\$ 515.480,00	R\$ 800.000,00	75%	R\$ 515.480,00	R\$ 800.000,00	36%
14	13000	4	5485	7500	-27%	1335	4000	-67%	298,1	150	-96%	18,4	50	63%	R\$ 555.252,32	R\$ 800.000,00	73%	R\$ 555.252,32	R\$ 800.000,00	31%
15	14000	4	5910	7500	-21%	1267	4000	-68%	321,2	150	-114%	18,4	50	63%	R\$ 595.024,64	R\$ 800.000,00	71%	R\$ 595.024,64	R\$ 800.000,00	26%
16	15000	4	6335	7500	-16%	1206	4000	-70%	344,3	150	-130%	18,4	50	63%	R\$ 634.796,96	R\$ 800.000,00	70%	R\$ 634.796,96	R\$ 800.000,00	21%
17	16000	8	6760	7500	-10%	3634	4000	-9%	183,7	150	-22%	36,8	50	26%	R\$ 716.569,28	R\$ 800.000,00	63%	R\$ 716.569,28	R\$ 800.000,00	10%
18	17000	8	7205	7500	-4%	3500	4000	-12%	195,8	150	-31%	36,8	50	26%	R\$ 758.235,52	R\$ 800.000,00	61%	R\$ 758.235,52	R\$ 800.000,00	5%
19	18000	8	7610	7500	-1%	3387	4000	-15%	206,8	150	-38%	36,8	50	26%	R\$ 796.113,92	R\$ 800.000,00	59%	R\$ 796.113,92	R\$ 800.000,00	0%
20	19000	8	8015	7500	-7%	3281	4000	-18%	217,8	150	-45%	36,8	50	26%	R\$ 833.992,32	R\$ 800.000,00	58%	R\$ 833.992,32	R\$ 800.000,00	-4%
21	20000	8	8460	7500	-13%	3172	4000	-21%	229,9	150	-53%	36,8	50	26%	R\$ 875.658,56	R\$ 800.000,00	56%	R\$ 875.658,56	R\$ 800.000,00	-9%
22	21000	8	8865	7500	-18%	3078	4000	-23%	240,9	150	-61%	36,8	50	26%	R\$ 913.536,96	R\$ 800.000,00	54%	R\$ 913.536,96	R\$ 800.000,00	-14%
23	22000	8	9310	7500	-24%	2982	4000	-25%	253	150	-69%	36,8	50	26%	R\$ 955.203,20	R\$ 800.000,00	53%	R\$ 955.203,20	R\$ 800.000,00	-19%
24	23000	8	9715	7500	-30%	2899	4000	-28%	264	150	-76%	36,8	50	26%	R\$ 993.081,60	R\$ 800.000,00	51%	R\$ 993.081,60	R\$ 800.000,00	-24%
25	24000	8	10120	7500	-35%	2821	4000	-29%	275	150	-83%	36,8	50	26%	R\$ 1.030.960,00	R\$ 800.000,00	49%	R\$ 1.030.960,00	R\$ 800.000,00	-29%
26	25000	8	10565	7500	-41%	2740	4000	-31%	287,1	150	-91%	36,8	50	26%	R\$ 1.072.626,24	R\$ 800.000,00	48%	R\$ 1.072.626,24	R\$ 800.000,00	-34%
27	26000	8	10970	7500	-46%	2670	4000	-33%	298,1	150	-99%	36,8	50	26%	R\$ 1.110.504,64	R\$ 800.000,00	46%	R\$ 1.110.504,64	R\$ 800.000,00	-39%
28	27000	8	11415	7500	-52%	2597	4000	-35%	310,2	150	-107%	36,8	50	26%	R\$ 1.152.170,88	R\$ 800.000,00	44%	R\$ 1.152.170,88	R\$ 800.000,00	-44%
29	28000	8	11820	7500	-58%	2534	4000	-37%	321,2	150	-114%	36,8	50	26%	R\$ 1.190.049,28	R\$ 800.000,00	43%	R\$ 1.190.049,28	R\$ 800.000,00	-49%
30	29000	8	12265	7500	-64%	2469	4000	-38%	333,3	150	-122%	36,8	50	26%	R\$ 1.231.715,52	R\$ 800.000,00	41%	R\$ 1.231.715,52	R\$ 800.000,00	-54%
31	30000	8	12670	7500	-69%	2412	4000	-40%	344,3	150	-130%	36,8	50	26%	R\$ 1.269.593,92	R\$ 800.000,00	39%	R\$ 1.269.593,92	R\$ 800.000,00	-59%
32																				
33																				
34																				
35																				
36																				
37																				
38																				

Figura 65: Resultados da execução dos cenários para a solução com empilhadeiras tri-laterais para corredores estreitos.

4.2 Comparativo de soluções em função da área ocupada

De forma análoga à seção anterior, 4.1, definem-se as seguintes etapas para o método proposto neste trabalho, *MTcsl*, em sua utilização neste segundo estudo de caso:

1. Reaproveitamento do modelo de referência *MDref*, elaborado para sistemas de armazenamento logístico;
2. Construção de um estudo de viabilidade *TSdf* para a produção do comparativo de soluções, utilizando *MDref*, em função da área ocupada por cada solução.

4.2.1 Reutilização do modelo de referência

Este segundo estudo de caso tem como objetivo simular não apenas uma situação real de projeto de sistemas de armazenamento logísticos (neste estudo denominado *Pdf* - *dimensional footprint*), no qual o método *MTcsl* tem aplicação, mas também seu uso recorrente, onde um novo modelo de referência não precisa ser montado, do início, a cada novo projeto.

Assim, este segundo estudo de caso fará uso do modelo de referência *MDref* exatamente conforme montado para o estudo anterior, seção 4.1. Eventuais alterações que *MDref* deva sofrer serão tratadas após a elicitação dos requisitos do projeto em questão, *Pdf*, através dos procedimentos de extensão abordados no capítulo 3, seção 3.2.1.2.

4.2.2 Construção do segundo estudo de viabilidade

Seguindo os mesmos passos do estudo anterior, este estudo de viabilidade *TSdf* deve ser iniciado com a apreensão do conjunto de requisitos de projeto, *LRdf*, junto ao cliente. Seja *LRdf* dado como se segue:

- "A empresa deve adquirir um novo terreno para a expansão de sua rede de centros de distribuição. Tal centro deverá oferecer, minimamente, 40.000 posições para armazenamento de pallets padrão PBR, e 8000 de capacidade de giro diário. Os custos de instalação e operação não deverão superar os R\$15.000.000,00 e R\$800.000,00, respectivamente². Os turnos de trabalho deverão ser divididos em três, de oito horas cada um, e considerando 85% de eficiência nos trabalhos. Os dois primeiros turnos deverão ser realizados em ciclos duplos, e o terceiro em ciclo simples, somente com entrada de pallets."

Figura 66: Requisitos não-funcionais para o estudo *TSdf*.

²custos iguais aos do estudo anterior

Figura 67: Premissas para o estudo *TSdf*.

O objetivo do estudo *TSdf*, bem como a derivação dos requisitos de LRdf nos subconjuntos não-funcionais e premissas, fica conforme apresentado nas figuras 66, 67 e 68. Requisitos funcionais e de interface não foram mencionados portanto, novamente, não implicarão em alterações sobre *MDref*.

Ao contrário do estudo anterior, o mapeamento dos conjuntos derivados de LRdf para o *template* TR mostra que há premissas no primeiro que não puderam ser mapeadas no segundo (figura 69). Isto implica que o modelo de referência original, *MDref*, não poderá ser utilizado neste estudo *TSdf* sem que passe antes por um processo de extensão.

Figura 68: Objetivo do estudo *TSdf*.

Figura 69: Mapeamento de requisitos do projeto *Pdf* (linhas) para o *template* TR (colunas). Três premissas não puderam ser mapeadas.

Com novas premissas passando figurar em TR, *MDref* deverá ser estendido em um novo modelo *MDref'*, de modo a abarcá-las. Como a extensão se dá apenas sobre premissas, não haverá novas propriedades de valor, operações nem portas em *MDref'*. Entretanto, deverão ser averiguadas quais alterações em equações e instanciações es-

sas novas premissas promovem, preferencialmente com a ajuda de um especialista de domínio.

Neste estudo de caso, averiguou-se que as novas premissas afetam exclusivamente a MOE '*throughput*' (giro), pois mudam a forma de se calcular o tempo de ciclo dos transportadores. Tal variável, denominada '*ShuttleCycleTime*', se encontra dentro do bloco *RackStructure*, e é calculada pelo bloco de equação *ShuttleThroughputCalcDoubleCycle* (figura 70). Tal bloco deverá ser redefinido na nova extensão *MDref*', a fim de contemplar em sua equação as novas premissas de turnos, ciclos e eficiência. O equacionamento completo do novo bloco será omitido a fim de tornar menos complexa esta passagem do trabalho. Não obstante, a operação de extensão, com a redefinição do bloco *ShuttleThroughputCalcDoubleCycle*, pode ser analisada na figura 71, onde *MDref*' aparece identificado como *MDref-v2*. Com o mesmo arranjo de relações de generalização, seria possível a redefinição de outras propriedades, se assim fosse necessário.

Feita a extensão de *MDref*, a montagem do estudo *TSdf* pode ser retomada, pois agora tem-se a segurança de que todos os requisitos elicitados para o projeto *Pdf* em questão serão mapeáveis nos subconjuntos de TR.

Com relação aos parâmetros de entrada, aqui novamente recai-se na situação de número um da listagem apresentada no capítulo 3, subseção 3.2.1.1, onde os parâmetros de entrada possíveis para *TSdf* consistem em todos os requisitos do subconjunto *TRnf* de TR que não foram mapeados por *LRnf*, que vem a ser o subconjunto de requisitos não-funcionais derivados do total de requisitos de projeto *LRdf*. Tal situação pode ser analisada na figura 72, onde se escolhe os parâmetros 'largura' e 'comprimento' com entrada, o que indiretamente acaba por determinar o parâmetro 'área', a partir do qual o cliente deseja ver os comparativos.

Após a definição dos parâmetros de entrada, transfere-se os requisitos não funcionais *LRnf* para o bloco encarregado de armazená-los, sob o formato de propriedades

Figura 70: Diagrama paramétrico do bloco *RackStructure* para cálculo de tempos de ciclo.

Figura 71: Extensão de *MDref* através de relações de generalização.

de valor, conforme mostrado no estudo anterior.

As figuras 74a e 74b mostram, respectivamente o diagrama de blocos e o bloco

Figura 72: Requisitos candidatos a parâmetros de entrada para o estudo *TSdf*.

Figura 73: Requisitos apreendidos em bloco para o estudo *TSdf*.

principal, em detalhes, para o estudo em questão *TSdf*.

O equacionamento se dá de forma semelhante ao estudo apresentado na seção anterior. As conversões dos parâmetros de entrada de *TSdf*, em termos das propriedades independentes de *MDref* podem ser visualizadas nas imagens 75a e 75b. O restante dos diagramas, incluindo os cálculos das margens de segurança dos requisitos (MOS) é equivalente ao do primeiro estudo.

Avançando para a instanciação de *TSdf*, serão produzidos comparativos para as mesmas três soluções do estudo anterior: empilhadeiras contrabalanceadas, trilaterais para corredores estreitos e sistemas AS/RS com trans-elevadores. A figura 76 mostra um exemplo de instanciação de *TSdf* utilizando a solução AS/RS.

(a) Diagrama de blocos do estudo de viabilidade *TSdf* (b) Bloco principal do estudo de viabilidade *TSdf*, com detalhes para os parâmetros de entrada e de saída.Figura 74: Os blocos constituintes do estudo de viabilidade *TSdf*.

Neste estudo de caso foram produzidos comparativos, considerando cada uma das três instâncias de solução, para doze cenários distintos ao invés de vinte e nove, como no primeiro estudo. Pelo comparativo produzido, nota-se que nenhuma das três soluções foi capaz de atender aos requisitos de projeto *Pdf* (figuras 77, 78 e 79).

(a)

(b)

Figura 75: Equacionamento de conversão de parâmetros do estudo de viabilidade *TSdf*.

Figura 76: Instanciação do estudo de viabilidade *TSdf*.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
	ASRS															
Comprimento	Largura	Posições Pallet		Throughput (UoH/Day)		Custo Instalação (R\$)		Custo Operacional (R\$)		Corredores						
		Area	Real	Req. (min)	% safety	Real	Req. (min)	% safety	Real	Req. (max)	% safety	Real				
1																
2																
3																
4	60	20	1200	9600	40000	-76%	2823	8000	-65%	R\$ 7.500.000,00	R\$ 15.000.000,00	50%	R\$ 368.610,00	R\$ 800.000,00	54%	3
5	80	20	1600	12800	40000	-68%	2598	8000	-68%	R\$ 9.100.000,00	R\$ 15.000.000,00	39%	R\$ 471.230,00	R\$ 800.000,00	41%	3
6	80	30	2400	17067	40000	-57%	3464	8000	-57%	R\$ 12.133.333,33	R\$ 15.000.000,00	19%	R\$ 628.306,67	R\$ 800.000,00	21%	4
7	100	30	3000	21333	40000	-47%	3206	8000	-60%	R\$ 14.266.666,67	R\$ 15.000.000,00	5%	R\$ 765.133,33	R\$ 800.000,00	4%	4
8	100	40	4000	32000	40000	-20%	4810	8000	-40%	R\$ 21.400.000,00	R\$ 15.000.000,00	-43%	R\$ 1.147.700,00	R\$ 800.000,00	-43%	6
9	100	50	5000	37333	40000	-7%	5611	8000	-30%	R\$ 24.966.666,67	R\$ 15.000.000,00	-66%	R\$ 1.338.983,33	R\$ 800.000,00	-67%	7
10	125	80	10000	80000	40000	100%	8803	8000	10%	R\$ 50.800.000,00	R\$ 15.000.000,00	-239%	R\$ 2.808.500,00	R\$ 800.000,00	-251%	12
11	150	100	15000	120000	40000	200%	10142	8000	27%	R\$ 73.500.000,00	R\$ 15.000.000,00	-390%	R\$ 4.152.000,00	R\$ 800.000,00	-419%	15
12	180	100	18000	144000	40000	260%	9272	8000	16%	R\$ 85.500.000,00	R\$ 15.000.000,00	-470%	R\$ 4.921.650,00	R\$ 800.000,00	-515%	15
13	200	100	20000	160000	40000	300%	8679	8000	8%	R\$ 93.500.000,00	R\$ 15.000.000,00	-523%	R\$ 5.434.750,00	R\$ 800.000,00	-579%	15
14	200	120	24000	192000	40000	380%	10414	8000	30%	R\$ 112.200.000,00	R\$ 15.000.000,00	-648%	R\$ 6.521.700,00	R\$ 800.000,00	-715%	18
15	200	150	30000	245333	40000	513%	13307	8000	66%	R\$ 143.366.666,67	R\$ 15.000.000,00	-856%	R\$ 8.333.283,33	R\$ 800.000,00	-942%	23
16																
17																
18																
19																
20																
21																
22																
23																
24																
25																
26																
27																
28																
29																
30																
31																
32																

Figura 77: Resultados da execução dos cenários para a solução AS/RS.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
				Counter Balanced	Posições Pallet											
	Comprimento	Largura	Area	Real	Req. (min)	% safety	Real	Req.(min)	% safety	Real	Req. (max)	% safety	Real	Req. (max)	% safety	Corredores
1																Real
2																3
3																3
4	60	20	1200	1309	40000	-97%	1191	8000	-85%	R\$ 396.818,18	R\$ 15.000.000,00	97%	R\$ 99.994,91	R\$ 800.000,00	88%	3
5	80	20	1600	1745	40000	-96%	1095	8000	-86%	R\$ 484.090,91	R\$ 15.000.000,00	97%	R\$ 127.326,55	R\$ 800.000,00	84%	3
6	80	30	2400	2327	40000	-94%	1459	8000	-82%	R\$ 645.454,55	R\$ 15.000.000,00	96%	R\$ 169.768,73	R\$ 800.000,00	79%	4
7	100	30	3000	2909	40000	-93%	1350	8000	-83%	R\$ 761.818,18	R\$ 15.000.000,00	95%	R\$ 206.210,91	R\$ 800.000,00	74%	4
8	100	40	4000	4364	40000	-89%	2025	8000	-75%	R\$ 1.142.727,27	R\$ 15.000.000,00	92%	R\$ 309.316,36	R\$ 800.000,00	61%	6
9	100	50	5000	5091	40000	-87%	2363	8000	-70%	R\$ 1.333.181,82	R\$ 15.000.000,00	91%	R\$ 360.869,09	R\$ 800.000,00	55%	7
10	125	80	10000	10909	40000	-73%	3703	8000	-54%	R\$ 2.721.818,18	R\$ 15.000.000,00	82%	R\$ 755.290,91	R\$ 800.000,00	6%	12
11	150	100	15000	16364	40000	-59%	4263	8000	-47%	R\$ 3.947.727,27	R\$ 15.000.000,00	74%	R\$ 1.114.936,36	R\$ 800.000,00	-39%	15
12	180	100	18000	19636	40000	-51%	3894	8000	-51%	R\$ 4.602.272,73	R\$ 15.000.000,00	69%	R\$ 1.319.923,64	R\$ 800.000,00	-65%	15
13	200	100	20000	21818	40000	-45%	3682	8000	-54%	R\$ 5.038.636,36	R\$ 15.000.000,00	66%	R\$ 1.456.581,82	R\$ 800.000,00	-82%	15
14	200	120	24000	26182	40000	-35%	4418	8000	-45%	R\$ 6.046.363,64	R\$ 15.000.000,00	60%	R\$ 1.747.898,18	R\$ 800.000,00	-118%	18
15	200	150	30000	32000	40000	-20%	5400	8000	-33%	R\$ 7.390.000,00	R\$ 15.000.000,00	51%	R\$ 2.136.320,00	R\$ 800.000,00	-167%	22
16																
17																
18																
19																
20																
21																
22																
23																
24																
25																
26																
27																
28																
29																
30																
31																
32																
	TS_AreaASRS	TS_AreaCB	TS_ArealIA													

Figura 78: Resultados da execução dos cenários para a solução com empilhadeiras contrabalanceadas.

	A	B	C	D	Narrow Aisled		E	F	G	H	I	J	K	L	M	N	O	P
	Comprimento	Largura	Area	Real	Req. (min)	Posições Pallet	Req. (min)	% safety	Real	Req. (min)	Throughput (UoH/Day)	Real	Req. (max)	% safety	Real	Req. (max)	% safety	Corredores Real
1	60	20	1200	2618	40000	-93%	8000	-73%	R\$ 1.254.545,45	R\$ 15.000.000,00	92%	R\$ 134.804,73	R\$ 800.000,00	83%				4
2	80	20	1600	3491	40000	-91%	8000	-75%	R\$ 1.472.727,27	R\$ 15.000.000,00	90%	R\$ 169.239,64	R\$ 800.000,00	79%				4
3	80	30	2400	5236	40000	-87%	8000	-62%	R\$ 2.209.090,91	R\$ 15.000.000,00	85%	R\$ 253.859,45	R\$ 800.000,00	68%				6
4	100	30	3000	6545	40000	-84%	8000	-65%	R\$ 2.536.363,64	R\$ 15.000.000,00	83%	R\$ 305.511,82	R\$ 800.000,00	62%				6
5	100	40	4000	8727	40000	-78%	8000	-54%	R\$ 3.381.818,18	R\$ 15.000.000,00	77%	R\$ 407.349,09	R\$ 800.000,00	49%				8
6	100	50	5000	10909	40000	-73%	8000	-42%	R\$ 4.227.272,73	R\$ 15.000.000,00	72%	R\$ 509.186,36	R\$ 800.000,00	36%				10
7	125	80	10000	23182	40000	-42%	8000	-11%	R\$ 8.345.454,55	R\$ 15.000.000,00	44%	R\$ 1.048.552,27	R\$ 800.000,00	-31%				17
8	150	100	15000	34364	40000	-14%	8000	0%	R\$ 11.740.909,09	R\$ 15.000.000,00	22%	R\$ 1.521.249,55	R\$ 800.000,00	-90%				21
9	180	100	18000	41236	40000	3%	8000	-10%	R\$ 13.459.090,91	R\$ 15.000.000,00	10%	R\$ 1.792.424,45	R\$ 800.000,00	-124%				21
10	200	100	20000	45818	40000	15%	8000	-15%	R\$ 14.604.545,45	R\$ 15.000.000,00	3%	R\$ 1.973.207,73	R\$ 800.000,00	-147%				21
11	200	120	24000	56727	40000	42%	8000	5%	R\$ 18.081.818,18	R\$ 15.000.000,00	-21%	R\$ 2.443.019,09	R\$ 800.000,00	-205%				26
12	200	150	30000	69818	40000	75%	8000	30%	R\$ 22.254.545,45	R\$ 15.000.000,00	-48%	R\$ 3.006.792,73	R\$ 800.000,00	-276%				32
13																		
14																		
15																		
16																		
17																		
18																		
19																		
20																		
21																		
22																		
23																		
24																		
25																		
26																		
27																		
28																		
29																		
30																		
31																		
32																		

Figura 79: Resultados da execução dos cenários para a solução com empilhadeiras tri-laterais para corredores estreitos.

5 CONCLUSÕES

O problema que este trabalho buscou resolver foi o processo de escolha de soluções logísticas de armazenamento e recuperação.

A abordagem utilizada para resolver o problema apresentado foi a construção de um método para seleção de soluções logísticas de armazenamento, seguindo os princípios da engenharia de sistemas baseada em modelos (MBSE).

Esta abordagem contribuiu com conhecimentos inéditos através da complementação de trabalhos já existentes que propuseram objetivos semelhantes, em particular (BAKER; CANESSA, 2009) e (MCGINNIS; SCHMIDT; SPEE, 2013). A contribuição de Baker e Canessa (2009), através de ampla pesquisa realizada junto a empresas responsáveis por empreendimentos logísticos, apontou para a ausência de métodos de referência para o design de centros de distribuição e de sistemas de armazenamento. Apontou que, até o momento da elaboração de seu trabalho, as metodologias existentes eram essencialmente *ad-hoc*, fazendo uso de ferramentas de software genéricas. Já McGinnis, Schmidt e Spee (2013) seguem na direção de tratar o design de centros de distribuição como um problema pertencente ao escopo da engenharia de sistemas, e fornece uma abordagem baseada em modelos orientados a objeto, descritos através da linguagem OMG SysML™, para tal.

O método proposto neste trabalho consistiu na construção de dois artefatos distintos: um modelo de referência para sistemas logísticos de armazenamento e um conjunto de estudos de viabilidade (*trade-studies*). Os dois artefatos foram construídos

como modelos orientados a objeto, expressos na linguagem OMG SysML™, e foram montados dentro da ferramenta CASE MagicDraw™.

A aplicação do método proposto para se escolher uma solução de armazenamento logístico também ocorre dentro da ferramenta CASE utilizada para a construção dos artefatos, e consiste na sequência de quatro etapas apresentadas na seção 1.2 Objetivo: apreensão dos requisitos através do *template* formulado para tal; criação de instâncias dos *trade-studies* e do modelo de referencia, sendo que para este último serão tantas instâncias quanto forem as soluções de armazenamento logístico que se desejar avaliar; execução das equações através do resolvidor numérico da ferramenta CASE e comparação dos resultados através de softwares auxiliares para visualização de dados, como o MS Excel™.

Os resultados foram considerados satisfatórios, pois foi possível endereçar, ao longo do capítulo 3 Desenvolvimento, cada uma das condições C_1 a C_7 impostas ao novo método na seção 1.2 Objetivo, que assim o foram com o intuito de avaliá-lo em relação aos métodos tradicionais *ad-hoc*, sem abordagem sistêmica e baseados essencialmente em documentos.

Ao fazer uso do método proposto neste trabalho, o usuário tem à sua disposição informações de domínios distintos, como financeiro, estrutural ou dinâmico, dispostas graficamente dentro de um único ambiente, representado pela ferramenta CASE (condições C_1 e C_2). Junto a isso, terá a possibilidade de criar quantas instâncias de soluções de armazenamento logístico julgar necessário, sem a necessidade de promover alterações no modelo de referencia, ou criá-lo repetidas vezes (condição C_3 , figura 58). E além de permitir a discriminação de cada uma das equações utilizadas através dos *constraint blocks*, ao invés de deixá-las ocultas em células de planilhas, o método proposto também prevê a execução das mesmas e a exportação de seus resultados para ferramentas externas, fazendo o casamento entre a modelagem gráfica do sistema de armazenamento logístico, e sua análise numérica (condições C_4 e C_5). Por fim, o

método ainda permite mecanismos para realizar sua extensão, quando assim for necessário, por conta de mudanças de premissas ou de requisitos que não se adequam ao *template* proposto (condição C₆, figura 71), e também mecanismos de rastreabilidade (condição C₇), fornecendo informações gráficas a respeito do que está se passando no interior do modelo de referência e de seus *trade-studies*, como requisitos não atendidos (figura 40), ou relações de dependência entre propriedades e equações (figura 43).

Em resumo, as contribuições deste trabalho podem ser sumarizadas, em ordem de relevância, como se segue:

1. Um método de apoio à tomada de decisão referente à escolha de sistemas de armazenamento logístico, que segue os princípios da MBSE.
2. Um modelo de referência para sistemas de armazenamento logístico, que pode se prestar a outros fins que não o comparativo de soluções, mas sim servir como ponto de partida para o design de novas.

Mesmo produzindo bons resultados, o trabalho aqui apresentado poderia ter desempenhado melhor em pontos como a organização do modelo, e no nível de detalhamento do mesmo. Acredita-se que com um modelo de referência mais detalhado sobre sistemas de armazenamento logístico, *trade-studies* mais complexos e aderentes à realidade poderiam ter sido montados. Para tal, mais pesquisas de campo poderiam ter sido realizadas, e mais equipes habituadas às técnicas da MBSE, entrevistadas. O tempo disponível foi o principal fator limitante para ambos. Não obstante, o artigo (PEARCE; FRIEDENTHAL, 2013), a dissertação (PLEGT, 2011), e os livros (R.KARBAN et al., 2011) e (FRIEDENTHAL; MOORE; STEINER, 2012) forneceram boas orientações em termos de organização do modelo em pacotes e na sequência de montagem do mesmo, apontando também os principais relatórios a serem extraídos. Já o livro (BARTHOLDI; HACKMAN, 2014) foi bastante relevante como fonte de conhecimentos primários a cerca do funcionamento dos centros de distribuição.

Como trabalhos futuros, e com base na bibliografia pesquisada, ficam por fazer o restante dos modelos de referência em OMG SysMLTM dos demais setores dos centros de distribuição, como o *Picking* por exemplo, que é o que possui maior peso nos custos operacionais. E também a elaboração de métodos, análogos ao apresentado neste trabalho, mas para a escolha de soluções nesses outros setores. Pode-se pensar, num médio prazo, na elaboração de bibliotecas de modelos e métodos, orientados a objeto, a servirem como referência nos projetos de centros de distribuição inteiros, minimizando as decisões tomadas exclusivamente de forma empírica e transformando o conhecimento tácito de especialista de domínio em conhecimentos explícitos, disponibilizados através da MBSE a toda equipe encarregada dos projetos.

REFERÊNCIAS

- ABDOLI, S.; KARA, S. Designing Warehouse Logical Architecture by Applying Object Oriented Model Based System Engineering. *Procedia CIRP*, Elsevier B.V., v. 50, p. 713–718, 2016. ISSN 22128271. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S2212827116302967>.
- BAKER, P.; CANESSA, M. Warehouse design: A structured approach. *European Journal of Operational Research*, Elsevier B.V., v. 193, n. 2, p. 425–436, 2009. ISSN 03772217. Disponível em: <http://dx.doi.org/10.1016/j.ejor.2007.11.045>.
- BARTHOLDI, J. J.; HACKMAN, S. T. *Warehouse & Distribution Science*. Release 0. [s.n.], 2014. Disponível em: www.warehouse-science.com.
- BKCASE Editorial Board. *Guide to the Systems Engineering Body of Knowledge (SEBoK)*. Hoboken, NJ: The Trustees of the Stevens Institute of Technology ©2016, 2014. ISBN 1800843232. Disponível em: <http://sebokwiki.org>.
- DACO. *Selective Pallet Rack Shelving Systems*. 2016. <http://www.dacocorp.com/selective-p-302-l-en.html>. Acessado em: Sep/2016.
- DICKERSON, C. E.; MAVRIS, D. A brief history of models and model based systems engineering and the case for relational orientation. *IEEE Systems Journal*, v. 7, n. 4, p. 581–592, 2013. ISSN 19328184.
- ELBOWROOM. *Pallet Storage Solutions*. 2016. <http://elbowroom.com.au>. Acessado em: Sep/2016.
- ELLINGER, M. How to Choose an Order-Picking System. *International Material Handling Research Colloquium*, 2012.
- EQUIPAMENTOS, N. D. de. *Empilhadeiras VRE -1.25 a 1.5 toneladas*. 2016. http://www.neq.com.br/produtos/bt/empilhadeiras-trilaterais/modelo/1-BT_VRE. Acessado em: Sep/2016.
- ESTEFAN, J. A. Survey of Model-Based Systems Engineering (MBSE) Methodologies 2 . Differentiating Methodologies from Processes , Methods , and Lifecycle Models. *Environment*, 2008.
- FRIEDENTHAL, S.; MOORE, A.; STEINER, R. *A practical guide to SysML: the systems modeling language*. Morgan Kaufmann. [S.l.: s.n.], 2012. ISBN 978-0-12-385206-9.
- GOETSCHALCKX, M.; MITAL, P.; HUANG, E. A framework for the robust design of unit load storage systems. *International Material Handling Research Colloquium*, p. 1–21, 2014.

GU, J.; GOETSCHALCKX, M.; MCGINNIS, L. F. Research on warehouse operation: A comprehensive review. *European Journal of Operational Research*, v. 177, n. 1, p. 1–21, 2007. ISSN 03772217.

James M., A. J.; MELLER, R. D.; John A., W. J. Empirically-Based Warehouse Design: can academics accept such approach? *11th International Material Handling Research Colloquium*, 2010. Disponível em: <<http://www.mhi.org/downloads/learning/cicmhe/colloquium/2010/apple.pdf>>.

LIMÈRE, V. et al. In-plant logistics systems modeling with sysml. *ESM 2010 - 2010 European Simulation and Modelling Conference*, n. July, p. 383–387, 2010. Disponível em: <<http://www.scopus.com/inward/record.url?eid=2-s2.0-84898602815&partnerID=tZOtx3y1>>.

LLC, I. *ParaMagic Tutorials*. [S.l.: s.n.], 2014. 1-88 p.

LYKINS; FRIEDENTHAL; MEILICH. Adapting UML for an Object-Oriented Systems Engineering Method (OOSEM). *Proceedings of the INCOSE International Symposium*, n. Dockerill, p. 490–497, 2000.

MCGINNIS, L. F. An object oriented and axiomatic theory of warehouse design. *12th International Material Handling Research Colloquium*, 2012.

_____. Integrating analysis into a warehouse design workflow: Design process for material handling systems. *International Material Handling Research Colloquium*, 2014.

MCGINNIS, L. F.; SCHMIDT, M.; SPEE, D. Model Based Systems Engineering and Warehouse Design. In: *International Logistics Science Conference (ILSC) 2013*. [S.l.: s.n.], 2013. p. 161–178. ISBN 9783319013770.

PEARCE, P.; FRIEDENTHAL, S. A Practical Approach For Modelling Submarine Subsystem Architecture In SysML. *Submarine Institute of Australia Science, Technology & Engineering Conference*, p. 347–360, 2013.

PEARCE, P.; HAUSE, M. Model-Based Submarine Design. *SETE and APCOSE 2012*, 2012.

PLEGT, T. *Model Based Systems Engineering in Warehouse Design: Automatic report generation from SysML models*. [S.l.], 2011. 1–26 p. Disponível em: <<http://essay.utwente.nl/60138/>>.

PROVDANOV, C. C.; FREITAS, E. C. D. *Metodologia do trabalho científico: métodos e técnicas da pesquisa e do trabalho acadêmico*. [s.n.], 2013. 276 p. ISSN 1098-6596. ISBN 9788577171583. Disponível em: <<http://www.feevale.br/Comum/midias/8807f05a-14d0-4d5b-b1ad-1538f3aef538/E-bookMetodologiadoTrabalhoCientifico.pdf>>.

R.KARBAN et al. *MBSE Initiative - SE2 Challenge Team Change record. COOKBOOK FOR MBSE WITH SYSML*. [S.l.: s.n.], 2011. 1–129 p.

SOLUTIONS, B. *Unit Load AS/RS*. 2016. <<https://www.bastiansolutions.com/solutions/technology/asrs/unit-load-asrs>>. Acessado em: Sep/2016.

APÊNDICE A - A LINGUAGEM SysML

A SysML, cujo nome oficial é OMG SysMLTM, é uma linguagem gráfica de modelagem, multi-propósito, que dá suporte à análise, especificação, design, verificação e validação de sistemas complexos de engenharia. Tais sistemas podem incluir hardware, software, dados, pessoas, procedimentos, infraestruturas, e demais elementos pertencentes tanto a sistemas artificiais quanto naturais. O intuito da linguagem é auxiliar na arquitetura de sistemas e na especificação de seus componentes, para que estes possam então ser detalhados nas linguagens e ferramentas específicas de cada um de seus domínios, como a linguagem UML para softwares, ou ferramentas de CAD para componentes mecânicos. A SysML torna, assim, mais fácil a utilização de modelos coesos e consistentes na construção de sistemas e condução de projetos de engenharia (FRIEDENTHAL; MOORE; STEINER, 2012)

A linguagem SysML se originou da UML - *Unified Modeling Language* - sendo a primeira um estereótipo da segunda. No início dos anos 1980, linguagens gráficas de modelagem foram criadas para ajudar desenvolvedores de software a definirem e comunicarem a estrutura e comportamento de seus programas através de diagramas padronizados. Vários esforços foram feitos por Booch, Rumbaugh e Jakobsson para definir notações OO para esses diagramas, e seus trabalhos foram eventualmente combinados em 1997 para formar a UML (PEARCE; HAUSE, 2012).

Em meados de 2000, já era evidente para a comunidade global de engenheiros de sistema que faltava uma linguagem padronizada, independente de domínios de aplicação, para a modelagem de sistemas complexos de engenharia. Neste mesmo período,

o método OOSEM (ver apêndice B) foi desenvolvido com o intuito de equipar engenheiros de sistema com técnicas OO e de modelagem (LYKINS; FRIEDENTHAL; MEILICH, 2000). Até então, a UML já havia demonstrado sua utilidade em suportar tanto os processos de ciclo de vida de sistemas quanto conceitos abstratos, ambos já familiares para os engenheiros de sistemas. Isto levou o *International Council on Systems Engineering* - INCOSE e o *Object Management Group* - OMG a desenvolver uma extensão da linguagem UML para a modelagem de sistemas. Em 2007 a OMG lançou a primeira especificação da linguagem de modelagem de sistemas (OMG SysML™), tomando emprestado, e estendendo, muitos dos conceitos de OO, elementos, relações e diagramas encontrados na UML.

Figura 80: Os quatro pilares da linguagem OMG SysML™

APÊNDICE B - O MÉTODO OOSEM

A aplicação da MBSE não implica na adoção de uma única metodologia específica, e várias já foram propostas para esse fim (DICKERSON; MAVRIS, 2013; ESTEFAN, 2008). Dentre elas, o Método da Engenharia de Sistemas Orientada a Objeto - OOSEM, que dá título a essa seção, se destaca no âmbito da engenharia de computação por ter como uma de suas características conseguir tornar mais simples a integração entre softwares desenvolvidos segundo o paradigma da orientação a objetos, aos demais componentes de hardware do sistema, bem como os respectivos testes. O método OOSEM é baseado em modelos e pode ser implementado com o uso da SysML, sendo, assim, suportado por qualquer ferramenta *case* que trabalhe com essa linguagem.

O método OOSEM é constituído das seguintes atividades em seu nível mais elevado, sendo essas, por sua vez, constituídas de diversas sub-atividades, algumas delas exploradas no capítulo 3 deste trabalho:

- 1 Organização do modelo;
- 2 Análise das necessidades do cliente;
- 3 Análise dos requisitos de sistema;
- 4 Gerenciamento da rastreabilidade dos requisitos;
- 5 Avaliação de soluções alternativas;
- 6 Definição da arquitetura lógica do sistema;

7 Sintetizar posibilidades de arquitecturas físicas;

FIGURE 17.2

OOSEM *Specify and Design System* process. The action numbers refer to subsection where the action is described.

Figura 81: Fluxograma principal do método OOSEM (FRIEDENTHAL; MOORE; STEINER, 2012).

APÊNDICE C - A FERRAMENTA *CASE* MAGICDRAW

O software MagicDrawTM é uma ferramenta de modelagem produzida e mantida pela empresa No Magic Inc, situada em Allen, Texas - USA, e que possui como público-alvo analistas de negócio, de software, programadores, engenheiros, gerentes de projeto, dentre outros. Seu objetivo principal, assim como de seus concorrentes, é ser uma ferramenta *case* - *computer aided systems engineering* - que possa facilitar a análise e o design de sistemas OO, sejam eles aplicações de software, bancos de dados ou até mesmo sistemas de engenharia completos. A ferramenta permite trabalhar com os principais padrões de linguagens gráficas de modelagem, como a UML, a SysML e o BPMN.

Ferramentas *case* são importantes recursos para se produzir softwares e sistemas de alta qualidade, com baixo índice de falhas. Durante as décadas iniciais do design de software, os profissionais da área cunharam esse termo para discutirem sobre possibilidade de se ter programas de computador que ajudassem os desenvolvedores a criar novos programas e sistemas.

A ideia essencial por trás de uma ferramenta *case*, de maneira análoga a ferramentas CAD para sistemas essencialmente mecânicos, é que quando modelos de sistemas são concebidos antes do início da construção dos mesmos, suas análises são facilitadas e o resultado final de seus projetos, geralmente, são de mais qualidade.