



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA

Gabriel da Silva Sezerino

**Sistema especialista para configuração de motores elétricos: uma aplicação de
programação por restrições**

Florianópolis
2022

Gabriel da Silva Sezerino

Sistema especialista para configuração de motores elétricos: uma aplicação de programação por restrições

Dissertação submetida ao Programa de Pós-Graduação em Engenharia Mecânica da Universidade Federal de Santa Catarina para a obtenção do título de Mestre em Engenharia Mecânica
Orientador: Prof. Jonny Carlos da Silva, Dr. Eng.

Florianópolis

2022

Ficha de identificação da obra

Sezerino, Gabriel da Silva

Sistema especialista para configuração de motores elétricos : uma aplicação de programação por restrições / Gabriel da Silva Sezerino ; orientador, Jonny Carlos da Silva, 2022.

160 p.

Dissertação (mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós-Graduação em Engenharia Mecânica, Florianópolis, 2022.

Inclui referências.

1. Engenharia Mecânica. 2. Sistemas especialistas. 3. Configuração de produto. 4. Programação por restrições. 5. Projeto mecânico. I. Silva, Jonny Carlos da. II. Universidade Federal de Santa Catarina. Programa de Pós Graduação em Engenharia Mecânica. III. Título.

Gabriel da Silva Sezerino

Sistema especialista para configuração de motores elétricos: uma aplicação de programação por restrições

O presente trabalho em nível de mestrado foi avaliado e aprovado por banca examinadora composta pelos seguintes membros:

Prof. Rodrigo Bastos Fernandes, Dr. Eng.
Universidade Federal de Santa Catarina

Prof. Ricardo José Rabelo, Dr. Eng.
Universidade Federal de Santa Catarina

Certificamos que esta é a **versão original e final** do trabalho de conclusão que foi julgado adequado para obtenção do título de Mestre em Engenharia Mecânica.

Prof. Henrique Simas, Dr. Eng.
Coordenação do Programa de Pós-Graduação

Prof. Jonny Carlos da Silva, Dr. Eng.
Orientador

Florianópolis

2022

Este trabalho é dedicado à minha família.

AGRADECIMENTOS

À Universidade Federal de Santa Catarina.

Ao Professor Jonny Carlos da Silva, por ter aceitado ser meu orientador e ter prestado todo apoio e orientação necessária para o desenvolvimento do trabalho.

À WEG, por ter disponibilizado tempo, recursos e especialistas para que esse trabalho pudesse ser realizado.

Aos meus pais, pelo dom da vida e pela educação passada desde o ensino infantil até a pós-graduação.

À minha noiva Laíra, pela compreensão e incentivo.

“Any customer can have a car painted any colour that he wants, so long as it is black”. (FORD, Henry, 1922)

RESUMO

Algumas companhias encontraram na customização em massa uma fonte de diferenciação em relação aos concorrentes. Esse paradigma de produção é auxiliado pela adoção de sistemas de configuração de produto, que são sistemas especialistas que transcrevem requisitos de clientes em especificações válidas dentro do espaço de produtos da companhia. A fim de suportar a customização em massa, a literatura recomenda que sejam aplicados conceitos de modularização ou seja, estabelecendo uma arquitetura modular de produtos – com a definição de uma estrutura hierárquica que representa uma linha de produtos. Outra recomendação é a utilização de um sistema de configuração de produto, que pode ser utilizado em um processo de vendas de produtos customizados em massa. Existem configuradores de primeira geração (limitados por regras unidirecionais e sem uma estrutura de objetos adequada), segunda geração (com recursos de regras bidirecionais e estrutura de objetos) e terceira geração (espaço de busca pré-compilado). O recurso de regras bidirecionais pode ser modelado dentro do paradigma de programação por restrições, constituindo numa ferramenta poderosa de solução de variados tipos de problemas. No presente trabalho, estabeleceu-se um método de implantação de um sistema especialista de configuração de produtos, desenvolvendo um protótipo aplicado no subsistema mecânico de um motor elétrico, tendo utilizado uma multinacional fabricante de motores elétricos como estudo de caso. Essa multinacional já possuía um sistema de configuração de primeira geração e uma *shell* apta a programação por restrições. No desenvolvimento do protótipo que se caracterizou como um sistema de configuração de segunda geração, percorreram-se as etapas de construção de um sistema especialista, seguindo o método proposto e convertendo dados e regras dos sistemas legados. Nesse processo, abriram-se oportunidades para integrar conhecimentos de especialistas de vendas, engenharia e manufatura através da codificação de novas restrições à configuração do produto. Também se integrou no configurador a utilização de atributos para controle de processos internos da empresa-alvo. Comprovaram-se os ganhos esperados com as ferramentas aplicadas, validando os resultados através da análise comparativa com a situação atual na empresa-alvo e também coletando a opinião de um especialista no domínio de configuração. Por fim, destacaram-se aspectos específicos do domínio mecânico de configuração e apresentaram-se sugestões para trabalhos futuros, que envolvem a expansão para outros domínios do conhecimento, integrações com sistemas de cálculo e sistemas ERP, além da utilização de outras tecnologias como configuradores de produto de terceira geração e identificação de padrões.

Palavras-chave: Sistemas especialistas, configuração de produto, programação por restrições, projeto mecânico.

ABSTRACT

Some companies have found in mass customization a source of differentiation from competitors. This production paradigm is aided by the adoption of product configuration systems, which are expert system that translate customer requirements into valid specifications within the company's product space. In order to support mass customization, the literature recommends that modularization concepts be applied, establishing modular product architecture – with the definition of a hierarchical structure that represents a range of products. Another recommendation is the use of a product configuration system, which can be used in a sales process for mass customized products. There are first generation configurators (limited by unidirectional rules and without a proper object structure), second generation (with bidirectional rules and object structure) and third generation (precompiled search space) configurators. Bidirectional rules feature can be modeled within the constraint programming paradigm, consisting in a powerful tool for solving several types of problems. In the current work, a method for the implementation of a expert system for product configuration was established, developing a prototype applied to the mechanical subsystem of an electric motor, using a multinational electric motors manufacturer as a case study. This multinational already had a first-generation configuration system and a shell capable of programming with constraints. In the development of the prototype, which was characterized as a second-generation configuration system, the developing stages for an expert system were covered, following the proposed method and converting legacy system data and rules. In the process, opportunities were opened up to integrate expert knowledge from sales, engineering and manufacturing by coding new constraints into the product configurator. The use of attributes to control internal processes of the target company was also integrated into the configurator. The expected gains with the applied tools were verified, by validating the results through comparative analysis with the current situation in the target company and also by collecting the opinion of an expert in the configuration domain. Finally, specific aspects of the mechanical domain were highlighted and suggestions for future works were presented, which involve the expansion to other domains of knowledge, integration with calculation systems and ERP systems, as well as the use of other technologies such as third generation product configurators and pattern identification.

Keywords: Expert systems, product configuration, constraint programming, mechanical project.

LISTA DE FIGURAS

Figura 1.1 – Mudança nos paradigmas de produção	16
Figura 1.2 – Fluxo de venda customizada	18
Figura 2.1 – Mecanismos de estruturação de famílias de produto	26
Figura 2.2 – Tipos de modularidade.....	27
Figura 2.3 – Diagrama UML da estrutura de dados em sistemas PLM	29
Figura 2.4 - Como optar entre um sistema aberto ou fechado	34
Figura 2.5 – Engenharia sequencial e simultânea	44
Figura 2.6 – Representação gráfica do modelo PRODIP	45
Figura 2.7 – Processo vendas-entrega com produtos configuráveis	49
Figura 3.1 – Estrutura de um sistema especialista.....	55
Figura 3.2 – Exemplo de rede antes de aplicar a consistência em arco (esquerda) e após (direita).....	60
Figura 4.1 – Motor elétrico e seus componentes.....	70
Figura 4.2 – Fluxo de processamento de pedidos na empresa-alvo	72
Figura 4.3 – Total de projetos vendidos em 2021 (à esquerda) e tipos de customizações nos projetos customizados (à direita)	74
Figura 4.4 – Diferença de altura entre motores com formas construtivas distintas ...	75
Figura 4.5 – Diferentes formas de fixação de um motor elétrico.....	75
Figura 4.6 – Motor vertical com flange e disco de proteção	76
Figura 4.7 – Sistema iMech para cálculo de ponta de eixo e rolamentos	77
Figura 4.8 – Algumas formas construtivas de um motor elétrico	78
Figura 4.9 – Alteração da posição da caixa de ligação	79
Figura 4.10 – Visão da caixa de ligação principal (ao centro) complementada com caixa auxiliar (à esquerda) e caixa de resistências (à direita).....	80
Figura 4.11 – Exemplo de pedido de cotação	81
Figura 4.12 – Tela inicial do configurador de produto com as principais divisões ...	82
Figura 4.13 – Atributo com um <i>combo box</i>	83
Figura 4.14 – Condição SE da regra de dependência 6217056.....	84
Figura 4.15 – Condição ENTÃO da regra de dependência 6217056.....	84
Figura 4.16 – Exemplo de regra de consistência.....	85
Figura 4.17 – Atributos iniciais do configurador de produto	86

Figura 4.18 – Estrutura de nível único (à esq.) x estrutura multinível (à direita)	89
Figura 4.19 – Processo de filtro de valores de atributo	90
Figura 4.20 – Fluxo proposto de funcionamento do novo sistema	92
Figura 4.21 – Estrutura geral de módulos da parte mecânica do motor elétrico	94
Figura 4.22 – Processo expandido de filtro de valores de atributos.....	96
Figura 4.23 – Detalhe da montagem ventilador - eixo	97
Figura 4.24 – Motor com saída dos cabos para baixo (à esquerda) e saída dos cabos para ponta do eixo (à direita).....	98
Figura 4.25 – Visão de usuário nível básico (à esquerda) e visão de usuário nível avançado (à direita)	100
Figura 4.26 - Fluxo proposto de funcionamento do novo sistema com resolução de conflitos	103
Figura 4.27 – Exemplo de preenchimento de uma restrição tabular com variantes de um componente (ventilador).....	106
Figura 5.1 – Visão geral da tela de configuração	110
Figura 5.2 – Árvore completa apresentada no sistema-protótipo.....	111
Figura 5.3 – Tela inicial após seleção do tamanho da carcaça.....	112
Figura 5.4 – Parte da lista técnica extraída do ERP de um produto	113
Figura 5.5 – Tela do nó eixo após seleção do tamanho da carcaça.....	114
Figura 5.6 – Recorte da tabela de restrições intra-componente para a caixa de ligação	119
Figura 5.7 – Motor com caixa de ligação com 3 furos na caixa de ligação	120
Figura 5.8 – Seleção inicial para o primeiro estudo de caso	120
Figura 5.9 – Opções de furos para a posição A2 da caixa de ligação	121
Figura 5.10 – Opções de furos para a posição A3 da caixa de ligação	122
Figura 5.11 – Tela indicando a existência de um conflito na configuração	122
Figura 5.12 – Tela de resolução de conflitos indicando a restrição que deve ser desligada.....	123
Figura 5.13 – Resultado após solução do conflito no nó “configurador simplificado”	123
Figura 5.14 – Resultado após solução do conflito no nó “caixa de ligação”	124
Figura 5.15 – Cadastro da caixa com três furos	125
Figura 5.16 – Nó caixa de ligação usinada com item 17000000 selecionado.....	126

Figura 5.17 – Caixa de ligação com três furos	127
Figura 5.18 – Nó caixa de ligação com configuração impossível.....	128
Figura 5.19 – Fluxo de processamento proposto de pedidos na empresa-alvo	129
Figura A.1 – Declaração de uma classe representando o componente motor.....	148
Figura A.2 – Declaração de uma porta para o conjunto da carcaça na classe do motor	148
Figura A.3 – Declaração de algumas propriedades na classe do ventilador	149
Figura A.4 – Aplicação Attributes administrando o atributo “Design”	150
Figura A.5 – Exemplo de codificação de uma restrição tabular	151
Figura A.6 – Restrição de montagem ventilador – eixo.....	151
Figura A.7 – Exemplo de recomendação na shell CBConf.....	152
Figura A.8 – Regra desnecessária no novo sistema	153
Figura A.9 – Primeira restrição para o primeiro estudo de caso	154
Figura A.10 – Segunda restrição para o primeiro estudo de caso	154

LISTA DE TABELAS

Tabela 2.1 – Tipos de configuradores	41
Tabela 5.1 - Amostra de itens reais dos produtos versus itens selecionados pelo protótipo	115
Tabela 5.2 – Tamanhos de rosca para caixa de ligação para o mercado brasileiro..	119
Tabela 5.3 – Avaliação do especialista	131
Tabela A.1 – Itens reais dos produtos versus itens selecionados pelo protótipo.....	153

LISTA DE ABREVIATURAS E SIGLAS

ATO - Assembly-to-order

BOM - Bill-of-materials

CAD - Computer aided design

CLP - Controlador lógico-programável

CTO - Configure-to-order

DFMC - Design for mass customization

E-BOM – Engineering bill-of-materials

ERP – Enterprise resource planning

ETO - Engineering-to-order

GM – General Motors

IA – Inteligência Artificial

M-BOM – Manufacturing bill-of-materials

MFD - Modular function deployment

MTS - Make-to-stock

MVP – Minimum viable product

NeDIP – Núcleo de Desenvolvimento Integrado de Produtos

PLM - Product lifecycle management

PRODIP – Processo de Desenvolvimento Integrado de Produtos

PTO - Pick-to-order

SCP - Sistema de configuração de produtos

STO - Select-to-order

UML - Unified modeling language

XOR - Exclusive OR

SUMÁRIO

1	INTRODUÇÃO	15
1.1	Justificativa	17
1.1.1	Relevância do tema	19
1.2	OBJETIVOS	20
1.2.1	Objetivo Geral.....	20
1.2.2	Objetivos Específicos	20
1.3	LIMITAÇÃO DO TEMA.....	21
1.4	ORGANIZAÇÃO DA DISSERTAÇÃO.....	21
2	CUSTOMIZAÇÃO EM MASSA	23
2.1	MODULARIDADE.....	23
2.1.1	Estrutura modular de produto	24
2.1.2	Histórico da modularidade	25
2.1.3	Módulo	26
2.1.4	Tipos de modularidade.....	26
2.1.5	Elementos-chave da modularidade	29
2.1.5.1	<i>Arquitetura de produto</i>	29
2.1.5.2	<i>Conjunto</i>	30
2.1.5.3	<i>Sistema e subsistema.....</i>	30
2.1.5.4	<i>Componente e componente-chave</i>	31
2.1.5.5	<i>Interface.....</i>	31
2.1.5.6	<i>Atributos</i>	31
2.1.5.7	<i>Portas.....</i>	32
2.1.5.8	<i>Restrições.....</i>	32
2.1.5.9	<i>BOM, E-BOM E M-BOM</i>	32
2.1.6	Sistemas abertos e fechados	33

2.2	DEFINIÇÃO DE CUSTOMIZAÇÃO EM MASSA	35
2.2.1	Benefícios	36
2.2.2	Desafios e pontos de atenção	36
2.2.3	Como resolver os problemas da customização em massa	37
2.2.4	<i>Design for Mass Customization – DFMC</i>	38
2.2.5	O papel dos configuradores de produto na customização em massa	39
2.2.6	Estratégia de conexão vendas-manufatura	40
2.3	PROCESSOS ENVOLVENDO CUSTOMIZAÇÃO EM MASSA.....	42
2.3.1	Processo de desenvolvimento de novos produtos	42
2.3.2	Engenharia simultânea	43
2.3.3	PRODIP – Processo de desenvolvimento integrado de produtos	44
2.3.3.1	<i>Planejamento do projeto</i>	46
2.3.3.2	<i>Elaboração do projeto do produto</i>	46
2.3.3.3	<i>Implementação do lote piloto</i>	47
2.3.4	Processo de desenvolvimento de produtos para customização em massa	48
2.3.5	Processo de vendas de produtos sob customização em massa	49
3	SISTEMAS ESPECIALISTAS PARA CONFIGURAÇÃO DE PRODUTO	51
3.1	INTELIGÊNCIA ARTIFICIAL	51
3.2	SISTEMAS ESPECIALISTAS	55
3.2.1	Processo de construção de um sistema especialista	56
3.2.2	Ferramentas para construção de um sistema especialista	57
3.2.3	Aplicação de um sistema especialista	58
3.3	Programação por restrições.....	59
3.3.1	Consistência em arco	60
3.3.2	Sintaxe de restrições	61
3.3.3	Restrições <i>soft</i>	62
3.4	Sistema de configuração de produto.....	62

3.4.1	As três gerações de Sistemas de Configuração de Produto.....	63
3.4.2	Capacidades de um configurador de produto.....	64
3.4.3	Relação dos configuradores de produto com a inteligência artificial e sistemas especialistas	66
4	DESENVOLVIMENTO DO PROTÓTIPO DO SISTEMA DE CONFIGURAÇÃO DE MOTORES ELÉTRICOS	68
4.1	CONSTRUÇÃO DE UM SISTEMA ESPECIALISTA DE CONFIGURAÇÃO.	68
4.2	IDENTIFICAÇÃO DO PROBLEMA	70
4.2.1	Produto-alvo	70
4.2.2	Uma introdução a motores elétricos	70
4.2.3	Estratégia de conexão vendas-manufatura	71
4.2.4	Fontes de variação no produto	73
4.2.5	<i>Design for Mass Customization (DFMC) no produto-alvo</i>	<i>78</i>
4.2.6	Funcionamento do configurador de produto no fluxo atual.....	80
4.3	Conceitualização.....	87
4.3.1	Conceito geral.....	87
4.3.2	Arquitetura do sistema de configuração	89
4.4	Formulação	93
4.4.1	Estrutura do produto	93
4.4.2	Definição dos valores disponíveis na configuração do produto.....	95
4.4.3	Definição dos valores padrões na configuração do produto	98
4.4.4	Verificação de inconsistências na configuração e mecanismo de explicações	99
4.5	Implementação.....	100
4.5.1	<i>Shell</i> CBCConf.....	100
4.5.1.1	<i>Algoritmo de resolução de conflitos</i>	<i>102</i>
4.5.2	Implementação da estrutura do produto.....	104
4.5.3	Implementação dos domínios.....	105

4.5.4	Implementação dos valores disponíveis	105
4.5.5	Implementação dos valores padrões	107
4.5.6	Implementação das consistências e mecanismo de explicações	107
5	VERIFICAÇÃO E AVALIAÇÃO DO SISTEMA	109
5.1	INTERFACE E UTILIZAÇÃO DO SISTEMA-PROTÓTIPO	109
5.2	VERIFICAÇÃO DAS RESTRIÇÕES	111
5.3	NÚMEROS DO SISTEMA-PROTÓTIPO	115
5.4	Resultados da verificação	117
5.4.1	Estudo de caso 1: Acrescentar nova variante de módulo ao sistema-protótipo	118
5.4.2	Estudo de caso 2: Restringir condição impossível	126
5.4.3	Alteração no processo de consulta técnica na empresa-alvo	128
5.5	Validação do sistema-protótipo	129
6	CONCLUSÕES.....	134
6.1	Sugestão para trabalhos futuros	135
	REFERÊNCIAS.....	138
	APÊNDICE A – Detalhamento das implementações	148
	APÊNDICE A1 – DETALHES DE IMPLEMENTAÇÃO DA ESTRUTURA DE MÓDULOS.....	148
	APÊNDICE A2 – DETALHES DE IMPLEMENTAÇÃO DOS DOMÍNIOS	149
	APÊNDICE A3 – DETALHES DE IMPLEMENTAÇÃO DOS VALORES DISPONÍVEIS	150
	APÊNDICE A4 – DETALHES DE IMPLEMENTAÇÃO DOS VALORES PADRÕES	152
	APÊNDICE A5 – ITENS SELECIONADOS PELO CONFIGURADOR DE PRODUTO	152
	APÊNDICE A6 – EXEMPLO DE REGRA DESNECESSÁRIA NO SISTEMA PROPOSTO	153

APÊNDICE A7 – DETALHAMENTO DAS RESTRIÇÕES PARA PRIMEIRO ESTUDO DE CASO	154
---	------------

1 INTRODUÇÃO

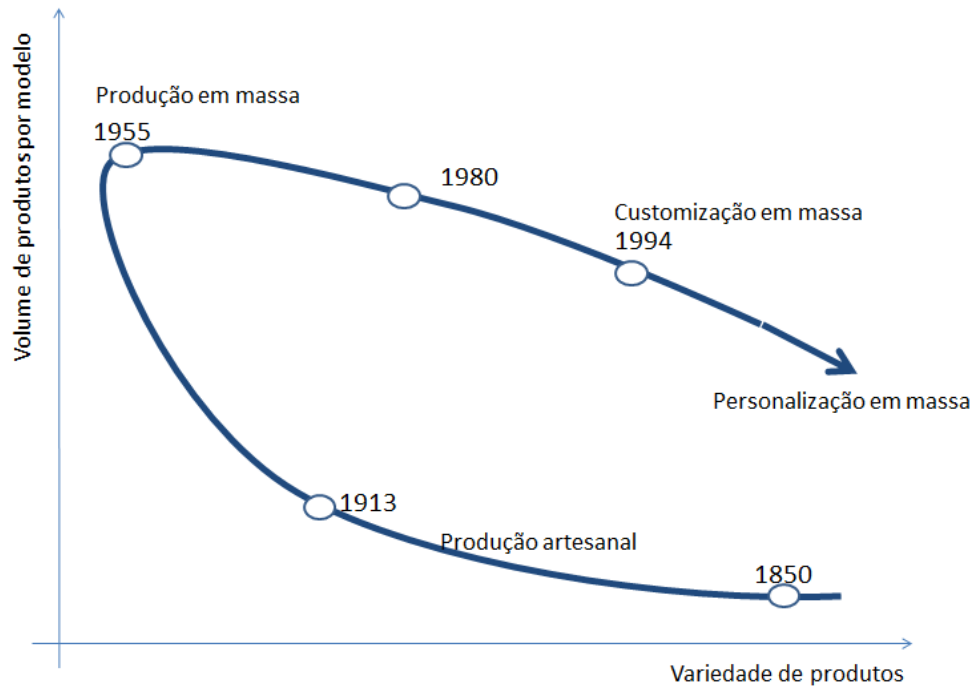
Encontrar uma fonte de diferenciação sobre concorrentes é essencial em um mundo competitivo para empresas de diversos segmentos. Nesse contexto, algumas companhias encontraram no paradigma de customização em massa (SHAO, 2020) – caracterizado pela produção de elevadas quantidades de produtos ajustados exatamente às necessidades dos clientes com eficiência próxima à produção em massa tradicional (CAO et al., 2021) – uma forma de diferenciar-se da concorrência e alcançar uma significativa fatia do mercado nos negócios em que atua.

A customização em massa frequentemente é suportada pela adoção de configuradores de produto, que são sistemas especialistas, focados em transcrever os requisitos dos clientes em especificações válidas dentro do espaço de produtos da companhia (TRENTIN; PERIN; FORZA, 2013). Essa categoria de sistema permite guiar o cliente de forma mais eficiente por uma gama de produtos que pode ser bastante vasta, além de prevenir que produtos impossíveis de serem fabricados sejam vendidos. Ao endereçar essas questões, a empresa pode aumentar as margens e/ou volume de vendas, devido às possibilidades de customização que oferece em relação aos concorrentes (TRENTIN; PERIN; FORZA, 2013).

No paradigma de customização, a estratégia de produção adotada é a *Configure-To-Order* (CTO), na qual os produtos só são fabricados e montados mediante venda, eliminando estoques de itens finais (NYAGA et al. 2007), em oposição à estratégia *Make-To-Stock* (MTS) – em que há formação de estoques, e diferente da estratégia *Make-To-Order*, onde não há o processo de configuração. Estratégias intermediárias como a *Assembly-To-Order* (ATO) também são possíveis conforme a mesma referência.

No início do século XIX, a estratégia predominante era *Make-To-Order* (MTO), e a produção dos mais variados bens era artesanal. A estratégia MTS só passou a ser largamente adotada com o advento da produção em massa, que ocorreu no início do século XX, conforme explora a Figura 1.1. Segundo Mladineo et al. (2021), os produtos caminham para uma personalização cada vez maior.

Figura 1.1 – Mudança nos paradigmas de produção



Fonte: Adaptada de Mladineo et al. (2021)

Muito se discute também com a ascensão da filosofia de manufatura enxuta acerca da redução de estoques finais e eliminar produção sem venda. Nesse caso, é possível observar como os modos de produção CTO, MTO e ATO concordam, ao menos parcialmente, com os preceitos de produção enxuta (LIKER, 2004), pois há uma efetiva redução de estoque, i.e os produtos só são efetivamente produzidos quando requisitados.

Alguns pontos de atenção surgem ao adotar a customização em massa como paradigma dominante em uma companhia. Esses pontos têm sido intensamente discutidos na literatura, como em Zipkin (2001), Baker et al. (2007) e Sloan (2014).

Em primeiro lugar, a grande quantidade de variações pode afetar o desempenho da manufatura e ameaçar a sustentabilidade do modelo de negócio. Os lotes de componentes inevitavelmente se tornarão menores, exigindo layouts de produção diferentes dos layouts de produção em larga escala, como a produção em linha. Por este motivo, boa parte do processo produtivo precisa recorrer a layouts funcionais e processos produtivos modulares, que permitem uma boa relação de compromisso entre produtividade e flexibilidade (WEG, 2021).

Em segundo lugar, o prazo de entrega necessariamente é superior ao prazo das empresas que adotam a produção em massa, pois no regime MTS seus produtos estarão estocados e disponíveis para entrega imediata. As empresas MTS podem ainda valer-se de

técnicas preditivas de demanda, que tem alcançado um novo patamar com a adoção de inteligência artificial para encontrar e automatizar a busca por padrões ocultos nas vendas (BANERJEE et al., 2021). Em uma empresa que utiliza o modo CTO, devido à elevada quantidade de produtos possíveis de configurar, provavelmente não será factível alcançar uma precisão aceitável seguindo tal abordagem. Considerando ainda que os consumidores têm-se tornado mais exigentes com prazos, os prazos de entrega no processo CTO devem ser os mais exíguos possíveis, pois em alguns casos há preferência pelo tempo de entrega ao invés do preço para escolha de produtos (MATERIAL HANDLING & LOGISTICS, 2016).

Em terceiro lugar, a customização em massa, por definição, leva a uma enorme quantidade de produtos diferentes na base de dados, causando uma dificuldade adicional no gerenciamento dessas informações. Além disso, uma categorização imprecisa ou insuficiente desses dados impossibilitará encontrar projetos idênticos ou similares – com o objetivo de sugeri-los a clientes com requisitos idênticos ou similares – fazendo com que a base de dados cresça ainda mais. Esse volume de dados também provoca lentidão no processamento de aplicações computacionais e elevado custo administrativo.

1.1 JUSTIFICATIVA

A presente proposta de trabalho se fundamenta na necessidade de evolução dos processos internos das empresas, a fim de encontrar uma fonte de diferenciação em relação aos concorrentes (SHAO, 2020). Para tanto, será desenvolvido um protótipo aplicado em um caso real de uma companhia.

A proposta do presente trabalho é desenvolvida em uma multinacional brasileira, a WEG Equipamentos Elétricos S.A. A empresa será doravante denominada empresa-alvo, envolvendo a subdivisão que desenvolve, fabrica e comercializa motores elétricos sob o paradigma de customização em massa. A empresa-alvo intenta desde os seus primórdios a customização de produtos como solução para os clientes. Evidentemente, esse propósito compeliu a empresa-alvo a buscar tecnologias que ajudassem a estruturar sua área de Engenharia. Como consequência, a empresa-alvo construiu uma vasta experiência, tanto na utilização de configuradores de produto, quanto na construção de sistemas especialistas como nos trabalhos de Massirer (2007) e Haselein (2018), e vêm ao longo do tempo buscando customizar soluções devido à falta de ferramentas prontas de mercado que atendam às necessidades internas. Este trabalho, portanto, utiliza as experiências com trabalhos anteriores

também no contexto da empresa-alvo, como de Massirer (2007), Vegini (2018) e Haselein (2018), que construíram sistemas especialistas para lidar com um problema específico de engenharia.

Diversas áreas internas à empresa-alvo são severamente afetadas pela obsolescência das tecnologias utilizadas nos sistemas legados de configuração e estruturação dos produtos. Por exemplo, a área de Engenharia de Produto é responsável pelo controle de todas as regras de configuração, mesmo as que não são de seu domínio de conhecimento, e essa concentração da administração ocorre devido às ferramentas atuais serem pouco amigáveis aos administradores. Os sistemas atuais também demandam uma elevada carga administrativa. Por exemplo, uma das diversas tabelas que compõe a base de conhecimento do sistema de configuração de produto possui mais de 1 milhão de linhas, tanto devido à grande variedade de produtos, quanto devido à forma como as regras estão estruturadas, e é atualizada com uma frequência elevada, conforme fontes internas da empresa-alvo. O sistema de estruturação de produto, entre regras de estruturação e regras de validação, possui mais de 10 mil regras codificadas, sendo que se criam em média 1000 novas regras anualmente, devido a constantes alterações nos produtos e inclusão de novas possibilidades.

Observando os sistemas de configuração e estruturação de produto, há um problema inconciliável na arquitetura atual. Considerando a Figura 1.2, tem-se o fluxo de venda customizada considerando os dois sistemas principais, que parte dos requisitos do cliente até chegar ao produto final. Nem sempre é possível que exista a regra R1 no sistema de configuração de produto, pois nessa etapa o nível de detalhamento do produto é baixo e as informações estão incompletas.

Figura 1.2 – Fluxo de venda customizada



Fonte: Adaptada de MLADINEO et al. (2021)

Por outro lado, se existir somente a regra R2 no sistema de estruturação de produto (que define todo o produto para a fabricação), pode haver retrabalho caso se descubra tardiamente nessa etapa que o produto configurado não pode ser fabricado. E, por fim, se a regra for codificada nos dois sistemas, incorre-se no problema de duplicidade de regras, o que exige constante vigilância para que futuras atualizações das regras sejam feitas nos dois sistemas de forma perfeitamente sincronizada – resultando em esforço duplicado. Considerando essa situação, reafirma-se a necessidade do presente trabalho, que se propõe a mitigar problemas desse tipo.

1.1.1 Relevância do tema

Tanto o conceito de customização em massa quanto à programação por restrições tem elevada relevância para além do estudo. A customização em massa é apontada por diversas referências como uma tendência e tem ganhado espaço em alguns segmentos. Empresas de porte multinacional como a Dell e Nike adotaram sistemas de customização (SLOAN, 2014).

A programação por restrições não é um paradigma exatamente novo, sendo inclusive altamente correlato aos sistemas especialistas clássicos. Por exemplo, na *shell* Clips (SILVA, 1998), se codifica de forma não-sequencial através da adição e retração de fatos. O surgimento de *frameworks* de programação facilitada como o Or-Tools (GOOGLE, 2020) e Choco-Solver (CHOCO-SOLVER, 2021), além da consolidação de ferramentas comerciais de configuração como Configit (CONFIGIT, 2020) e a solução da empresa Sparkling Logic, baseadas no Algoritmo de Rete (FORGY, 1982) comprova a maturidade desse paradigma.

Na empresa-alvo, os sistemas mostrados na Figura 1.2 e o sistema desenvolvido por Haselein (2018) utilizam o paradigma de regras sequenciais, enquanto que as soluções comerciais supracitadas utilizam regras declarativas, que operam em formato não-sequencial. Segundo Hendler, Kitano e Nebel (2006), o paradigma de programação por restrição é superior ao paradigma de regras sequenciais para a resolução de problemas de configuração.

Por fim, a consolidação das tecnologias de programação por restrição que objetivam suportar a configuração em massa atraiu a atenção de indústrias das mais diversas, como a Siemens, que utiliza programação por restrições há mais de 25 anos (FALKNER et al., 2016). Nesse ponto, ressalta-se que o reduzido know-how da empresa-alvo nesse paradigma impele a busca por mais conhecimento de um ponto de vista acadêmico.

1.2 OBJETIVOS

1.2.1 Objetivo Geral

O objetivo deste trabalho é estabelecer um método de implantação de um Sistema de Configuração de Produtos (SCP) baseado em restrições, voltado ao projeto mecânico de um motor elétrico, e validá-lo em estudos de caso aplicado junto à empresa-alvo. Este objetivo visa considerar a pergunta: como a programação por restrições pode contribuir no processo de organização do conhecimento sobre o projeto mecânico de motores elétricos?

Como mencionado, o presente trabalho constrói conhecimento a partir de experiências anteriores com sistemas especialistas aplicados na empresa-alvo advindos dos trabalhos de Massirer (2007) e Haselein (2018), buscando estabelecer um método modelo de implantação de um sistema de configuração, que não foi encontrado na literatura relevante do tema. Além disso, os estudos de caso exemplificam o aspecto de expansão da base de conhecimento, também pouco difundido na literatura.

A seguir, são apresentados os objetivos específicos que suportam o objetivo geral.

1.2.2 Objetivos Específicos

- i. Propor um método de implantação de um sistema especialista de configuração de produtos, aplicável a outros tipos de produto e em diferentes estágios de maturidade em uma companhia, a ser validados em um caso real.
- ii. Prototipar um configurador de produto da parte mecânica de um motor elétrico, a fim de validar os conceitos de modelagem.
- iii. Identificar quais as especificidades e desafios que o projeto mecânico de um motor elétrico traz ao processo e ao sistema especialista de configuração.
- iv. Realizar dois estudos de caso adotando um *shell* de programação para validar os conceitos discutidos no presente trabalho, sendo um estudo para permitir novas configurações de um produto e outro para restringir configurações de um produto.
- v. Avaliar a oportunidade do novo sistema para integrar demais melhorias e inovações encontradas na literatura, como, por exemplo, o conceito de programação por restrições.

- vi. Avaliar com os especialistas se o conhecimento encontra-se integralmente transcrito sob a forma de regras; em caso negativo, propor incorporar os aspectos faltantes na elaboração de um novo sistema.

1.3 LIMITAÇÃO DO TEMA

Os sistemas em estudo na empresa-alvo apresentam histórico de desenvolvimento de duas décadas, com diversos colaboradores trabalhando em tempo integral. A base de conhecimento é formada por uma imensa quantidade de regras e definições, não sendo, portanto, objetivo do presente trabalho implementá-los e/ou substituí-los por completo, devido ao esforço que tal tarefa representaria.

A proposta se limita, por conseguinte, a um estudo em menor escala, inicialmente restrito ao escopo do projeto mecânico e a um número reduzido de tipos de produto e regras. Considera-se que essa redução de escala não prejudica o estudo, conforme será apresentado nos capítulos 5 e 6, com a possibilidade de expansão do sistema para outros tipos de produto e regras.

1.4 ORGANIZAÇÃO DA DISSERTAÇÃO

O capítulo 1 é uma introdução geral ao trabalho, estabelecendo os objetivos e limitando o escopo. Trazem-se também as justificativas para o presente trabalho, que englobam problemas atuais e possíveis oportunidades.

O capítulo 2 trata de uma revisão bibliográfica dos conceitos sobre customização em massa, envolvendo aspectos como a modularidade e os processos envolvendo a customização em massa, como o processo de desenvolvimento de novos produtos e o processo de vendas de produto sob customização em massa.

O capítulo 3 trata de uma revisão bibliográfica sobre sistemas especialistas, inteligência artificial, programação por restrições e sistemas de configuração de produto, trazendo o relacionamento entre esses temas.

O capítulo 4 dispõe do desenvolvimento do protótipo do sistema de configuração de produto, objetivo principal do presente trabalho utilizando o método proposto de desenvolvimento.

O capítulo 5 apresenta a verificação e avaliação do protótipo desenvolvido, considerado aspectos como interface e alguns estudos de caso aplicados.

Por fim, o capítulo 6 apresenta as conclusões alcançadas com o trabalho, revisitando os objetivos e discutindo propostas de trabalhos futuros.

2 CUSTOMIZAÇÃO EM MASSA

No que diz respeito a conceitos relevantes para o presente trabalho, a customização em massa surge como um dos conceitos centrais. Uma empresa que adota a customização em massa como paradigma de produção de um ou vários produtos de seu portfólio, o faz através de sistemas de configuração de produto (MLADINEO et al., 2021). Nas próximas seções será explorado esse conceito de forma mais aprofundada, começando pela definição e benefícios, para posteriormente discutir os desafios, problemas e possíveis soluções. Uma das possíveis abordagens é o *Design for Mass Customization* – DFMC- assim como o uso de sistemas de configuração de produto – SCP. Ainda na esfera de soluções, boas práticas e preceitos, o conceito de modularidade – bem como suas definições, histórico, tipos, entre outras discussões – mereceu uma seção à parte, devido aos desdobramentos envolvidos.

2.1 MODULARIDADE

Viabilizar a customização em massa em uma companhia de forma organizada e sustentável requer a adoção de diversos preceitos simultaneamente, alguns já discutidos na seção anterior, como o DFMC. Nesta seção, discutir-se-ão conceitos que orbitam a modularidade, de elevada relevância nesse contexto.

A modularidade é uma das formas primárias de alcançar variabilidade, economia de escala e variedade de produtos, atendendo necessidades que mudam frequentemente. Duran et al. (2012) ressaltam a facilidade de variação dos produtos caso sejam construídos de forma modular de um ponto de vista de manufatura. Isto é, fica evidente que um dos pilares do sucesso da customização em massa pode justamente ser a estruturação modular de produtos, pois os objetivos são os mesmos: flexibilidade de requisitos e custos reduzidos.

Um trabalho relevante no tema Modularidade é *Modular Function Deployment* – MDF – (ERIXON 1998 apud LEHTONEN, 2007). O conceito de modularidade, dentro dessa teoria, é a decomposição de um produto em blocos construtivos com interfaces definidas, dirigidas por razões específicas da companhia. Sob esse aspecto, nota-se que existe certa liberdade na definição dos módulos, sendo algo que pode ser alterado em função das necessidades de modelagem do produto. Um motor diesel de um caminhão poderia ser representado como um módulo único ou uma estrutura mais completa com diversos módulos e sub-módulos, dependendo da necessidade.

2.1.1 Estrutura modular de produto

Para aplicar a modularidade, é necessário definir uma estrutura modular para os produtos da companhia. A estrutura modular de produto pode ser definida, de acordo com Bruun et al. (2014), como sendo uma estrutura com elementos autocontidos, com interfaces padronizadas e em acordo com uma definição de sistema. Um tipo desses elementos autocontidos são os módulos, que estão arranjados segundo uma estrutura hierárquica, onde um elemento pode conter o outro.

Tiihonen et al. (1998) também destacam a importância da estruturação do produto no contexto da customização em massa. Segundo eles,

A utilização de produtos configuráveis requer um processo sistemático de vendas-entrega e modelagem da família de produto como um produto configurável. Ao invés de definir explicitamente um conjunto de variantes de produto em uma família de produto, um produto configurável tem um modelo de configuração que contém toda a informação sobre as possibilidades de adaptar o produto às necessidades dos clientes...

Por produto configurável, entende-se o produto submetido à customização em massa em uma companhia e disponível em um sistema de configuração de produto; já o processo sistemático de vendas será discutido em maiores detalhes nas próximas seções. Seguindo Tiihonen et al. (1998), o produto configurável possui algumas propriedades de elevada importância na discussão, também em consonância com aspectos já percorridos no presente trabalho:

- i. O produto entregue ao cliente é feito sob medida para as necessidades do cliente
- ii. O produto foi pré-projetado para atender uma variedade de diferentes requisitos do cliente
- iii. O produto possui uma estrutura geral pré-projetada e formada por uma combinação de componentes também pré-projetados (ou módulos). Portanto, não haveria necessidade de novos componentes no processo normal de vendas-entrega. A exceção seriam sistemas abertos, como será exposto na seção 2.1.6.

Desenvolver produtos considerando uma estrutura modular, segundo Bruun et al. (2014), envolve focar o desenvolvimento não em produtos individuais, mas nas variantes de produto, reuso do conhecimento e utilização de economias de escala para atender clientes com

os mais diversos requisitos de forma eficiente, o que é equivalente ao que foi discutido sobre DFMC.

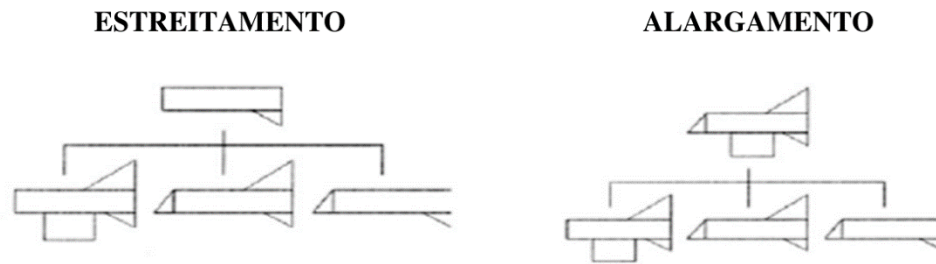
2.1.2 Histórico da modularidade

A tese de Timo Lehtonen (2007) – *Designing modular product architecture in the new product development* – é um extenso compilado a respeito do tema modularidade, permitindo observar uma evolução tanto tecnológica quanto de modelagem ao longo das últimas décadas. A teoria de Peter Van Mensch (1990 apud Lehtonen, 2007) é fundamentada em ciclos de vida e segmenta o produto em propriedades físicas, funcionais, de contexto e a representação do produto.

A teoria dos domínios possui quatro níveis (ANDREASEN 1980 apud LEHTONEN, 2007), sendo eles sistema de processos, sistema de efeitos, sistema de órgãos e sistema de peças, tendo sido utilizada em configuração sistemática de produtos, ainda sem necessariamente um sistema computacional de auxílio. Os primeiros exemplos de modularidade aplicados ao ciclo de vida de produto são dos anos 70, como por exemplo, a série de locomotivas Dash 2 da EMD (GM), onde a locomotiva inteira era modular e permitia reduzir tempos de parada pela substituição rápida de um módulo defeituoso. Similarmente, nos anos 80, a Scania S/A introduziu a linha de caminhões Série 2, que através de modularização, conseguiu atingir simultaneamente a redução da quantidade de peças e ferramentas e ainda aumentar a quantidade de variantes disponíveis.

Thomas Miller e Per Ergard (1999 apud LEHTONEN, 2007) propuseram dois mecanismos para estruturar famílias de produtos: estreitamento e alargamento. No princípio de estreitamento, devem ser buscados os requisitos mais comuns para formar produtos base, de modo que, partindo de uma variante, seja necessário “estretar” as capacidades para chegar a um produto base. Por outro lado, no princípio de alargamento, devem ser incluídos todos os requisitos das variantes no produto padrão, de modo que, partindo de uma variante, haja um “alargamento” das capacidades no produto base. Esses conceitos são explorados visualmente na Figura 2.1:

Figura 2.1 – Mecanismos de estruturação de famílias de produto



Fonte: Adaptada de Lehtonen (2007)

À esquerda na Figura 2.1, tem-se o mecanismo de estreitamento, no qual, a partir de um produto base, diferentes produtos são derivados acrescentando características (no exemplo da figura, outras figuras geométricas). Já na parte direita da imagem, observa-se o mecanismo de alargamento, onde o produto base é a “versão completa”, sendo necessário remover partes do mesmo para obter os diferentes produtos.

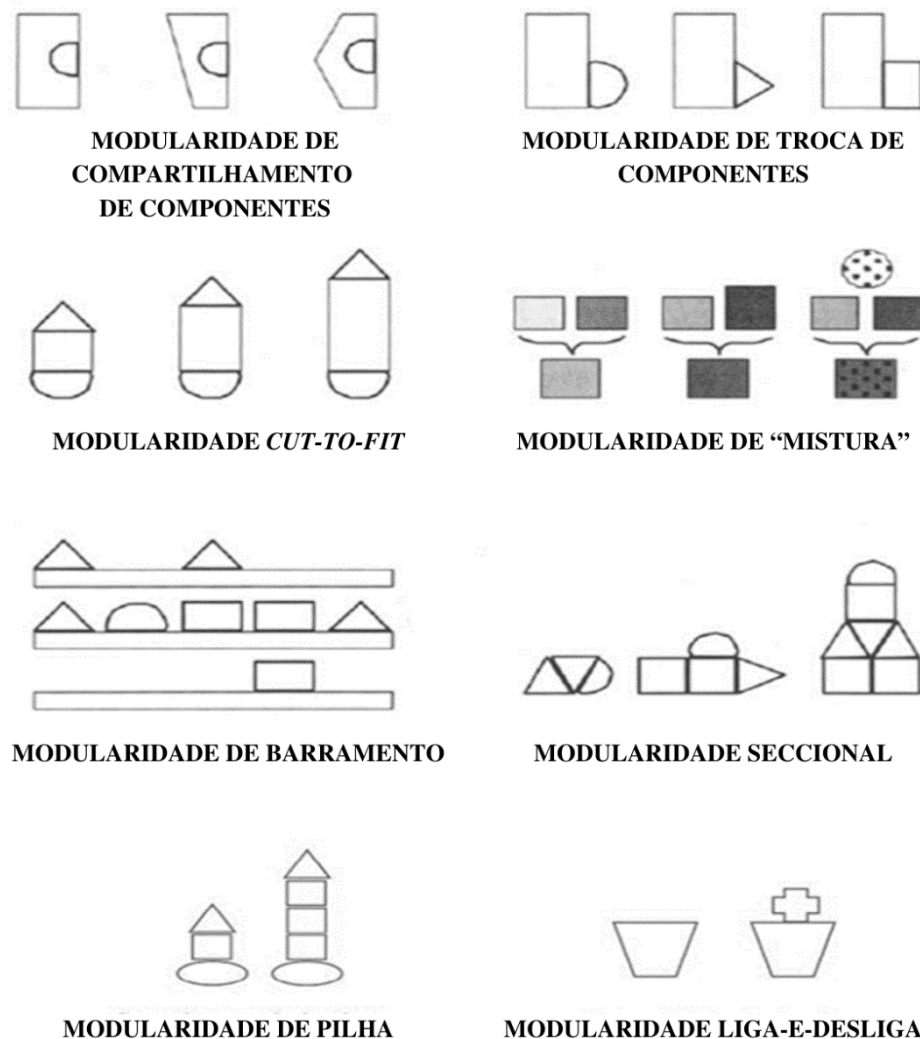
2.1.3 Módulo

Existem diversas definições equivalentes na literatura para módulo. Uma definição bastante aceita é a de Masahi ko Aoki (2003 apud LI et al., 2015), onde o módulo é “um subsistema parcialmente autocontido, com certas funções independentes que pode ser interconectado com outros subsistemas parcialmente autocontidos de acordo com certas restrições, formando um sistema mais complexo. É também uma entidade com interfaces fixas”. Um módulo pode ser, portanto, uma peça física inteira ou uma combinação de várias peças, desde que atenda aos requisitos citados pela literatura. Pakkanen (2021) também discute o conceito de variantes de módulo, que seriam as diferentes peças de um mesmo tipo disponíveis em uma família de produto.

2.1.4 Tipos de modularidade

A modularidade pode ser enquadrada em tipos. Esses tipos dizem respeito à forma como se obtém variações, relacionando os componentes e seus parâmetros a essas variações. Ulrich e Tung (1991 apud Lehtonen 2007) trazem uma proposta de classificação de seis tipos, posteriormente sendo complementada em mais dois tipos. Exemplos gráficos simplificados são mostrados na Figura 2.2:

Figura 2.2 – Tipos de modularidade



Fonte: Adaptada de Lehtonen (2007)

São elas:

i. Modularidade de compartilhamento de componentes

Trata-se de uma modularidade oriunda do compartilhamento de componentes entre diferentes produtos finais. Os diferentes produtos finais podem ser inclusive de diferentes famílias de produto.

ii. Modularidade de troca de componentes

Estabelecendo um componente central, esse tipo de modularidade é definido pela troca de componentes periféricos a esse componente central.

iii. Modularidade *cut-to-fit*

Um determinado componente que compõe o produto final pode ser cortado para se adequar às necessidades do cliente, mantendo os demais componentes intactos. Um exemplo

são produtos como calhas de iluminação, que podem ser cortados no comprimento adequado ao espaço existente, mas mantendo as ponteiros originais.

iv. Modularidade de “mistura”

Fundamenta-se na mistura de elementos, combinando-se para formar um novo elemento. Sistemas de cores para pintura podem ser vistas como modularidade de mistura. Na Figura 2.2, é mostrado um exemplo de blocos quadrados em escala de cinza que são combinados para formar novas cores, formatos e padrões de preenchimento.

v. Modularidade de barramento

A modularidade de barramento é típica de componentes elétricos como painéis. O barramento presente nos painéis permite inserir os mais variados componentes, como disjuntores, contadoras, CLP's, entre outros.

O elemento mais importante, nesse caso de modularidade, é a própria definição do barramento, suas dimensões e padrões de conexão, pois todos os componentes terão de ser complacentes a este padrão. O barramento tipicamente apresentará simetria axial, isto é, possui um perfil estendido ao longo do comprimento.

vi. Modularidade seccional

A modularidade seccional é baseada em interfaces padronizadas que permitem montagens livres. Um bom exemplo seria as peças de montagem LEGO®, que possuem um padrão de pino macho-fêmea e permite montagem de quaisquer peças que possuam este padrão em posições quaisquer.

vii. Modularidade de pilha

A modularidade de pilha é descrita como um caso especial da modularidade *cut-to-fit*, mas onde o comprimento é obtido multiplicando um determinado módulo base para alcançar o comprimento desejado. O exemplo descrito por Lehtonen (2007) é um ajuste de distância de uma peça utilizando placas distanciadoras.

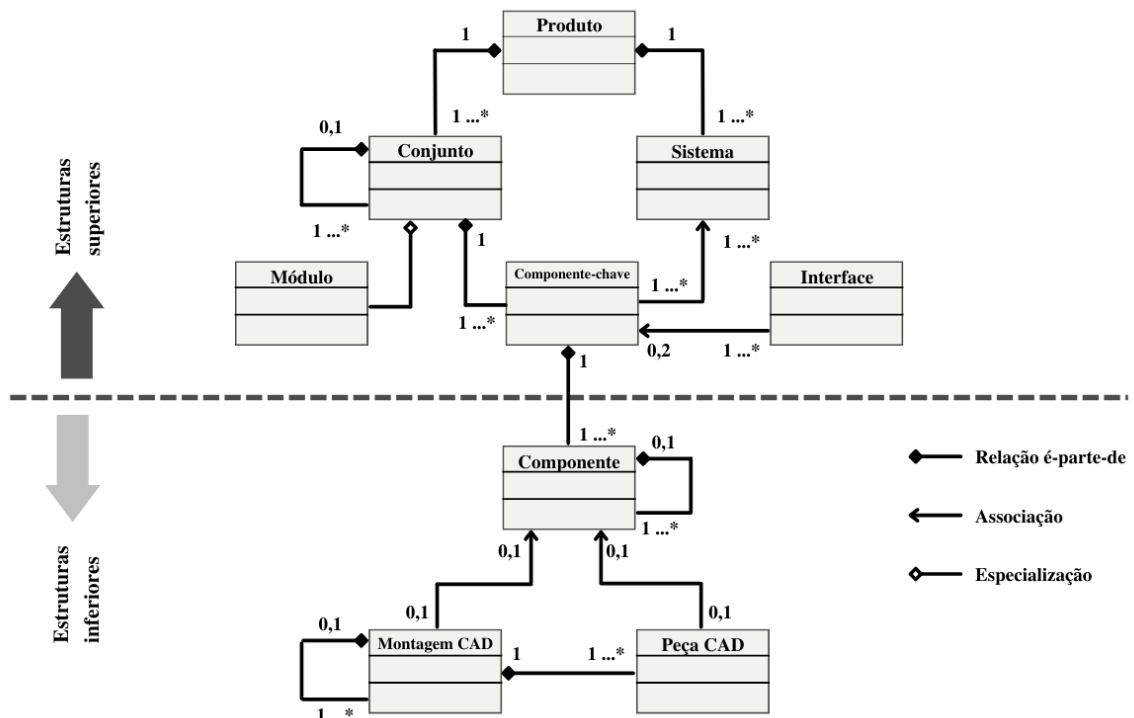
viii. Modularidade liga-e-desliga

A modularidade liga-e-desliga é um tipo especial de intercambialidade de componentes, onde o módulo que liga e desliga é opcional, e seu espaço pode ser deixado vazio sem prejuízo ao produto.

2.1.5 Elementos-chave da modularidade

É comum que na literatura sejam descritos diferentes elementos-chave para modelar uma estrutura de produto. Os conceitos em geral são muito similares, mas podem haver pequenas diferenças entre os modelos de cada autor. Nas seções a seguir, serão descritos os principais elementos encontrados na literatura, do qual se aproveitarão parte das definições para o estudo de caso do presente trabalho. Um diagrama geral é mostrado na Figura 2.3:

Figura 2.3 – Diagrama UML da estrutura de dados em sistemas PLM



Fonte: Adaptada de Bruun et al. (2014)

Alguns dos elementos mostrados nessa figura serão explorados a seguir, tratando de suas definições e relacionamentos entre as demais entidades. No entanto, para efeitos do presente estudo, nem todos os elementos serão necessários.

2.1.5.1 Arquitetura de produto

Considerando Bruun et al. (2014), a arquitetura de produto descreve como uma família de produtos é constituída para atender as variantes. É definida pela atividade de

compor um sistema em módulos independentes que devem ser tratados como unidades lógicas, definindo as possibilidades de produtos que existirão. Existem relações 1:1, n:1 ou ainda 1:n entre módulos e componentes físicos, mas também existem relações entre função e módulos. É importante garantir que os módulos sejam intercambiáveis, através de interfaces bem definidas.

O projeto modular permite, mas também exige, que as atividades de projeto de produto sejam paralelizadas, com colaboração entre disciplinas. Uma ferramenta que habilita a paralelização e controle desse tipo de atividade é a Engenharia Colaborativa, como discutido em Back et al. (2008).

2.1.5.2 *Conjunto*

Bruun et al. (2014) destaca que o conjunto é um agrupamento de módulos ou outros conjuntos, compondo uma estrutura similar a uma *bill-of-materials (BOM)*. A visão de conjuntos é diferente da visão de sistema, pois não pode haver sobreposição entre os conjuntos, i.e cada módulo só pode pertencer a um único conjunto.

2.1.5.3 *Sistema e subsistema*

Um sistema pode ter as mais diversas definições em diferentes contextos, mas dentro de projeto de produto, o sistema é, segundo Back et al. (2008), “uma combinação de elementos ou partes que forma um todo complexo que tem ou serve a um propósito”, e ainda complementa que “esse propósito pode ser declarado por uma função geral ou global”. Tal definição, mesmo no estudo da modularidade, é muito similar ao proposto por Bruun et al. (2014), onde “um sistema é visto como um meio de um produto, que realiza sua função e carrega suas propriedades”, mas faz uma conexão com a necessidade de conceituar os módulos, pois “o propósito do desenvolvimento do sistema é suportar funcionalidades que estão dispersas entre diversos módulos”. A estrutura de sistemas é composta por componentes-chave, mas é sobreposta, pois um componente-chave pode pertencer a diferentes sistemas simultaneamente. No entanto, deve haver um sistema primário, de modo que se um filtro for aplicado e apenas restarem as relações primárias, a estrutura torna-se uma árvore hierárquica.

2.1.5.4 *Componente e componente-chave*

Ainda de acordo com Bruun et al. (2014), o componente-chave é o menor elemento considerado no diagrama de interfaces, e seu propósito é decompor sistemas e módulos em blocos construtivos menores. Os componentes-chave, por sua vez, são constituídos por componentes, e estes estão diretamente associados a arquivos CAD (peças únicas ou montagens). De um ponto de vista de sistema, um componente-chave pode ser associado a um sistema primário e um ou mais sistemas secundários. Para fins desse trabalho, no entanto, assumiu-se que somente o conceito de componente era relevante para fins de modelagem, e o termo componente-chave não será utilizado adiante.

2.1.5.5 *Interface*

As interfaces são definidas por Bruun et al. (2014) como sendo uma relação entre dois componentes, módulos ou subsistemas (dependendo do caso), sendo que esta relação pode ser uma conexão física, um transporte de energia, um fluxo de informação ou fluxo de material. A importância de definir as interfaces é sistematizar a intercambialidade entre os componentes, caso isso seja relevante para a estrutura de produto. Para que isso ocorra, é necessário garantir a manutenção dos 3F's – form, fit, function - isto é, o componente substituído deve possuir mesma forma, encaixe e função (MORRIS, 2009).

2.1.5.6 *Atributos*

Os atributos são elementos essenciais para representar quais valores são assumidos pelos elementos de uma estrutura de produto, como componentes, módulos e conjuntos. Dependendo da literatura, podem ser usados termos cognatos, como “características” ou “propriedades”. De acordo com Tiihonen et al. (1998), alguns atributos tem valores fixos e outros podem receber valores customizados. Geralmente são compostos pelo nome, um tipo de valor (e.g. texto, numérico) e uma definição de necessidade (opcional ou obrigatório). Podem ser utilizados para representar grandezas físicas – como um comprimento ou corrente elétrica – ou atributos textuais – como a linha a qual o produto pertence.

2.1.5.7 *Portas*

Tiihonen et al. (1998) define que as portas são utilizadas para modelar conexões e compatibilidade entre componentes, ou seja, as interfaces são modeladas como portas. É possível definir tipos de porta e um conjunto de restrições, de modo que um componente possa ser conectado a outro se eles possuem portas compatíveis e que satisfaçam os valores de atributos exigidos pelas restrições. As portas podem ser lógicas ou físicas, sendo que portas físicas podem ser representadas de forma simplificada. Por exemplo, pode ser suficiente – para fins de configuração – definir que um componente possui quatro pontos de conexão, ao invés de modelar uma série de dimensões e geometrias da interface se essas informações não são úteis para estabelecer a intercambialidade.

2.1.5.8 *Restrições*

As restrições, na definição de Tiihonen (1998), são um mecanismo geral para determinar relações de uma estrutura de produto, podendo ser uma regra formal, lógica, matemática ou uma combinação dos tipos. Deve ser possível avaliar, para uma dada combinação de atributos, se a restrição está sendo satisfeita ou não. As restrições devem ser utilizadas quando os demais elementos não forem eficientes em capturar a lógica do produto. É possível, por exemplo, modelar o exemplo da potência elétrica de uma fonte de computador através de restrições, mas nesse caso existe um elemento mais adequado para fazê-lo que seria o conceito de recurso, que não será explorado no presente trabalho. As restrições tipicamente possuem algum tipo de classificação, como por exemplo, restrições técnicas (de engenharia) ou marketing.

2.1.5.9 *BOM, E-BOM E M-BOM*

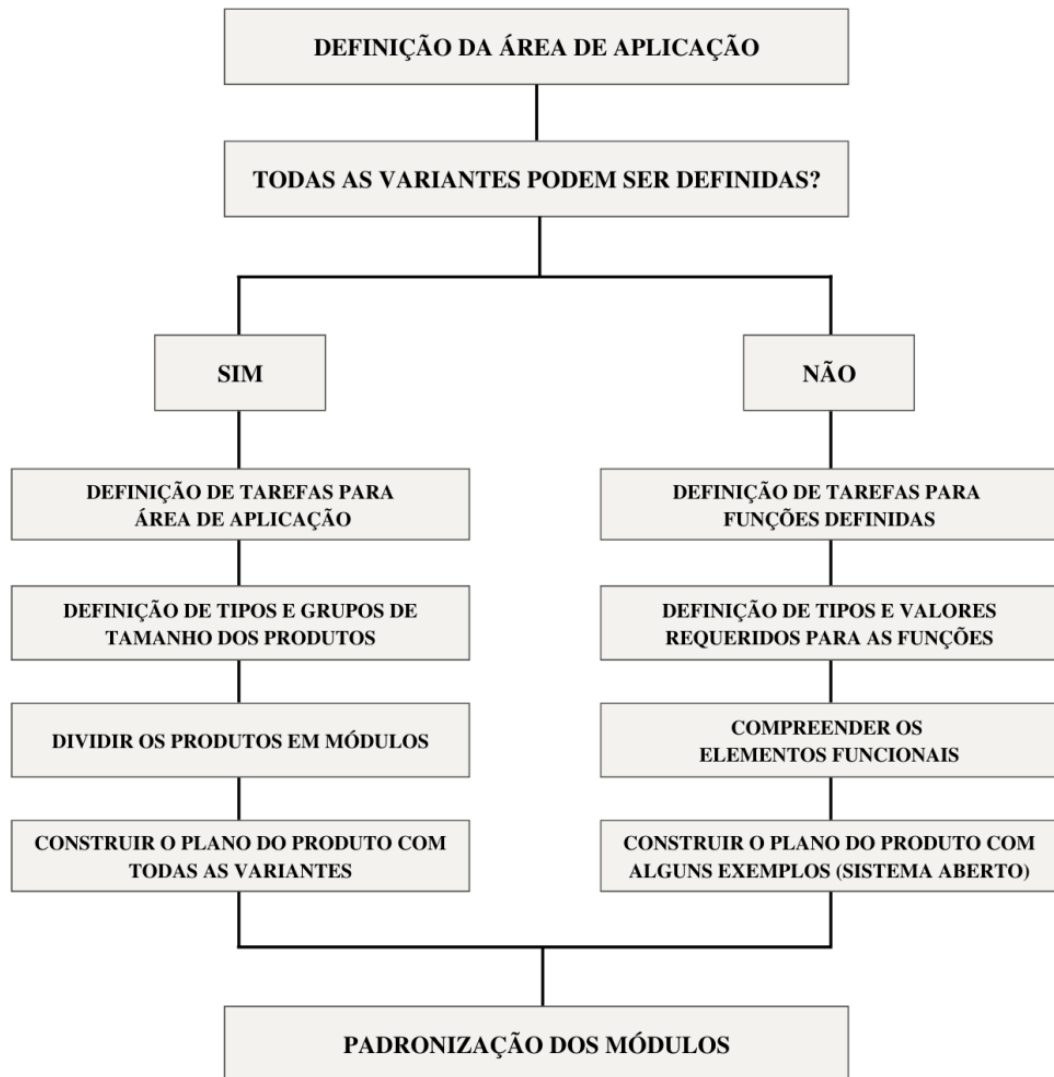
O conceito de *BOM* (*bill-of-materials*) é bastante presente na literatura relacionada à gestão de produtos, sendo definida como uma lista de componentes necessários para manufaturar um produto (PINQUIE et al., 2019). Existem diversos tipos, como os discutidos por Massirer (2007), mas os tipos mais comuns são a *E-BOM* (*engineering manufacturing bill-of-materials*), que consiste em uma lista de componentes considerados durante os processos de engenharia da empresa, e *M-BOM* (*manufacturing bill-of-materials*), que é a lista

considerada durante a fabricação (PINQUIE et al., 2019). Uma M-BOM, por exemplo, conterá itens que só serão utilizados no processo produtivo, ou utilizarão também itens fantasma (itens que somente são utilizados para organização da lista, conforme Massirer (2007)), que não serão encontrados em uma E-BOM. As listas podem ser hierárquicas, sendo organizadas em conjuntos (conforme item 2.1.5.2). Para identificar os componentes, normalmente é utilizado algum tipo de código único e a quantidade daquele componente.

2.1.6 Sistemas abertos e fechados

A definição de sistemas abertos e fechados é de interesse para o presente estudo. Borowski (1961 apud LEHTONEN, 2007) define que um sistema fechado é um sistema em que todas as variantes de projeto estão definidas, enquanto em um sistema aberto, somente algumas variantes são definidas. Isso significa que, em um sistema aberto, será necessário projetar variantes ainda não desenvolvidas conforme elas forem necessárias para atender requisitos de clientes. Um exemplo de fluxo de decisão entre os dois sistemas em tempo de desenvolvimento do produto segue na Figura 2.4:

Figura 2.4 - Como optar entre um sistema aberto ou fechado



Fonte: Adaptada de Lehtonen (2007)

Conforme exposto na figura, a diferença fundamental entre um sistema aberto e um fechado é o fato de todas as variantes serem definidas ou não. As atividades de desenvolvimento de um produto variam em função dessa definição, e é bastante provável que produtos mais complexos cairão na categoria de sistema aberto. No entanto, há notáveis exceções. Automóveis são produtos complexos, com teoricamente muitas variantes possíveis se fosse permitido combinar quaisquer acessórios, motores e transmissões, mas os fabricantes frequentemente limitam as combinações (STÄBLEIN et al., 2011).

2.2 DEFINIÇÃO DE CUSTOMIZAÇÃO EM MASSA

Estando agora bem delimitado o tema de modularidade, convém discutir a customização em massa. Há certo consenso na literatura no que diz respeito ao conceito de customização em massa. Jiao e Tseng (2004) dissertam que a customização em massa objetiva atender os clientes de forma individualizada, com produtividade próxima à produção em massa, conceito similar ao apresentado por Trentin, Perin e Forza (2013). Pine (1993 apud Salvador et al. (2002)) salientam que não somente a produtividade é importante – num sentido estritamente de manufatura – mas principalmente o custo e tempo de entrega devem ser similares à produção em massa.

Um aspecto interessante surge na questão de identificação de necessidades, e impacta diretamente no projeto de produto. Para Piller e Blazek (2014), a companhia que aplica customização em massa precisa identificar as necessidades na qual os clientes divergem mais, em contraste com a companhia que realiza produção em massa, pois esta última foca em atender necessidades universais, que idealmente são compartilhadas por todos os clientes. Simultaneamente, é essencial compreender qual a extensão da variedade de características que o mercado quer e aceita pagar (SALVADOR et al., 2002), pois é preciso extrair valor da customização em massa para que ela se viabilize.

Uma possível necessidade por customização em massa surge das diferentes regulações e restrições específicas, geralmente no nível de país. A todo o momento, surgem novas regulações ambientes e de segurança (SALVADOR et al., 2002), originando variações por si só.

Dentro das definições, é pertinente ressaltar o que não é customização em massa. Conforme Piller e Blazek (2014), a possibilidade de customização não significa que as opções disponíveis são infinitas – i.e, customização em massa não é “produção artesanal em massa” - pelo contrário, é necessário justamente definir e delimitar as opções de forma prévia e sistematizada. A essa atividade, dá-se o nome de “definição do espaço de solução”, que compreende definir quais opções serão oferecidas e quais não serão. Em adição, é essencial mapear as permutações de funcionalidades permitidas. O espaço de solução pode ser definido com diferentes métodos – realizar pesquisas de mercado convencionais é a estratégia mais comum, mas também é possível fornecer ao cliente softwares de co-criação (por exemplo, um aplicativo ou CAD simplificado), ou ainda utilizar-se de inteligência de mercado para coletar continuamente informações e atualizar as opções disponíveis.

Um precursor do processo de customização em massa são os processos artesanais de fabricação. Nesse tipo de processo, no entanto, há contínua reinvenção dos produtos e processos a cada nova entrega, enquanto que na customização em massa, há um processo bem estabelecido para alcançar eficiência, através de processos estáveis e pouco mutáveis, mas, ainda sim, flexíveis. Essa flexibilidade pode ser alcançada, por exemplo, através de automação com robôs (PILLER e BLAZEK, 2014) e impressão 3D (GUO et al., 2021).

2.2.1 Benefícios

A literatura apresenta uma vasta relação de benefícios da customização em massa. Além do imediato atendimento mais próximo dos requisitos do cliente, Piller e Blazek (2014) destacam que se estabelece uma relação de lealdade entre cliente e fabricante, que serve como barreira para evitar a migração do cliente para um concorrente. Essa relação decorre do processo de definição do produto, que frequentemente acontece em um regime de co-criação. Salvador et al. (2002) evidenciam que a grande quantidade de opções disponíveis reduz a habilidade dos clientes de compararem diretamente os produtos, o que por conseguinte ajuda a consolidar lealdade a uma companhia.

Ademais, benefícios nas áreas produtivas podem ser observados. Como os produtos são majoritariamente ou na sua totalidade fabricados após serem vendidos, há uma redução nos estoques de produtos acabados, em concordância aos preceitos de manufatura enxuta (LIKER, 2004). Especificamente sob esse aspecto, pode-se dizer que a customização em massa é *lean*.

Do ponto de vista estratégico, pode haver mercados inexplorados, que aceitarão pagar mais pela customização. As companhias devem ser capazes de estabelecer uma análise de *trade-off*, i.e avaliar a preferência do cliente por customização e estimar custos adicionais de manufatura ao prover a variedade necessária (JIAO E TSENG, 2004). Essa pode ser uma estratégia da companhia para uma determinada área de negócio, produto, ou mesmo para a empresa inteira.

2.2.2 Desafios e pontos de atenção

A customização em massa traz custos adicionais para uma companhia que trabalha com produção em massa. Jiao e Tseng (2004) discutem que há um paradoxo nesse modelo,

pois é necessário constantemente reestruturar a variedade externa, induzida pelo mercado, levando em consideração a complexidade interna resultante da diferenciação dos produtos. Zipkin (2001) chega à conclusão semelhante, destacando a inerente contradição no termo “customização em massa”.

Baker et al. (2007) relata uma série de possíveis problemas, como maior tempo de engenharia para ordens customizadas, requisitos do cliente não sendo corretamente traduzidos, demora para resposta nas cotações e dificuldade para prever questões relacionadas a entrega.

Do ponto de vista de gestão da cadeia de suprimentos, podem surgir desafios ao tentar atender de maneira rápida a necessidade de mudanças (SALVADOR et al., 2002). Determinadas solicitações do cliente podem resultar na necessidade de componentes que tem prazo de fornecimento mais longo que o habitual, o que atrasará a entrega do produto final ao cliente. Para evitar manter estoques elevados de uma grande variedade de itens, o mais prudente nesse cenário é dispor da informação do prazo de entrega mais longo já no período de negociação do produto, antecipando essa informação ao cliente.

No processo produtivo, existem dificuldades relacionadas a tempos de *set-up* elevados, a complexidade do planejamento de materiais e o controle do chão de fábrica (SALVADOR et al., 2002). É compreensível que um número maior de operações e variedade de produtos trará esse tipo de adversidade nas etapas de fabricação.

Do ponto de vista financeiro, Zipkin (2001) também destaca a dificuldade de traduzir a customização em maior faturamento, pois evidentemente há um limite no mercado em pagar mais por certo requisito. É possível adotar uma estratégia de apenas atender clientes muito específicos, que aceitarão pagar por requisitos extremamente especiais, mas essa abordagem certamente limita bastante o *market share* que pode ser alcançado.

2.2.3 Como resolver os problemas da customização em massa

É importante destacar que nem todas as empresas passam pelo mesmo processo de customização em massa. As diferentes estruturas de produto, volumes de produção e variedade disponibilizada requererão diferentes formas de implementar e gerenciar os processos internos.

Assim como o conceito da customização em massa está bem estabelecido, também existe uma vasta literatura de recomendações para lidar com os problemas discutidos

anteriormente. A importância da gestão da arquitetura de produtos dentro de uma visão de modularidade do produto é um aspecto crucial para o sucesso da implementação é destacada por Salvador et al. (2020), que consiste em antecipar ao máximo os requisitos de mercado na etapa de projeto do produto, a fim de reduzir a necessidade de atuação de projetistas e engenheiros em avaliações futuras. A arquitetura de produto, conforme já discutido, diz respeito a maneira que os atributos solicitados pelos clientes são distribuídos nos componentes e combinadas para obter as diferentes configurações de produto.

Salvador (1993 apud Piller e Blazek, 2014) trazem outros preceitos da customização em massa sustentável, como o desenvolvimento do espaço de solução e o projeto de processo robusto. A definição do espaço de solução traduz-se na associação das opções existentes desse espaço com as necessidades do mercado-alvo. O passo seguinte é definir uma arquitetura modular da estrutura de produto. Esse ponto será melhor explorado na próxima seção.

Uma estratégia que tira proveito da arquitetura de produto é a postergação da diferenciação da produção (FORZA e TRENTIN, 2006). Essa estratégia se baseia em ter um processo produtivo que, em sua maior parte, acontece de forma igual para os produtos, e o produto somente recebe a customização nas etapas finais de produção. Outras estratégias já mencionadas como automação com robôs (PILLER e BLAZEK, 2014) e impressão 3D (GUO et al., 2021) também são estratégias válidas de um ponto de vista produtivo.

2.2.4 Design for Mass Customization – DFMC

Um tópico à parte em termos de recomendações para alcançar uma estratégia bem sucedida de customização em massa em uma companhia é o conceito de *Design for Mass Customization – DFMC*, que consiste em desenvolver produtos levando em consideração que eles serão fornecidos em um processo de customização em massa. Alguns preceitos principais discutidos na literatura são (i) estratégia de projeto em nível de família de produtos, (ii) modularidade combinatorial, (iii) índices de compartilhamento, (iv) customizabilidade e (v) foco em *fit*, *form* e *function*. Esses preceitos serão discutidos a seguir.

Durante o desenvolvimento, a estratégia de projeto pode ser estabelecida em nível de família de produtos (SALVADOR et al., 2002), agrupando produtos finais que são minimamente parecidos entre si do ponto de vista funcional, tecnológico e de processos produtivos. A família de produtos deve satisfazer a modularidade combinatorial, que se

fundamenta em definir interfaces padronizadas entre os componentes, o que permite a intercambialidade.

Durante o processo de projeto do produto – ou família de produtos, Collier (1981) estabelece os índices de compartilhamento de uma *bill-of-materials* – ou *BOM* (apud JIAO e TSENG, 2004). Esse índice revela o quanto um componente é reaproveitado em diferentes produtos, sendo idealmente o maior valor possível – o que indica alto reaproveitamento. Essa métrica é relevante para comparação entre produtos antigos da mesma companhia, por exemplo, pois pode não ser aceitável que uma família de produtos em etapa de desenvolvimento tenha um índice de compartilhamento inferior.

A customizabilidade do projeto também é um item fundamental. Jiao e Tseng (2004) conceituam que a customizabilidade diz respeito à facilidade e flexibilidade com a qual as customizações previstas podem ser executadas, sem provocar mudanças profundas na estrutura de produto - evidenciando uma arquitetura robusta de produto.

2.2.5 O papel dos configuradores de produto na customização em massa

O conceito de configurador de produto será explorado de forma mais aprofundada na seção 3.4, mas é importante ressaltar sua importância dentro do conceito de customização em massa. Myrodia et al. (2017) destacam que a utilização de sistemas de configuração de produto (SCP) é um dos fatores-chave de sucesso no âmbito da customização em massa. Isso se deve a capacidade dos configuradores de produto de simular potenciais projetos antes da compra, possivelmente em uma interação próxima com o cliente, caracterizando um “co-projeto”. Cliente e fabricante precisam trabalhar para identificar as necessidades através do SCP, articulando os requisitos a serem customizados (PILLER e BLAZEK, 2014). Indo além, um SCP robusto o suficiente – em termos de garantia de que o espaço de solução é coerente com a realidade – e de fácil uso – no sentido de poder funcionar apenas com requisitos de alto nível do cliente – potencializa o ganho para a empresa, pois o cliente pode explorar sozinho o espaço de solução disponibilizado pela empresa através do sistema.

Os SCP para customização em massa devem ser diferentes dos SCP para requisitos técnicos complexos (TRENTIN et al., 2013). O SCP para customização em massa preconiza a utilização mais próxima do cliente final, enquanto que o SCP para complexos requisitos técnicos estará mais próximo das camadas de engenharia, logo, mais afastado do cliente.

Uma forma que tem se tornado popular nos últimos anos é oferecer a possibilidade de customizar um produto em canais online de vendas através de um configurador de produto (SANDRIN et al., 2017).

2.2.6 Estratégia de conexão vendas-manufatura

A conexão do cliente ao produto – através da área de vendas da companhia e chegando à manufatura, onde o produto será efetivamente produzido, pode estar sujeita a diferentes estratégias em uma companhia que adota a customização em massa. Conforme explorado na seção 1.1, o paradigma de produção típico é o *Configure-to-Order* (CTO). Piller e Blazek (2014), no entanto, apresentam diversos modelos possíveis de produção relacionando-os com o configurador de produto. Apresentam-se a seguir os tipos de configuradores na Tabela 2.1, uma adaptação extraída dos mesmos autores:

Tabela 2.1 – Tipos de configuradores

Tipo	Descrição	Complexidade
<i>Select-to-Order</i> (STO)	Cliente seleciona todos os componentes para um produto. Não há dependências entre os componentes	Baixa
<i>Pick-to-Order</i> (PTO)	Cliente seleciona componentes de um produto e cuida sozinho das dependências entre os componentes	Baixa
<i>Assemble-to-Order</i> (ATO)	O configurador combina componentes pré-fabricados considerando dependências	Média
<i>Configure-to-Order</i> (CTO)	O configurador suporta o cliente ao selecionar os componentes que satisfazem as dependências baseados em um sistema modular	Média
<i>Make-to-order</i> (MTO)	O configurador permite o cliente definir parâmetros específicos baseados em regras. A fabricação ocorre após a efetivação do pedido.	Média-alta
<i>Engineer-to-order</i> (ETO)	Elevado grau de liberdade. Novos componentes e regras podem ser necessários para satisfazer as necessidades de configuração do cliente	Alta

Fonte: Adaptada de Piller e Blazek (2014)

As estratégias de produção também são assunto de elevado interesse para os softwares de planejamento de recursos da empresa – *Enterprise resource planning* – ERP, que é a espinha dorsal de toda a operação de muitas companhias. Não por acaso, os termos ETO, CTO, MTO, MTS são encontrados nos sistemas de gestão, como o SAP (JAIN, 2014).

Uma companhia pode eventualmente combinar os fluxos para diferentes produtos, ou ainda para diferentes vendas, buscando retirar o máximo de proveito das vantagens de cada estratégia. Por exemplo, uma companhia pode fornecer a maior parte de seus produtos através de uma estratégia CTO, suportado por um configurador de produto com elevadas capacidades de validação do produto, e quando o cliente exige requisitos muito complexos e que nunca foram avaliados antes, a venda torna-se ETO – de modo que será avaliado sua factibilidade pelos engenheiros responsáveis.

2.3 PROCESSOS ENVOLVENDO CUSTOMIZAÇÃO EM MASSA

Disponibilizar um novo produto ao mercado requer o desenrolar de um processo interno nas companhias de elevada complexidade. Diversas equipes, com diferentes especialistas, agem coordenadamente por um período pré-definido de tempo para que isso possa ocorrer. No âmbito da customização em massa, e conforme o conceito de DFMC – discutido de forma breve anteriormente – esse processo pode se modificar em relação ao que é tradicionalmente empregado.

Existe uma vasta literatura a respeito do processo de desenvolvimento de novos produtos. Diversos autores dissertaram sobre o tema, adaptando conceitos de outros pesquisadores para suas realidades locais. Esta seção se propõe a explorar o processo tradicional de desenvolvimento de novos produtos, para em seguida explorar o processo de desenvolvimento de novos produtos para customização em massa. Nesse segundo cenário, existem dois processos importantes: o desenvolvimento em si – isto é, a disponibilização do produto para a venda – e o processo de venda.

Diferentemente do processo tradicional de projeto de produtos, é necessário detalhar o processo de venda na customização em massa porque ela ocorre de forma completamente diferente da venda de um produto seriado, disponibilizado em um processo convencional de desenvolvimento de produto. Para um produto seriado, o processo de venda pode ser tão simples quanto o MTS – envolvendo a escolha de um produto em um estoque – ou MTO – envolvendo a venda e fabricação do produto sob demanda. Para um produto customizado em massa, a venda se torna um processo mais complexo, geralmente envolvendo um SCP; na sequência, pode haver ou não intervenção de projetistas e engenheiros na definição do produto; e só então o produto poderá ser fabricado. Esse processo será discutido em maiores detalhes nas próximas seções.

2.3.1 Processo de desenvolvimento de novos produtos

Por desenvolvimento de produtos, se compreende “os aspectos de planejamento de projeto, ao longo de todas as atividades da sequência do processo, desde a pesquisa de mercado, o projeto do produto, projeto do processo de fabricação, plano de distribuição e de manutenção até o descarte ou desativação do mesmo” (BACK et al., 2008). É um processo, portanto, que engloba todas as informações que compõem o ciclo de vida do produto. Uma

especialização do termo é o “desenvolvimento integrado de produtos”, que pondera que todo o processo deve ocorrer de forma simultânea, conduzido por uma equipe multidisciplinar – podendo ser utilizado também os termos de engenharia simultânea ou, em inglês, *concurrent engineering*. Por novo produto, entende-se três classificações principais: variantes de produtos existentes, que consiste em complementar e reposicionar produtos previamente desenvolvidos pela companhia; inovativos, que são produtos existentes que recebem alterações para agregar valor; e criativos, que são produtos completamente novos (BACK et al., 2008).

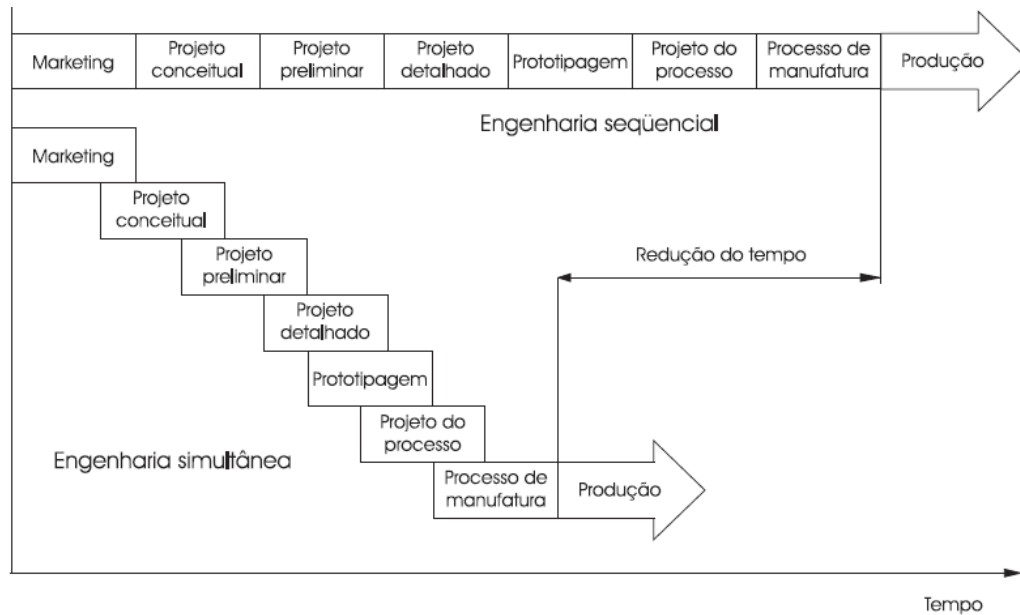
O projeto em si é um plano para obter um fim – no caso, o produto, através de meios científicos, informações técnicas e definição de estruturas, máquinas e sistemas (Back et al., 2008). Para isso, é essencial definir o que, quando, com o que e como fazer, estabelecendo uma metodologia a ser seguida. A justificativa para estudar esse tema é fortemente relacionada à influência que as etapas iniciais têm sobre o custo total do produto – 70% de custo é definido já na etapa de projeto (BARTON, LOVE e TAYLOR 2001 apud BACK et al., 2008).

2.3.2 Engenharia simultânea

A engenharia simultânea é um tema de relevância para o presente estudo, porque é um conceito de importância também para a customização em massa. Segundo Prasad et al. (1998), “é uma abordagem sistemática que considera todos os aspectos do gerenciamento do ciclo de vida do produto, incluindo integração do planejamento, projeto, produção e fases relacionadas” (apud BACK et al., 2008). Os preceitos da engenharia simultânea incluem ainda segundo Back et al. (2008) “tratamento simultâneo de restrições de projeto e manufatura, compartilhamento de conhecimentos associados ao desenvolvimento do produto, consideração do ciclo de vida do produto, ênfase às preferências dos consumidores no desenvolvimento do produto e desenvolvimento do produto considerando qualidade, custo e tempo para o mercado”.

É possível representar as diferenças entre a engenharia simultânea e o processo convencional – também denominado engenharia sequencial – de forma gráfica, conforme a Figura 2.5, como feito por Yazdani e Holmes (1999 apud BACK et al., 2008).

Figura 2.5 – Engenharia sequencial e simultânea



Fonte: Extraída de Back et al. (2008)

Conforme a figura, as etapas de projeto ocorrem com certa sobreposição, o que permite reduzir o tempo total até obter um produto fabricável. Outros benefícios são, segundo Clausing (apud BACK et al., 2008):

- o desenvolvimento dos sistemas de produção e das áreas de apoio começa cedo;
- a análise dos aspectos relacionados ao produto ocorre simultaneamente entre projeto, produção e logística, como um sistema único;
- a facilidade em obter um bom projeto para manufaturabilidade e apoio logístico;
- a produção e as pessoas das áreas de apoio ganham um claro entendimento do projeto e comprometem-se para seu sucesso;
- as modificações no protótipo são reduzidas porque o projeto se torna mais maduro desde as fases iniciais.

Esses benefícios são bastante evidentes considerando a Figura 2.5. Ao sobrepor as etapas, alcança-se uma antecipação máxima dos requisitos futuros, evitando idas e vindas típicas do processo sequencial, por conseguinte reduzindo retrabalho.

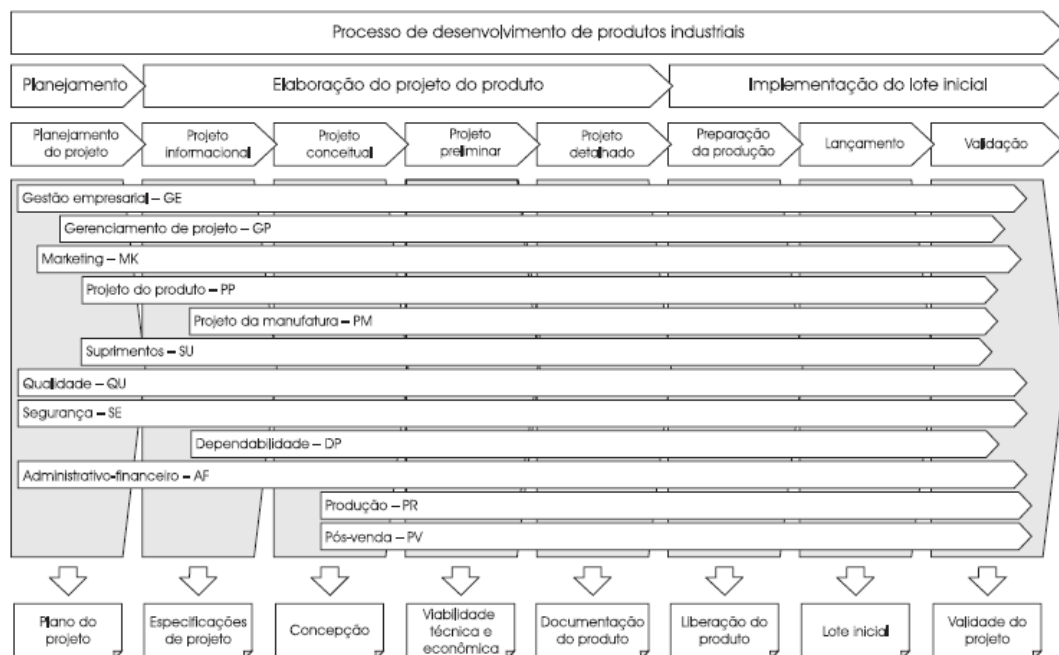
2.3.3 PRODIP – Processo de desenvolvimento integrado de produtos

O processo de desenvolvimento integrado de produtos proposto em Back et al. (2008) é denominado PRODIP, e consolida uma série de pesquisas e experiências desenvolvidas pelo NeDIP (Núcleo de Desenvolvimento Integrado de Produtos). Ainda pelos

mesmos autores, traz alguns preceitos principais, como: conformidade com a visão estratégica da organização; representação gráfica do processo de desenvolvimento; compõe-se em macrofases, fases, atividades e tarefas, sendo que as fases possuem uma passagem bem marcada; e suporte à estrutura organizacional matricial.

As três macrofases, ainda segundo Back et al. (2008), são o planejamento de projeto, cujo objetivo é elaborar um plano para executar o projeto do produto; a elaboração do projeto do produto, que se desdobra nas fases de projeto informacional, projeto conceitual, projeto preliminar e projeto detalhado; e por fim, a implementação do lote piloto, que consiste em executar o plano de manufatura, produzir o lote piloto, validar e liberar o produto. Essas fases podem ser visualizadas na Figura 2.6.

Figura 2.6 – Representação gráfica do modelo PRODIP



Fonte: Extraída de Back et al. (2008)

Para cada tarefa ou atividade, dentro dessa metodologia, é possível destacar os seguintes elementos: entradas, que são as pré-condições para a execução da tarefa – informações ou objetos físicos; os mecanismos, que são as metodologias técnicas e ferramentas necessárias; os controles, que servem para monitorar a execução da tarefa, e as saídas, que são as informações ou objetos físicos finais – estes serão utilizados na próxima tarefa.

2.3.3.1 *Planejamento do projeto*

A primeira macrofase do projeto traz o início de um processo que resultará em um novo produto. Inicia-se, nessa macrofase, o planejamento de marketing, que será executado ao longo de todas as macrofases, buscando informações no mercado que possam influenciar alguma etapa. Também se elaboram os planos de comunicação, escopo do projeto, justificativas e as saídas desejadas para cada etapa.

2.3.3.2 *Elaboração do projeto do produto*

A primeira fase da elaboração do projeto do produto é a fase informacional. Nela, estabelecem-se as necessidades dos clientes, através do mapeamento de requisitos, que podem ser classificados em atributos funcionais, ergonômicos, de segurança, de confiabilidade, de modularidade, estéticos, legais, etc. Os atributos mapeados a partir dos clientes serão desdobrados em especificações de projeto, considerando impactos na manufatura, cadeia de fornecedores, segurança, dependabilidade e custo do produto.

O projeto conceitual é a etapa que visa estabelecer a estrutura funcional do produto (BACK et al., 2008). Nessa etapa, elaboram-se propostas alternativas de estruturas, que serão comparadas e validadas considerando a especificação do projeto elaborada na etapa anterior, os custos resultantes, riscos, manufaturabilidade, qualidade, entre outros aspectos.

Após definido um conceito de produto, é necessário identificar ou desenvolver novos processos de fabricação, além de envolver fornecedores para estabelecer prazos. Ao final dessa etapa, o conceito deverá estar validado e seguro o suficiente para avançar para a etapa seguinte, que é o projeto preliminar.

O projeto preliminar consiste em transformar o conceito do produto em uma estrutura física do produto, com informações mais precisas a respeito de forma, posições, materiais, manufatura, segurança, entre outros aspectos. Nesse momento, as dimensões e posições são simplificadas, mas servem para validar os aspectos de fabricação e segurança através do uso de diversos tipos de protótipos.

Nessa etapa, são utilizados modelos icônicos, analógicos, numéricos e computacionais para executar tais validações. Novamente, tais validações servem para permitir prosseguir com segurança para a etapa seguinte, o projeto detalhado.

O projeto detalhado objetiva elaborar uma documentação precisa a respeito do produto, as quais poderão ser utilizadas sem prejuízo informacional na manufatura do lote piloto. Os componentes do produto passam por um processo de aprimoramento e otimização. Nessa etapa pode ser necessário concluir certificações de produto (HASELEIN e SILVA, 2019), bem como aprovar solicitações de investimentos em máquinas, processos e colaboradores que participarão da fabricação.

2.3.3.3 *Implementação do lote piloto*

Ainda considerando Back et al. (2008), a macrofase seguinte é a implementação do lote piloto, que é dividido em três partes. A primeira é a preparação para produção, onde inicia-se a implementação do planejamento de marketing – que foi desenvolvido simultaneamente ao longo da macrofase anterior, bem como a elaboração de documentos para fabricação de ferramental e montagem do produto. É necessário validar os procedimentos de montagem e treinar os colaboradores, além de validar o produto em si – se atende as especificações estabelecidas. Também corre paralelamente o treinamento das áreas de vendas e pós-vendas. Uma análise final da viabilidade econômica do produto é realizada, agora com dados reais. Essa análise é fundamental para que, uma última vez, esteja garantida a margem do produto conforme o planejado.

Em seguida, ocorre a fase de lançamento do produto. Produz-se o lote inicial, que deve estar em conformidade com as especificações e as melhorias implementadas ao longo do lote piloto, bem como o lançamento de material de marketing no mercado. Os primeiros produtos merecem uma atenção especial das áreas de pós-vendas, pois qualquer desvio ou problema deve ser imediatamente corrigido.

A última fase é a validação do produto, que ocorre coletando *feedback* dos usuários e passando pelas auditorias necessárias. Deve-se monitorar a performance do produto no cliente através de avaliações padronizadas, coletando informações sobre a operação e a segurança do produto.

2.3.4 Processo de desenvolvimento de produtos para customização em massa

Considerando aspectos discutidos anteriormente como o DFMC e o Projeto Integrado de Produtos, pode-se dizer que um Processo Integrado de Produtos para customização em massa precisaria sofrer certas adaptações em relação ao processo padrão.

As macrofases do PRODIP podem ser seguidas, mas considerando Salvador et al. (2002), não se deve focar o desenvolvimento de um produto de forma isolada, mas sim em uma família de produtos. Essa família de produtos é composta por variantes de produto, que compartilham uma mesma estrutura principal – conforme discutido na seção 2.1.

Algumas famílias de produto podem ter um número virtualmente infinito de variantes (HASELEIN e SILVA, 2019), o que torna impossível realizar todas as etapas para todos os produtos. Por este motivo, é necessário considerar os itens discutidos nas seções anteriores quando em projeto de produto apto a customização em massa, que incluem:

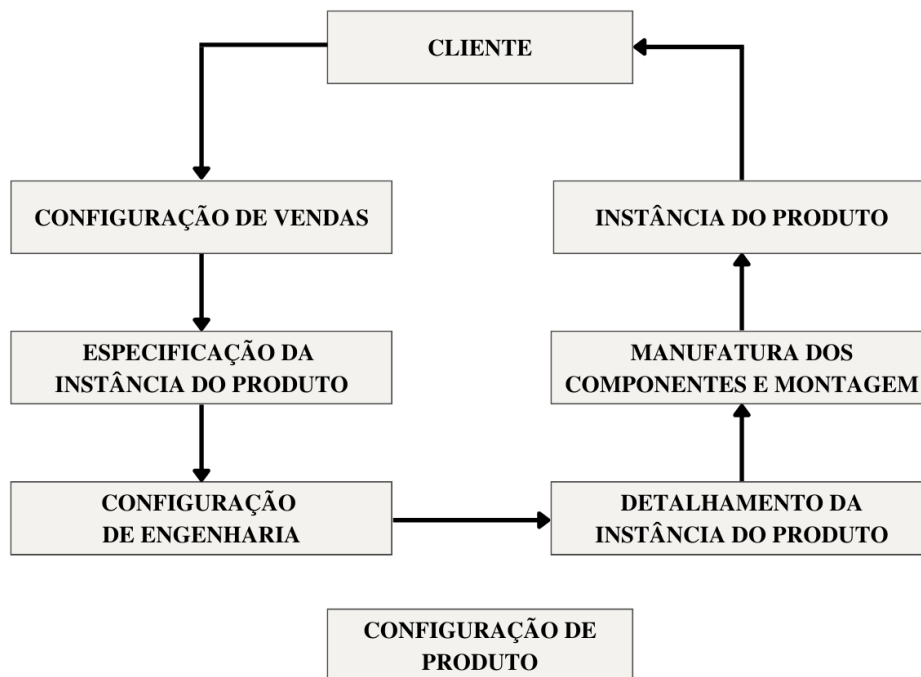
- i. Estabelecer uma arquitetura modular de produto, com um foco na intercambialidade de componentes através da padronização de interfaces, conforme discutido em 2.1.5.1, desenhar os componentes do produto de forma isolada, considerando as interfaces padronizadas;
- ii. Utilizar-se do auxílio de sistemas computacionais para realizar as combinações de componentes para obter diversas variantes de produto. Através desses sistemas, é possível validar uma parte relevante das variantes;
- iii. Selecionar os casos mais críticos para prototipar computacionalmente e depois fisicamente. Por exemplo, selecionar as variantes de um componente com a menor espessura de parede e a maior espessura de parede, a fim de validar resistência e frequências naturais dos casos limítrofes;
- iv. Estabelecer uma estratégia de conexão de vendas à manufatura: STO, PTO, ATO, CTO, MTO, ETO, conforme discutido na seção 2.2.6;
- v. Definir se serão utilizados sistemas computacionais no processo de venda; em caso afirmativo, incorporar a preparação desses sistemas no processo de projeto da família de produtos, para que, no lançamento do produto, seja suportado o processo de configuração de produtos, como será discutido em maiores detalhes na seção a seguir.

2.3.5 Processo de vendas de produtos sob customização em massa

O processo de venda de produtos sob customização em massa acontece de forma diferente do processo para produtos manufaturados em larga escala, sem maiores opções e variantes, e por isso é importante ressaltar como isso ocorre.

Brown (1998 apud LEHTONEN, 2007) definiu uma rotina sistemática representada pela Figura 2.7. Nela, o cliente entra em contato com um canal comercial, onde será realizada a configuração de vendas. Essa configuração opera em um nível mais alto do produto, resultando na especificação do produto solicitado. Em seguida, essa especificação é processada na configuração de engenharia, onde ocorrerá efetivamente a configuração do produto e uma descrição detalhada da instância do produto. Só a partir desse momento será possível manufaturar os componentes, montar e entregar o produto requisitado ao cliente. Esse processo também é descrito por Haselein e Silva (2019).

Figura 2.7 – Processo vendas-entrega com produtos configuráveis



Fonte: Adaptado de Lehtonen et al. (2007)

Esse processo pode variar de acordo com a complexidade do produto, ou se trata de um sistema aberto ou fechado conforme conceito discutido na seção 2.1.6. Em um sistema

aberto, possivelmente serão incluídas novas etapas de validação, pois é necessário validar a manufaturabilidade do produto e questões de *Compliance*, como em Haselein e Silva (2019).

Ainda em Haselein e Silva (2019), esse processo pode ser auxiliado por diversos sistemas computacionais. No início do processo, a configuração de vendas pode ser auxiliada por um SCP. Se houver uma interface bastante simplificada e amigável ao usuário, o próprio cliente poderá conduzi-la, conforme discutido em 2.2.5. Esse processo inicial, que envolve a interação com o cliente, também se pode valer de outras validações necessárias por parte de especialistas na empresa em *Compliance* e manufatura, e essas informações podem circular internamente através de desenhos auxiliares e campos textuais (HASELEIN e SILVA, 2019).

O processo de configuração na engenharia, que visa gerar um detalhamento completo do produto a ser fabricado – incluindo a *BOM* – pode ser também auxiliado ou completamente automatizado por sistemas computacionais (MASSIRER, 2007). Os processos posteriores, que envolvem a manufatura, também podem ser automatizados em larga escala como sugerido pelos preceitos da Indústria 4.0 (OLSEN e TOMLIN, 2020), mas não serão discutidos pois fogem do escopo do presente trabalho.

3 SISTEMAS ESPECIALISTAS PARA CONFIGURAÇÃO DE PRODUTO

Implementar a customização em massa como método de negócio para uma companhia requer diversas estratégias atuando em sincronia. Uma dessas estratégias é a modularidade, conforme explorado na seção 2.1. Projetar produtos considerando a modularidade possibilita elaborar uma estrutura de forma inteligente, entendendo possíveis reaproveitamentos e compartilhando os componentes em diversos produtos finais diferentes. Outra estratégia é o processo de desenvolvimento voltado para a customização em massa, conforme explorado na seção 2.3.4, cujo objetivo é desenvolver uma família de produtos (e não um produto avulso), considerando aspectos como a modularidade e intercambialidade entre componentes. Uma terceira estratégia é o processo de venda, que ocorre sob um fluxo especial – explorado na seção 2.3.5. Nesse processo, cliente e fornecedor trabalham juntos para construir uma definição de solução que atenda às necessidades do cliente, e este será produzido conforme as especificações em um processo flexível de manufatura.

A estratégia que falta ser abordada é justamente a que consolida as demais estratégias em um sistema computacional, sendo essencial nos dias de hoje pelas inúmeras vantagens que traz. Seja pelo lado da companhia, que utiliza sistemas computacionais para garantir a sobrevivência do conhecimento sobre os processos e produtos na companhia, seja pelo lado do cliente, que pode utilizar os mais diversos meios eletrônicos para adquirir o produto sem precisar aguardar a interação com um ser humano, os sistemas computacionais estão por toda a parte. O sistema computacional, no caso, é o sistema de configuração de produto, que será discutido nas próximas seções. Esse sistema pode ser visto como um sistema especialista, pois ele atua como um especialista humano que transcreve as necessidades do cliente em um produto bem definido. Antes, no entanto, abordar-se-ão aspectos introdutórios no tema, a começar pela inteligência artificial.

3.1 INTELIGÊNCIA ARTIFICIAL

A inteligência é tema de estudo de diversas áreas da ciência, como por exemplo, Piaget (1982), que a define como a capacidade de um ser em se adaptar a uma nova situação, mas uma definição precisa do termo permanece tema de discussão na literatura. De forma concomitante ao desenvolvimento da computação, surgiu o conceito de inteligência artificial, segundo a qual uma definição é a de que máquinas que fazem funções que requerem

inteligência similar a de pessoas (KURZWEIL 1990 apud HASELEIN, 2018). Também surgiu em 1950 um teste para definir se uma máquina é inteligente, o renomado teste de Turing (TURING, 1950), que consiste em uma pessoa avaliar respostas dadas por uma máquina e por um ser humano para as mesmas perguntas; caso não seja possível distinguir qual resposta veio da máquina e qual resposta veio do ser humano, trata-se de uma máquina inteligente.

A Inteligência Artificial (IA) foi reconhecida em 1956 como um dos ramos da Ciência da Computação (MATELLI, 2008), ano no qual se realizou a Conferência de Inteligência Artificial de Darmouth. Essa conferência surgiu na esteira do impacto causado pelas publicações em 1950 de Alan Turing.

Ainda haveria um longo caminho a ser percorrido até que aplicações úteis de IA saíssem do meio acadêmico, incluindo dois momentos chamados *AI winter* – períodos em que houve pouco avanço na área. O entusiasmo inicial com a Inteligência Artificial surgiu sob a forma de tentativas de criar algoritmos genéricos que pudessem alcançar a Inteligência Artificial Geral (HASELEIN, 2017). Em paralelo, desenvolviam-se também as primeiras redes neurais, mas eram incapazes de resolver o notável problema XOR, mas essas iniciativas não lograram grande êxito, o que levou ao primeiro *AI winter* – período que ocorreu nos anos 70 (MUTHUKRISHNAN et al., 2020).

Segundo Galbusera et al. (2019), a área de Inteligência Artificial é subdividida em alguns ramos principais, sendo eles a área de representação simbólica, algoritmos evolucionários e aprendizado de máquina (*machine learning*). A área de *machine learning* tem tido grandes avanços nos últimos anos, onde se encontram as sub-áreas de redes neurais artificiais, máquinas de vetor de suporte e métodos estatísticos. Dentro da área simbólica se encontram os sistemas especialistas, alvo do presente estudo. No entanto, é importante ressaltar que existe a possibilidade de haver aprendizado também na área simbólica, através de raciocínio baseado em casos.

Na área simbólica, nos anos 70 e 80, houve um grande crescimento das aplicações baseadas em regras – os sistemas especialistas. Essas aplicações eram muito eficientes em um domínio específico (MATELLI, 2008), como os primeiros sistemas especialistas em 1974 (NORDLANDER, 2001). A disseminação dos sistemas especialistas marcou o fim do primeiro *AI winter*.

Paralelamente no outro ramo da Inteligência Artificial, avanços importantes ocorreram na área de redes neurais, com redes multi-camadas, novos algoritmos de

otimização e *backpropagation*. No entanto, devido à falta de poder computacional, ocorreu o segundo *AI winter*, que só terminou na metade de 1990, graças ao avanço do poder computacional. Em 1998, alcançou-se precisão considerável na leitura de dígitos escritos à mão, e atualmente, existem redes estado-da-arte como a *Imagenet*, com 1.2 milhões de imagens e 1000 classificações, sendo possível construir redes neurais artificiais com baixa taxa de erro (MUTHUKRISHNAN et al., 2020).

No ramo da inteligência artificial simbólica, *frameworks* utilizados para construir sistemas especialistas se transformaram em linguagens mais abrangentes. Os usos se difundiram por aplicações na indústria, como os trabalhos de Massirer (2007), Matelli (2008), Haselein (2018) e Vegini (2018).

Apesar da elevada importância dada pelos mais diversos meios de comunicação para a área de *machine learning* e redes neurais nos últimos tempos, alguns pesquisadores tem salientado sinais de exaustão dessa abordagem – o que poderia indicar que um novo *AI winter* estaria se aproximando. As redes neurais estariam alcançando seu limite, mesmo considerando contínuo aumento do poder computacional. Os autores indicam que para reduzir taxas de erro das redes atuais pela metade, é necessário 500 vezes mais poder computacional, e que algumas empresas tem gasto valores da ordem de milhões de dólares para simplesmente treinar modelos – como Deep Mind com o Go, ao custo de \$35 milhões, e a linguagem GPT-3, ao custo de \$4 milhões (THOMPSON et al., 2021).

Uma discussão de longa data é a possibilidade de máquinas pensarem como seres humanos. Alguns autores como Berry (1983) consideravam plausível essa possibilidade, e em alguns casos, como o jogo de xadrez, os seres humanos foram vencidos por máquinas já em 1997 com o DeepBlue, sendo que atualmente os sistemas baseados em redes neurais artificiais são imbatíveis, como o AlphaZero (SILVER et al. 2017). No entanto, Larson (2021 apud DICKSON, 2021) tem uma visão mais pessimista, argumentando que construir redes neurais artificiais maiores e melhores não levará à inteligência artificial generalizada.

Existem três tipos básicos de inferência, que formam o raciocínio executado por uma inteligência: indução, dedução e abdução. A dedução é a utilização de conhecimento prévio para raciocinar sobre o mundo. Por exemplo, os sistemas especialistas, que utilizam ferramentas simbólicas e um conjunto predefinido de fatos e regras, são capazes de construir deduções a partir dos dados que recebe. A indução envolve a aquisição de conhecimento através da experiência, especialidade das redes neurais artificiais (*machine learning*), onde há aprendizado a partir de padrões observados entre dados de entrada e saídas. A inferência

abdutiva diz respeito a “habilidade cognitiva de apresentar intuições e hipóteses, que permitem fazer suposições melhores que palpites aleatórios sobre a verdade” (DICKSON, 2021). O ser humano tem essa capacidade inata de, a partir de um cenário com infinitas possibilidades, supor causas plausíveis. Essa definição é próxima, mas não equivalente ao senso comum no âmbito da IA, pois, conforme Rich et al. (2009), trata-se da relação entre objetos e relações entre eles, assim como as relações entre ações e suas consequências. Considerando que as redes neurais só permitem inferência indutiva, Larson (2021 apud DICKSON, 2021) justifica que para alcançar a inteligência artificial generalizada, será preciso combinar os três tipos de inferência.

Estudos em 1980 e 1990 tentaram construir uma lógica de programação abdutiva, mas foram abandonadas na sequência. Ainda em Larson (2021 apud DICKSON, 2021), é argumentado que sistemas simbólicos têm suas fragilidades, como a necessidade de regras escritas manualmente, a rigidez para acrescentar novo conhecimento e a dificuldade de transcrição de alguns tipos de conhecimentos em regras ou fatos, por exemplo, a classificação de imagens ou dígitos escritos à mão. Por outro lado, sistemas baseados em redes neurais artificiais não fornecem uma relação de causalidade, não fornecem explicações, lidam mal com casos-limite e necessitam de um volume enorme de dados – além da supramencionada necessidade computacional elevada. Em resumo, um modelo de inteligência baseado apenas em dados passados pode ser considerado incompleto na atualidade.

Uma possível resposta para o tópico de inferência abdutiva seriam as redes bayesianas. Elas tentam computar causalidade, mas ainda há muito campo a ser explorado nesse sentido. Sistemas mistos entre redes neurais artificiais e sistemas baseados em regras, como o IBM Watson despontaram como uma alternativa com potencial de resolver problemas de ambas as abordagens. O especialista em IA Petar Velickovic (ANADIOTIS, 2021) também vê a complementariedade do uso de sistemas algorítmicos baseados em regras e *machine learning*. Algoritmos, por um lado, apresentam vantagens como conseguir trabalhar bem com dados de entrada de tamanhos diferentes e continuar funcionando perfeitamente, além da saída ser de um tipo conhecido, facilitando a conexão entre diferentes sistemas. Por outro lado, as redes neurais se saem melhor em casos com ruídos nos dados. Utilizar os pontos fortes de cada abordagem pode ser a resposta para que a área não caia em um novo *AI winter*.

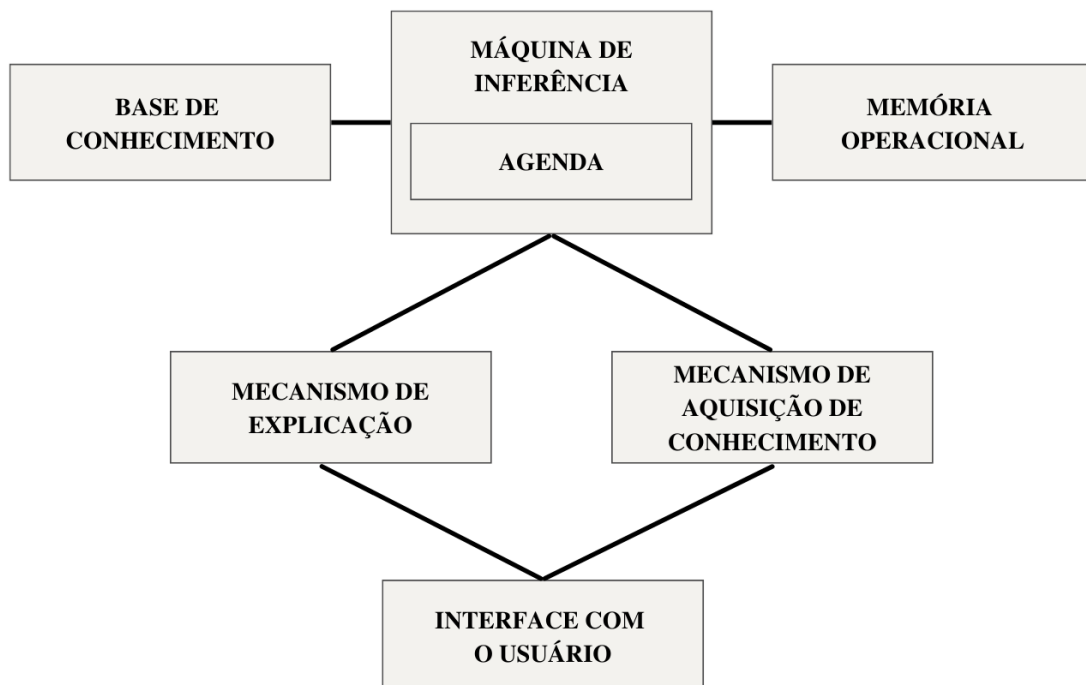
3.2 SISTEMAS ESPECIALISTAS

Os sistemas especialistas foram o primeiro tipo de IA com aplicações reais. São definidos como sendo sistemas que emulam a capacidade de inferência e possuem o conhecimento de um especialista humano (GIARRATANO e RILEY, 2005). Segundo Silva (1998), esse tipo de sistema é composto por quatro partes obrigatórias:

- i. Motor de inferência: encadeia a execução de regras para fatos e objetos
- ii. Base de conhecimento: regras estabelecidas pelos especialistas e modelagem orientada a objetos
- iii. Agenda: lista de regras que devem ser aplicadas num dado momento (memória de trabalho)
- iv. Interface com o usuário

Essas partes podem ser representadas pelo diagrama da Figura 3.1, conforme ilustrado a seguir:

Figura 3.1 – Estrutura de um sistema especialista



Fonte: Adaptado de Giarratano e Riley (2005)

Uma parte que pode ser acrescentada em determinadas implementações é um mecanismo de aquisição de conhecimento. Atkinson e Shiffrin (1968 apud HASELEIN,

2018), dentro das definições de inteligência, destacam a existência da memória em curto e longo prazo em uma entidade inteligente. Pode ser traçado um paralelo com as partes de um sistema especialista, sendo que a base de conhecimento corresponde à memória de longo prazo, enquanto que a agenda seria a memória de curto prazo.

Um sistema especialista pode ser construído a partir da interação de um pesquisador com um especialista em um determinado domínio, sendo necessário mapear o conhecimento para um espaço abstrato – simbólico (ANADIOTIS, 2021). A esse processo de construção de um sistema especialista, dá-se o nome de Engenharia do Conhecimento (WATERMAN 1986 apud SILVA, 1998), e o pesquisador é o Engenheiro do Conhecimento. Esse processo é essencial nas empresas, pois ajuda a mapear e manter a memória corporativa (SILVA, 1998).

A base de conhecimento é formada por regras do tipo SE-ENTÃO. Se a cláusula SE for atendida, i.e for verdadeira, as ações contidas na cláusula ENTÃO são executadas, o que incorre na adição e subtração de fatos que são armazenados na Agenda (MATELLI, 2008). O motor de inferência possui dois tipos de funcionamento, o encadeamento direto ou reverso. O encadeamento direto é utilizado em problemas de síntese, onde o usuário insere informações no sistema e o motor de inferência procura regras que se adequam às informações. No encadeamento reverso, a aplicação é para problemas de diagnóstico (MATELLI, 2008). Nesse caso, o sistema apresenta possíveis hipóteses conclusivas e tenta-se garantir a validade de uma hipótese através da interação com o usuário.

3.2.1 Processo de construção de um sistema especialista

Existe uma literatura bem estabelecida dos passos necessários para construir adequadamente um sistema especialista. Os passos são apresentados por Waterman (1986 apud HASELEIN, 2018).

A etapa I é a identificação do problema, sendo necessário delimitar o escopo, recursos, objetivos e quais serão os especialistas consultados. A etapa II trata da conceitualização, onde o engenheiro do conhecimento – o responsável pelo desenvolvimento do sistema especialista – e o especialista no problema escolhido terão de interagir a fim de construir estratégias e conceitos aplicáveis na solução do problema. A etapa III é a formulação, quando se deve efetivamente modelar e representar o conhecimento adquirido a partir das interações com o especialista. A etapa IV é a implementação, sendo necessário a codificação do sistema em uma *shell* apropriada (esse conceito será explorado logo a seguir).

Por fim, ocorre a etapa V que é a verificação seguida de validação, momento no qual se põe o sistema à prova, testando as respostas obtidas com os próprios especialistas no domínio.

Uma estratégia para ser considerada no ciclo de execução das etapas I até V é a estratégia de desenvolvimento incremental. Ela pode ser resumida em “fazer um pouco, testar um pouco”, e na qual o próprio sistema, ainda que parcialmente pronto, pode ajudar a guiar seu desenvolvimento (LIEBOWITZ 1987 apud HASELEIN, 2018). Na prática, quase todos os sistemas especialistas são desenvolvidos dessa maneira (MASSIRER, 2007). Portanto, haverá uma repetição das etapas diversas vezes até que se alcance um sistema com o resultado esperado.

3.2.2 Ferramentas para construção de um sistema especialista

Para construir um sistema especialista, não se utilizam linguagens procedurais convencionais – i.e, linguagens que só permitem instruções sequenciais. Conforme Giarratano e Riley (2005), as linguagens mais adequadas para construção de sistemas especialistas são não-procedurais e aptas a manipulação simbólica. Dessa forma, linguagens como ADA, PASCAL e C fornecem operações de mais baixo nível, como operações e atribuições de variáveis, mas falham em não fornecerem ferramentas eficientes para manipulação simbólica. As linguagens funcionais são baseadas na construção de funções para serem compostas em um programa mais complexo, seguindo uma lógica *bottom-up*, como a linguagem LISP e PROLOG.

Linguagens não-procedurais, por outro lado, não estão restritas a uma sequência fixa de execução de operações ou funções. Por exemplo, em linguagens orientadas a objeto, tem-se acesso a ferramentas poderosas de representação do conhecimento, como a abstração, encapsulamento, herança e polimorfismo (SILVA, 1998).

Apesar de boa parte das linguagens modernas serem multi-paradigma – por exemplo C ou Java, dificilmente se encontra suporte necessário para construir um sistema especialista sem utilizar um *shell* – ferramentas de desenvolvimento para sistemas especialistas (GIARRATANO e RILEY 2005 apud HASELEIN, 2018).

Para um sistema especialista é necessário um motor de inferência, conforme discutido anteriormente, que irá controlar a execução de regras através de um mecanismo de correspondência de padrões, mas é importante ressaltar que não existe uma sequência pré-definida nas execuções de regras; elas serão executadas conforme houver as condições para

disparo (GIARRATANO e RILEY, 2005). Considerando que construir um motor de inferência em C ou Java é uma tarefa bastante complexa, é mais adequado utilizar um *shell* pronto (MASSIRER, 2007).

O CLIPS é um desses *shell*, tendo seu desenvolvimento iniciado pela NASA-Johnson Space Center em 1985 (GIARRATANO e RILEY, 2005), e significa *C Language Integrated Production System*. Entre suas capacidades importantes para o desenvolvimento de um sistema especialista estão a orientação a objetos – incluindo herança, abstração e relação de pertencimento (SILVA, 1998), possibilidade de ser baseado em regras, programação procedural e capacidade de explicação. Por outro lado, originalmente não suporta encadeamento reverso (GIARRATANO e RILEY, 2005), apesar de poder ser adaptado para esse fim.

3.2.3 Aplicação de um sistema especialista

Um sistema especialista deve ser adotado sob certas circunstâncias. Silva (1998) elenca uma série delas, como:

- i. Existência de especialistas confiáveis na área de domínio, com elevado grau de concordância entre eles sobre as soluções para os problemas. A solução deve ter métodos claros, que podem ser explicados a um iniciante.
- ii. A solução do problema apresenta um elevado retorno, o que pode ser potencializado devido a dificuldade de encontrar especialistas, o ambiente de aplicação é hostil ou de difícil acesso, ou o conhecimento está ameaçado de ser perdido.
- iii. A solução deve ser obtida através de manipulação simbólica, e não problemas matemáticos puros – para os quais algoritmos comuns são mais aplicáveis.

Alguns benefícios de se utilizar um sistema especialista são também destacados por Silva (1998), como a rápida prototipação, a facilidade de obter a explicação do raciocínio utilizado pelo sistema para chegar à resposta, a existência de diversas ferramentas de implementação e a possibilidade de trabalhar com manipulação simbólica.

3.3 PROGRAMAÇÃO POR RESTRIÇÕES

A programação por restrições é um paradigma poderoso para resolver problemas de busca combinatória, fazendo-se valer de técnicas de IA, ciência de computação e pesquisa operacional. A ideia básica é que o usuário defina restrições e um *solver* de restrições multi-propósito é utilizado para resolvê-las. As restrições são as relações que devem ser mantidas entre variáveis de decisão (ROSSI et al. 2006). Os problemas de satisfação de restrições surgiram e se estabeleceram entre os anos 60 e 80.

Dentre os primeiros trabalhos da área, em 1964, Wilkes propôs equações algébricas utilizadas como restrições em linguagens procedurais, utilizando um método de relaxação de restrições. Em 1967, foi desenvolvida uma linguagem declarativa por Elcock, a Absys. Em 1970, Fikes desenvolveu um sistema de resolução geral de restrições baseado em propagação. O desenvolvimento do Prolog (linguagem de programação também utilizada em sistemas especialistas) por Colmerauer também foi um marco importante em 1972, podendo ser visto como uma das primeiras linguagens que suportava a programação por restrições – e também era utilizado para construção de sistemas especialistas, suportando fatos e regras com capacidades similares ao CLIPS. Inúmeros trabalhos de planejamento de escalas, um problema clássico da área, foram desenvolvidos simultaneamente (FREUDER e MACKWORTH, 2006).

As restrições são similares a regras, na qual são estabelecidas relações entre variáveis do problema. Não há obrigatoriedade que essas variáveis sejam numéricas, ou seja, podem ser simbólicas. Um *solver* de restrição, de forma similar a um motor de inferência de um sistema especialista, procuram no espaço de solução do problema através de algoritmos de *backtracking* e *branch and bound* (ROSSI et al, 2006), sendo que esse processo é denominado propagação de restrições. Nesse processo, domínios mais restritos para cada variável de decisão são inferidos a cada etapa de processamento.

Satisfazer as restrições consiste em encontrar um valor para cada uma das variáveis do problema, por exemplo, encontrar os componentes corretos para montagem de uma bicicleta (FREUDER e MACKWORTH, 2006). Esses componentes evidentemente possuem restrições bem estabelecidas entre eles, sendo que esquadro, rodas, freios e correntes devem ser mutuamente compatíveis.

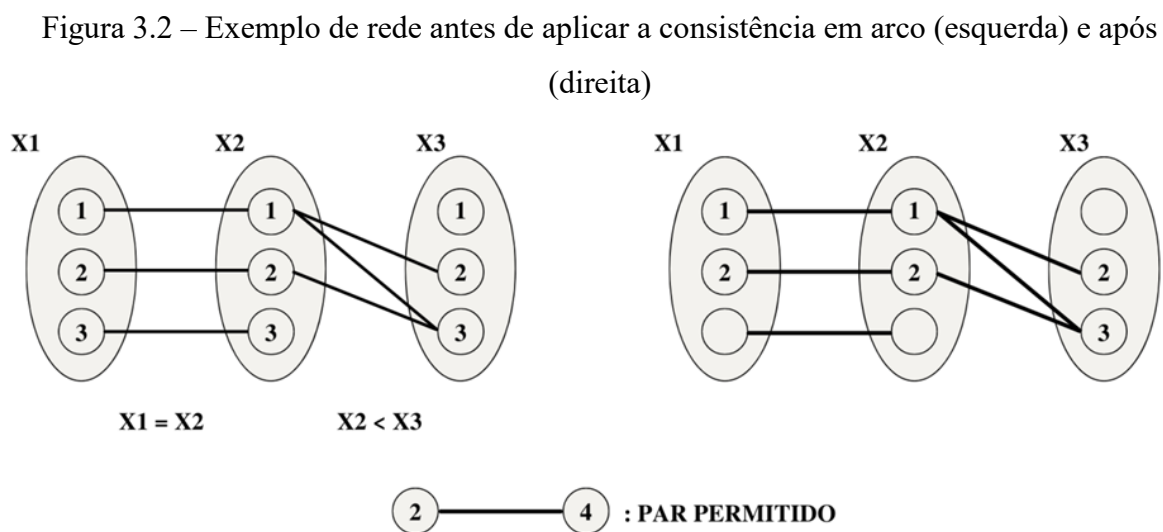
Após a aplicação das restrições, diversos resultados são possíveis (problema sub-restrito). Se as restrições forem muito limitantes, pode não haver um resultado possível para o problema (problema sobrerrestrito).

Abordagens alternativas para a solução de problemas de busca combinatória são a programação linear (LOVÁSZ, 2007) ou alternativas inspiradas em processos naturais, como algoritmos inspirados em colônias de formigas, (PINTEA et al., 2014). Para o problema em estudo, essas abordagens não foram consideradas, pois os sistemas *shell* para construção de sistemas especialistas não implementam essas abordagens.

3.3.1 Consistência em arco

A consistência em arco é um dos marcos da área de Programação por Restrições, tendo sido proposta por Waltz em 1975 (FREUDER e MACKWORTH, 2006). É uma forma de propagar as restrições, sendo um conceito bastante natural e simples que postula que cada valor do domínio deve ser compatível com cada restrição (BESSIERE, 2006).

Ainda de acordo com Bessiere (2006), traz-se uma explicação básica para a consistência em arco. Considerando a rede formada pelos atributos X_1 , X_2 , X_3 , com os domínios 1, 2 e 3, e as restrições $X_1=X_2$ e $X_2<X_3$, a rede mostrada à esquerda na Figura 3.2 não é consistente em arco, pois os valores $X_1=3$, $X_2=3$ e $X_3=1$ são inconsistentes com alguma restrição.



Fonte: Adaptado de Freuder e Mackworth (2006)

Para tornar a rede consistente em arco, é necessário remover esses valores dos domínios de cada variável, isto é, X_1 e X_2 pode assumir somente os valores 1 ou 2 e X_3 pode assumir somente os valores 2 e 3.

3.3.2 Sintaxe de restrições

As restrições podem ser construídas de variadas formas. O *framework* Choco-solver (CHOCO-SOLVER, 2022) implementa programação por restrições e apresenta algumas possibilidades. Podem ser construídas expressões relacionais entre duas variáveis x e y com as seguintes sintaxes:

- $x.lt(y)$ equivale a $x < y$
- $x.le(y)$ equivale a $x \leq y$
- $x.gt(y)$ equivale a $x > y$
- $x.ge(y)$ equivale a $x \geq y$
- $x.ne(y)$ equivale a $x \neq y$
- $x.eq(y)$ equivale a $x = y$

Outro tipo de restrição bastante útil na descrição da maior parte dos problemas são expressões lógicas. Considerando uma expressão relacional r e as expressões relacionais p_1 e p_2 , podem ser escritas restrições com as seguintes sintaxes:

- $r.and(p_1, p_2)$ equivale a $r \wedge p_1 \wedge p_2$
- $r.or(p_1, p_2)$ equivale a $r \vee p_1 \vee p_2$
- $r.xor(p_1, p_2)$ equivale a $r \oplus p_1 \oplus p_2$
- $r.imp(p_1)$ equivale a $r \Rightarrow p_1$
- $r.iff(p_1)$ equivale a $r \Leftrightarrow p_1$
- $r.not(p_1, p_2)$ equivale a $\neg r$
- $r.ift(p_1, p_2)$ retorna p_1 se r é verdadeiro, caso contrário retorna p_2

As funções aqui apresentadas permitem um grande poder de manipulação, inclusive simbólico, através de métodos de implementação. Esse ponto será discutido novamente mais adiante no capítulo referente à modelagem do problema do estudo de caso.

3.3.3 Restrições *soft*

É comum que seja difícil de satisfazer todas as restrições em um problema real. Isso pode acontecer porque o problema é muito restritivo, mas também existe a possibilidade de se estar utilizando restrições para modelar preferências. Nesse tipo de cenário, as restrições não podem ser violadas, mas as preferências podem (MESEGUER et al, 2006).

Um exemplo prático é dado por Meseguer et al. (2006). Num problema de alocação de professores, algumas restrições devem ser tratadas como invioláveis: o tamanho das salas, as horas que a escola está aberta e o fato de que o mesmo professor não pode dar aulas para duas turmas ao mesmo tempo. Por outro lado, as preferências são menos restritas, e podem ser algo como um professor não gostar de dar aula na sexta-feira, ou uma diretora por reduzir o tamanho das turmas se possível. Nesses cenários, metodologias foram desenvolvidas para se obter a satisfação das restrições com o mínimo de violação das preferências.

3.4 SISTEMA DE CONFIGURAÇÃO DE PRODUTO

A tarefa de configurar um produto consiste em definir componentes e atributos que compõem um produto. Um sistema de configuração de produto (SCP) se propõe, portanto, a permitir a execução dessa tarefa com o auxílio de um sistema computacional (JUNKER, 2006). Esse sistema é de fundamental importância para o paradigma de customização em massa, pois através dele os requisitos do cliente serão convertidos em uma especificação manufaturável de um produto. Considerando aspectos de modularidade discutidos na seção 2.1, os módulos que compõem o produto podem ser validados entre si através do SCP, garantindo assim sua manufaturabilidade.

Existem variados exemplos de sistemas de configuração de produtos para computadores, carros, caminhões, e mais recentemente também para serviços, como pacotes de internet, empréstimos, seguros e viagens (JUNKER, 2006). Também há aplicações industriais abundantes. Yost (1996) reporta um sistema de configuração aplicado a elevadores. Falkner et al. (2016) reportam uma série de aplicações em interruptores de telecomunicação, sistemas de sinalização ferroviária, e compressores a pistão, inclusive com interfaces *web*.

Encontrar configurações que podem ser fabricadas para um produto resultam em diversos problemas. Por um lado, podem existir muitas configurações diferentes que atendem

aos mesmos requisitos (problema sub-restrito), mas por outro, podem haver situações em que nenhuma configuração pode atender aos requisitos (problema sobre-restrito) (JUNKER, 2006). Ambas as situações não são desejáveis do ponto de vista do usuário, mas são igualmente difíceis de evitar em sua totalidade.

Uma maneira de evitar os problemas sub-restritos é sempre ter valores preferenciais para cada variável, podendo ser modelado como restrições *soft*. No caso de problemas sobre-restritos, metodologias de resolução de conflito associadas a explicações para o usuário podem ser adotadas.

Trentin, Perin e Forza (2013) propõe que pode haver dois tipos de configuradores, os configuradores de produto e os configuradores de venda. A diferença fundamental entre eles é que o configurador de vendas pretende a traduzir as necessidades do cliente em um produto válido no espaço de produtos da companhia. Para isso, ele apresenta atributos de mais alto nível, facilmente compreendidas por áreas comerciais ou até mesmo pelo cliente final. Um exemplo seria o local de aplicação de um motor elétrico (uma mina, uma indústria comum ou uma indústria alimentícia, por exemplo). Já o configurador de produto se propõe a suportar a tradução das necessidades do cliente em uma especificação mais completa do produto, incluindo a *BOM* e outras informações para a manufatura. Para isso, o configurador de produto apresenta atributos de mais baixo nível, isto é, atributos mais técnicos. Seguindo o mesmo exemplo, no configurador de produto, o tipo de vedação do mesmo motor elétrico seria uma opção disponível. Dispor de dois sistemas separados traz seus próprios desafios, como a manutenção e o sincronismo entre eles.

Há também uma questão referente ao tipo de interface que um configurador de produto pode apresentar. É possível construí-lo como um menu com diversos campos (abordagem tradicional) ou como um *chatbot*. No entanto, apesar dos recentes avanços no processo de linguagem natural, *chatbots* levam a um menor grau de satisfação do usuário (NGUYEN et al., 2021). Portanto, no presente estudo, serão considerados somente configuradores de produto com interfaces do tipo tradicional.

3.4.1 As três gerações de Sistemas de Configuração de Produto

Sistemas de configuração de produto surgiram nos anos 70 e 80 e utilizavam originalmente *shells* utilizadas para a construção de sistemas especialistas ou sistemas simples baseados em regras (HULGAARD, 2019). Essa abordagem inicial foi denominada por

Hulgaard como a primeira geração de configuradores, e as regras nesse tipo de sistema eram construídos no formato IF-THEN. Exemplos são o LO-VC e o SAP *Internet Pricing Configurator*, da empresa alemã SAP. Um problema fundamental com a primeira geração de configuradores é que eles só permitem a configuração dos atributos em uma ordem específica - a ordem que está estabelecida nas regras. É possível contornar esse problema criando mais regras, mas isso acaba levando a uma explosão combinatória de regras, pois todas as ordens possíveis devem ser mapeadas.

Conforme citado anteriormente, entre os anos 60 e 80, emergiu o paradigma de programação por restrições, que são a base de sistemas de configuração de segunda geração – e é também a tecnologia utilizada pela maior parte dos configuradores comerciais atuais (JUNKER, 2006). A cada seleção do usuário, e utilizando-se de uma base de restrições, o *solver* de restrições executa a redução dos domínios de cada variável do problema – conforme discutido na seção 3.3. Uma grande diferença, nesse caso, é que as restrições podem ser bidirecionais (e não somente unidirecionais como um IF-THEN), e, portanto, o usuário pode começar a configurar a partir de qualquer campo disponível (HOTZ et al., 2014), sem que seja necessário grande número de regras.

A terceira geração de configuradores, ainda segundo Hulgaard (2019), é baseada no paradigma de programação por restrições, mas com um espaço pré-compilado de busca, utilizando técnicas como diagramas de decisão binária (HADZIC et al. 2004 apud JUNKER., 2006). Para cenários no qual o número de componentes de um produto é fixo e os atributos têm domínios finitos, essa abordagem pode tornar o processo de configuração bastante rápido (JUNKER, 2006). Hulgaard (2019) assegura ganhos de tempo de resposta consideráveis e a possibilidade de guardar em arquivos compactos todo o espaço de solução mesmo que ele tenha 10^{50} configurações possíveis, apesar de acrescentar um passo de compilação – necessário toda vez que for alterado alguma regra do sistema.

3.4.2 Capacidades de um configurador de produto

Um sistema de configuração de produto possui pelo menos duas funções básicas, de acordo com Hulgaard (2019): um *framework* para manutenção das regras, o que envolve a alteração e testes das regras; e componentes como uma tela de sistema para executar a tarefa de configuração. Junker (2006) acrescenta que um SCP deve prover também um catálogo de componentes – que pode ser representado por um conjunto de restrições – que executarão

filtros e limitarão as opções de configuração a somente os componentes existentes nesse catálogo da companhia.

As restrições podem ser organizadas em camadas, sendo que cada área da companhia pode ser responsável pela sua própria camada de restrições (CONFIGIT, 2020). Exemplos de camadas trazidas pela literatura são marketing, engenharia e manufatura, mas certamente podem ser incorporadas camadas adicionais, como subdivisões nas camadas de engenharia entre domínios diferentes do conhecimento (mecânica, elétrica, química, entre outros), além de uma camada específica para *Compliance* (HASELEIN, 2018). Exemplos podem ser:

- i. Marketing: no mercado asiático, deve-se vender a linha de produto C.
- ii. Engenharia: componente A deve ser montável ao componente B.
- iii. Manufatura: não pode ser fabricado um produto com mais de 1000kg em determinada planta, devido às pontes rolantes existentes.
- iv. *Compliance*: determinado certificado é exigido para que o produto entre no mercado europeu.

É importante ressaltar que definições de marketing podem não ser exatamente restrições, pelo menos não no mesmo sentido das demais camadas que realmente impedem o produto de ser fabricado, podendo ser entendidas mais como uma preferência estratégica; nesse caso, podem se aplicar os conceitos de restrições *soft* discutidos na seção 3.3.3, e é importante que o *framework* de configuração utilizado suporte essa questão.

Trentin, Perin e Forza (2013) sustentam que configuradores modernos precisam oferecer alguns atributos adicionais do ponto de vista do usuário. Entre eles, a possibilidade de escolher facilmente os atributos de um produto, ao invés de escolher diretamente os produtos; usuários iniciantes devem possuir menos opções e campos para serem alterados, enquanto que usuários mais avançados devem conseguir alterar parâmetros mais técnicos do produto; preços e tempos de fabricação também devem estar entre as informações apresentadas.

Sistemas disponíveis de forma online também são essenciais para companhias (SANDRIN et al., 2017). Nesse caso, a interação humana pode ser completamente suprimida e o cliente acessar diretamente o espaço de solução da empresa de forma anônima, solicitando o produto de forma remota.

3.4.3 Relação dos configuradores de produto com a inteligência artificial e sistemas especialistas

Trentin, Perin e Forza (2013) também sustentam que os configuradores de produto são, de fato, sistemas especialistas. Conforme discutido na seção 3.4, a tarefa dos SCP é traduzir necessidades de cliente em um produto manufaturável segundo o espaço de solução da companhia. Tal tarefa pode ser atribuída a especialistas em vendas, que conhecem os catálogos e linhas de produto da empresa, antes de existir um sistema que pudesse fazer isso.

Com a existência de um sistema, a execução dessa tarefa pode ser aprimorada. Em um primeiro momento, podem-se utilizar as capacidades de modelagem e representação de um configurador de produto para organizar as informações de um produto de forma hierárquica e inteligente. Dessa forma, o especialista em vendas ganha produtividade, ao não precisar mais pesquisar ou memorizar muitas informações e detalhes sobre os produtos. Em um segundo momento, o configurador pode evoluir a ponto de tornar-se utilizável diretamente pelo cliente final, substituindo completamente a inteligência humana no processo de venda de um produto customizado.

O conceito *model-based*, também largamente empregado na literatura de sistemas especialistas (HOTZ et al., 2014) é um avanço importante entre a primeira e a segunda geração de configuradores de produto. Ao haver uma separação entre conhecimento e resolução do problema de configuração e com a utilização de uma máquina de inferência para a resolução, o engenheiro de conhecimento pode concentrar-se em efetivamente extrair e codificar o conhecimento, sem precisar se preocupar com a sequência de execução das regras; esse problema ocorria na primeira geração de configuradores, e também em implementações leigas de sistemas especialistas sem a utilização de um sistema *shell*.

Um segundo conceito é a orientação a objetos. Fazendo uso de uma *shell* que possua suporte, é possível modelar o produto em uma estrutura com diferentes classes e níveis, utilizando-se os conceitos explorados sobre modularidade. Dessa maneira, o produto pode ser dividido em montagens, sub-montagens e componentes, cada qual com uma classe correspondente.

Explorou-se ao longo deste capítulo a área de sistemas de configuração de produto, bem como tópicos mais abrangentes e outros mais específicos. Entre tópicos mais abrangentes, está a área de Inteligência Artificial, que tem com uma das sub-áreas sistemas especialistas. Também se discutiu sobre linguagens de programação para a construção de

sistemas especialistas e o paradigma de programação por restrições, que pode ser uma ferramenta importante na construção de novos sistemas especialistas.

A área de sistemas de configuração de produto foi considerada de forma mais aprofundada, apresentando-se um panorama geral do tópico e a existência de diferentes gerações de sistemas.

No próximo capítulo, explorar-se-á o desenvolvimento do protótipo de um sistema de configuração, a fim de atingir os objetivos iniciais do trabalho.

4 DESENVOLVIMENTO DO PROTÓTIPO DO SISTEMA DE CONFIGURAÇÃO DE MOTORES ELÉTRICOS

Sistemas para configuração de produtos técnicos complexos, incluindo ou não o paradigma de programação por restrições e aplicados a cenários reais em indústrias específicas não são uma novidade na literatura. O presente trabalho, no entanto, visa empregar os passos para construção de um sistema especialista abordados na seção 3.2.1 (WATERMAN 1986 apud HASELEIN, 2018), estabelecendo um método de construção de um sistema especialista de configuração. Para avaliar esse método, visa-se explorar um protótipo de um sistema de configuração na empresa-alvo, aplicando-o a um dos principais produtos dentro do catálogo de opções da companhia – os motores elétricos industriais.

No presente capítulo, enfoca-se nas etapas descritas de modelagem do conhecimento por Waterman (1986 apud HASELEIN, 2018), abrangendo as etapas de identificação, conceitualização, formulação e implementação de um primeiro ciclo na metodologia incremental previamente discutida, que se traduz em um sistema minimamente utilizável – ou *MVP (minimum viable product)*.

4.1 CONSTRUÇÃO DE UM SISTEMA ESPECIALISTA DE CONFIGURAÇÃO

O método de construção proposto se baseia nas quatro fases de construção de sistemas especialistas de Waterman (1986 apud HASELEIN, 2018), mas as desdobra em sub-fases para a construção do sistema. Essas etapas são brevemente declaradas a seguir, mas serão melhor detalhadas e justificadas na próxima seção do trabalho. A decomposição é, portanto, a seguinte:

(1) Identificação do problema

- (a) Produto-alvo: consiste em definir qual produto será utilizado para construção do SCP.
- (b) Fontes de variação no produto: consiste em identificar os principais requisitos do cliente e como eles impactam os atributos do produto.
- (c) Funcionamento do configurador de produto no fluxo atual: caso exista um sistema em uso, é necessário identificar seu funcionamento para poder replicá-lo ou aprimorá-lo.

(2) Conceitualização

- (a) Arquitetura do sistema de configuração: consiste em esboçar como funcionará o sistema proposto.
- (3) Formulação
- (a) Estrutura do produto: consiste em elaborar uma estrutura hierárquica que represente o produto
 - (b) Definição dos valores disponíveis na configuração do produto: consiste em estabelecer a maneira de disponibilização dos valores disponíveis de cada atributo.
 - (c) Definição dos valores padrões na configuração do produto: consiste em estabelecer o mecanismo de definição dos valores padrões do SCP.
 - (d) Verificação de inconsistências na configuração e mecanismo de explicações: consiste em estabelecer o mecanismo de inconsistências e explicações do SCP.
- (4) Implementação
- (a) *Shell* utilizada: consiste em escolher uma *shell* apropriada para o SCP.
 - (b) Implementação da estrutura de módulos: consiste em implementar a estrutura de produto previamente formulada.
 - (c) Implementação dos domínios: consiste em implementar os valores dos domínios de cada atributo da estrutura.
 - (d) Implementação dos valores disponíveis: consiste em implementar o mecanismo de definição dos valores disponíveis, incluindo restrições.
 - (e) Implementação dos valores padrões: consiste em implementar o mecanismo de definição dos valores padrões.
 - (f) Implementação das consistências e mecanismo de explicações: consiste em implementar o mecanismo que valida consistências e provê explicações em caso de conflitos.

Na seção seguinte, começar-se-á a explorar o método enunciado acima, podendo haver algumas seções extras que visam dar contexto ao estudo explorado. Por exemplo, após a seção sobre o produto-alvo, será feita uma breve discussão a respeito de motores elétricos, para melhor embasar as decisões tomadas na sequência.

4.2 IDENTIFICAÇÃO DO PROBLEMA

4.2.1 Produto-alvo

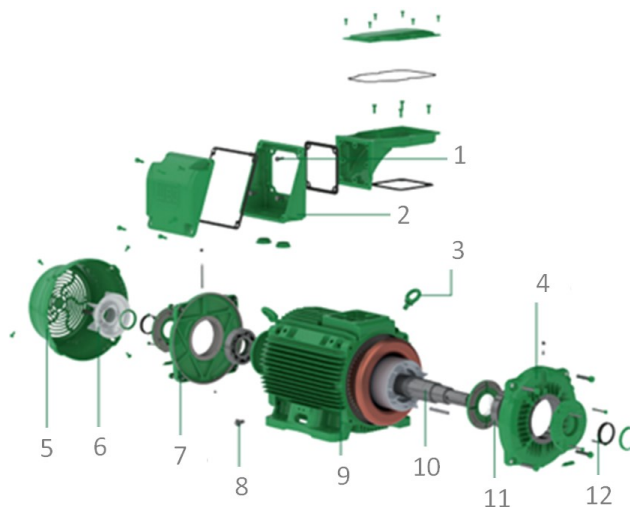
A definição do produto-alvo é necessária para delimitar o escopo do presente estudo, visto que a empresa-alvo possui uma enorme gama de soluções e linhas de produto para os mais variados mercados, com mais de 17 segmentos atendidos (WEG, 2021). Por esse motivo, optar-se-á pela linha de motores elétricos W22 para realizar o estudo, devido à grande flexibilidade de customização.

A linha W22 é bastante estratégica para a empresa-alvo, perfazendo relevante parcela do faturamento anual. Além disso, é uma linha de produtos com elevado potencial de customização.

4.2.2 Uma introdução a motores elétricos

O motor elétrico é um tipo de máquina que converte energia elétrica em energia mecânica (VUKOSAVIC, 2013). Existe um grande número de topologias para esse tipo de máquina, desde o princípio de funcionamento (indução, relutância, etc...) ao campo de aplicação (industrial, comercial, residencial). Um motor típico de aplicação industrial na empresa-alvo segue a topologia mostrada na Figura 4.1.

Figura 4.1 – Motor elétrico e seus componentes



Fonte: Adaptado de WEG (2021)

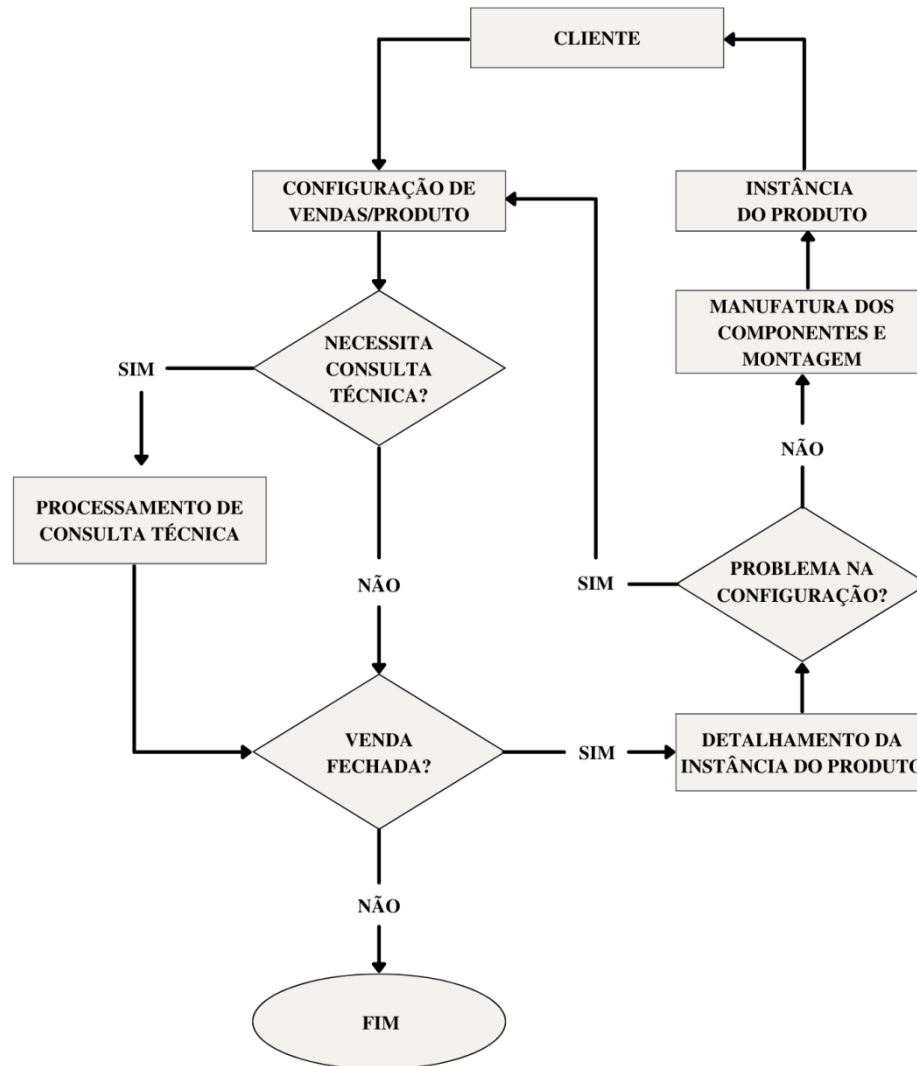
Esse tipo de motor é formado por alguns componentes principais, como a caixa de ligação (1 e 2), olhal (3), tampa dianteira (4), defletora (5), ventilador (6), tampa traseira (7), dreno (8), carcaça com estator bobinado (9), eixo com rotor (10), rolamentos (11) e vedação (12). Esse entendimento sobre os componentes principais do motor será importante para as discussões que se seguirão.

4.2.3 Estratégia de conexão vendas-manufatura

Dentro do processo de customização em massa de motores elétricos, praticamente todos os componentes podem sofrer variações a fim de atender a solicitações dos clientes. Por exemplo, os rolamentos podem ser substituídos para suportar cargas ou rotações maiores do que as rotações de projeto; a ponta do eixo pode ser alterada para se acoplar a uma máquina específica; as vedações podem ser substituídas por outras com desempenho superior quando submetidas a ambientes com pó ou água; o conjunto estator bobinado e rotor podem ser substituídos por outro com maior potência ou maior torque de partida.

O processo de customização é muito similar ao descrito por Lehtonen (2007) e mostrado previamente na Figura 2.7. Por este motivo, adaptou-se esta figura para desdobrá-la nos processos que efetivamente ocorrem na empresa-alvo, ilustrado pela Figura 4.2:

Figura 4.2 – Fluxo de processamento de pedidos na empresa-alvo



Fonte: Autoria própria

Partes desse fluxo foram exploradas em trabalhos prévios como o de Massirer (2007) e Haselein (2017). O processo de consulta técnica foi explorado em Haselein (2017), onde um sistema especialista foi construído para as consultas técnicas do tipo *Compliance*. Outros tipos de consultas técnicas ocorrem na empresa, como consultas às áreas de Engenharia Industrial (para a qual não existe nenhum sistema especialista de apoio) ou Engenharia de Aplicação (para a qual existem alguns algoritmos de cálculo para áreas mecânica e elétrica). Todavia, é importante ressaltar que o processo de consulta por si só é dispendioso e ocorre após a configuração interativa com o cliente, acrescentando tempo de espera do cliente pelo produto.

O processo de detalhamento da instância do produto ocorre em um sistema de estruturação da *bill-of-materials*, e este por sua vez é verificado/consistido por um sistema

especialista proposto por Massirer (2007). O fluxo de processamento de pedidos como um todo não é isento de problemas (conforme explorado na seção 1.1), podendo haver problemas na configuração e/ou consulta técnica que são identificados pelo sistema especialista de Massirer. Nesses casos, é necessário retornar ao processo inicial de configuração e eventualmente interação com o cliente. É oportuno ressaltar que esse retorno ao início do processo evidentemente se traduz em maiores tempos de entrega e insatisfação do cliente.

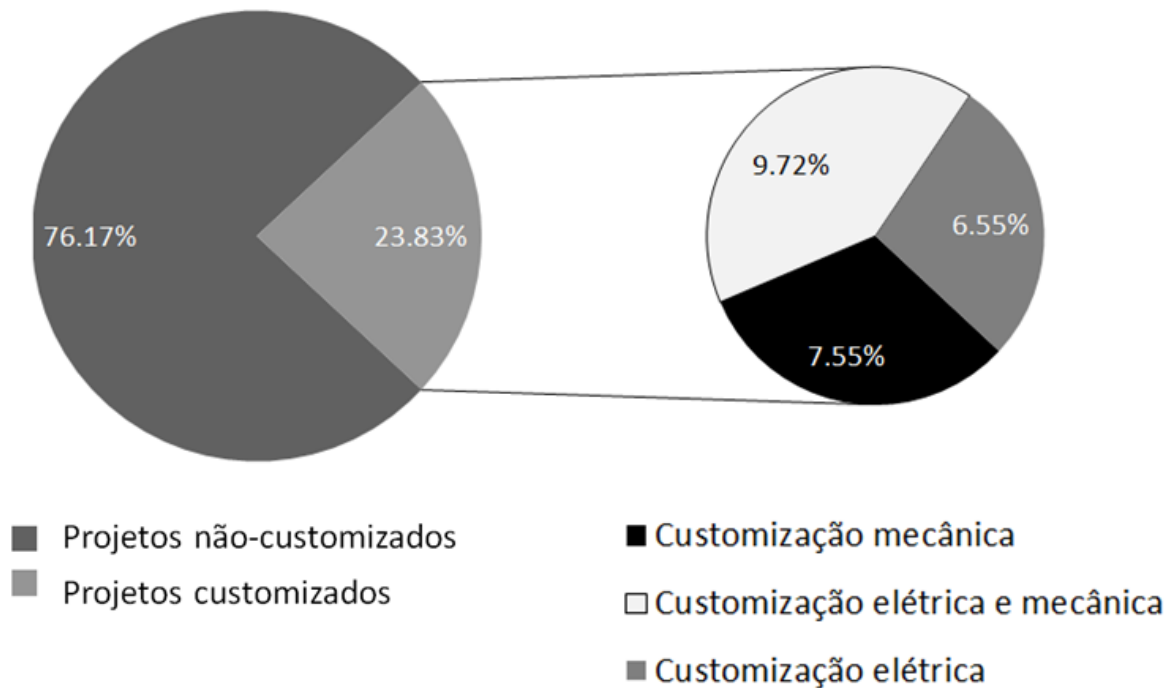
Nesse processo, frequentemente existem interações entre diferentes áreas, mas de maneira geral, pode-se subdividir o processo de projeto customizado na área de projeto elétrico e na área de projeto mecânico. Algumas outras áreas da empresa podem participar em casos específicos como apoiadoras, tais como: Manufatura, Embalagens e *Compliance*.

Considerando Piller e Blazek (2014), conforme explorado na Seção 2.2.6, o fluxo de trabalho da empresa-alvo pode ser caracterizado como uma combinação de ETO e CTO, onde os produtos que não passam por consulta técnica podem ser designados como CTO, de outro modo são ETO.

4.2.4 Fontes de variação no produto

Conforme dados internos da empresa-alvo, o tipo de customização mais frequente em um motor é do subsistema mecânico, mas seguido de perto pelas customizações elétricas. Segundos dados internos, na subdivisão da empresa responsável pela linha de produtos-alvo, foram realizados 21.648 projetos em 2021, o que representa aproximadamente 23,83% do total de projetos vendidos em 2021. Conforme mostrado na Figura 4.3, os projetos customizados dividem-se aproximadamente em 1/3 com customização elétrica, 1/3 com customização mecânica e 1/3 com customização elétrica e mecânica. Conforme fontes internas, o esforço envolvido em customizações elétricas e mecânicas é de similar magnitude.

Figura 4.3 – Total de projetos vendidos em 2021 (à esquerda) e tipos de customizações nos projetos customizados (à direita)



Fonte: Acervo interno da empresa-alvo

Tanto os projetos elétricos customizados quanto os mecânicos seguem o fluxo descrito na Figura 4.2, sendo necessário que a solicitação seja processada por um vendedor especializado, que utiliza o configurador de produto e outras ferramentas de apoio (Consulta técnica); posteriormente, a solicitação especial é detalhada na área de Engenharia de Produto, objetivando-se elaborar a especificação da instância do produto num nível de detalhe em que a fábrica poderá fabricar. Para a parte elétrica, esse processo não será explorado além deste entendimento geral, pois se desvia do escopo do presente trabalho.

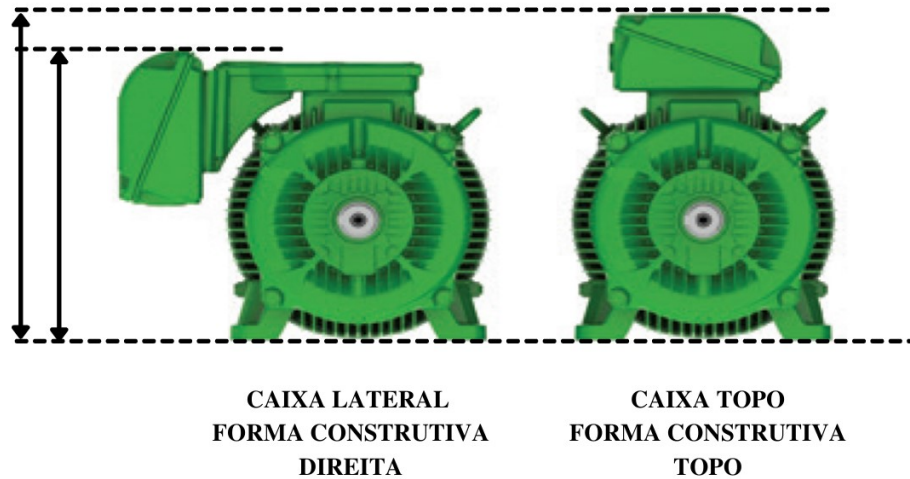
Os principais motivos para alterações na parte mecânica podem ser categorizados nos seguintes pontos principais:

i. Espaço disponível

Aspectos mecânicos são frequentemente relacionados com o espaço disponível para instalação do motor elétrico. Determinadas formas construtivas de motor não podem ser fornecidas devido ao espaço, por exemplo, se a altura disponível é limitada, um motor com a caixa de ligação na parte superior (denominada como “caixa de ligação no topo do motor”) não poderá ser utilizada. Para tal situação, será necessário utilizar uma caixa de ligação posicionada em uma das laterais da carcaça. A Figura 4.4 ressalta a diferença de altura entre

dois motores de mesmo diâmetro de carcaça (e, conseqüentemente, mesma potência), mas diferentes formas construtivas.

Figura 4.4 – Diferença de altura entre motores com formas construtivas distintas



Fonte: Acervo interno da empresa-alvo

ii. Posição de instalação

Outra situação é a posição de instalação do motor, conforme explorado pela Figura 4.5, que ilustra a possibilidade de montar o motor elétrico no chão, teto ou paredes; impactando diretamente nos componentes mecânicos, principalmente drenos e graxeiros.

Figura 4.5 – Diferentes formas de fixação de um motor elétrico

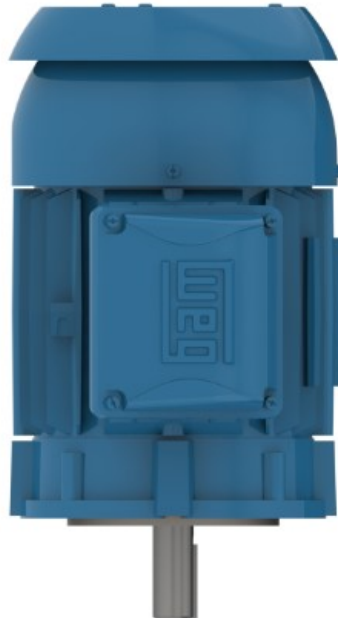


Fonte: Acervo interno da empresa-alvo

Uma possibilidade é realizar a montagem do motor na vertical, com o eixo apontando para cima ou para baixo. A título de exemplo, aplicações relacionadas a bombas

frequentemente exigem que o motor seja montado com o eixo apontando para baixo, conforme mostrado na Figura 4.6.

Figura 4.6 – Motor vertical com flange e disco de proteção



Fonte: Acervo interno da empresa-alvo

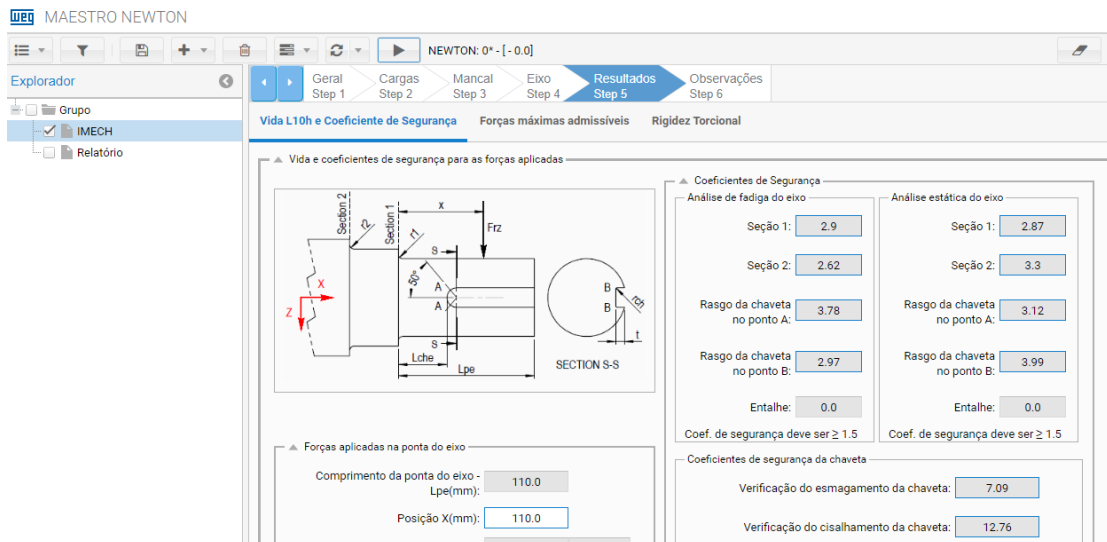
No caso de montagem vertical, devem-se avaliar cuidadosamente os esforços mecânicos nos mancais e tampas, além de assegurar-se de que os mancais serão relubrificadas adequadamente. Também é necessário evitar a entrada de água na defletora, motivo pelo qual se utiliza um disco de proteção afixado na defletora, funcionando como um “chapéu”. Nessas situações, diversos componentes sofrerão alterações, pois precisarão de usinagens extras para permitir o encaixe dos componentes adicionais.

iii. Cargas especiais

Disciplinas clássicas da Engenharia Mecânica estão envolvidas nesse aspecto, como esforços mecânicos, vibrações, fadiga e elementos de máquinas. Frequentemente é necessário dimensionar eixos com diâmetros e comprimentos diferenciados, para atender a necessidade de maior transmissão de potência. Para estes casos, pode ser necessário utilizar rolamentos de série superior, com maior diâmetro ou largura, ou combiná-los em pares. A vantagem é que a repetição desse tipo de problema permitiu o desenvolvimento de uma série de rotinas e programas que sistematizam o dimensionamento. Por exemplo, o sistema *iMech* permite

calcular rapidamente, com apenas alguns cliques do *mouse*, os fatores de segurança para o eixo partir das cargas sendo aplicadas na ponta do eixo, conforme mostrado na Figura 4.7.

Figura 4.7 – Sistema iMech para cálculo de ponta de eixo e rolamentos



Fonte: Acervo interno da empresa-alvo

iv. Acoplamento

Devido à máquina a qual o motor será acoplado, os dimensionais de flanges e furação dos pés dos motores também podem ser avaliados pelos projetistas mecânicos. Nos flanges, normalmente integrados à tampa dianteira do motor, o formato, dimensionais de encaixe e dos furos são importantes; nos pés, a altura do pé até o centro de eixo, distância do encosto do eixo até o primeiro furo do pé e quantidade de furos nos pés são informações essenciais para garantir o encaixe na máquina do cliente.

v. Aspectos térmicos



















O motor elétrico é uma máquina que apresenta perdas na forma de calor – perdas do tipo Joule nos enrolamentos (VUKOSAVIC, 2013), e necessitam de algum meio de troca de calor com o ambiente. A área de transferência de calor é, nesse aspecto, bastante importante para o projeto do motor, mas apesar de estar ligada à Engenharia Mecânica, não será considerada no escopo do trabalho. A razão para isso é que na empresa-alvo, a área de Transferência de Calor está fortemente ligada à Área de Projeto Elétrico, pois são os componentes de natureza elétrica que rejeitam a quase totalidade de calor produzido por um motor elétrico em funcionamento.

4.2.5 Design for Mass Customization (DFMC) no produto-alvo

A linha-alvo de produtos é amplamente customizável e foi projetada com esse objetivo. No catálogo técnico da linha, é possível entender que diversas alterações mecânicas e elétricas podem ser executadas, sendo possível produzir um número muito elevado de variantes. A seguir, serão apresentadas algumas customizações aplicáveis à linha-alvo, restringindo-se a questões mecânicas.

A customização de forma construtiva é uma das mais impactantes, e é facilmente identificável mesmo por alguém sem conhecimento na área. A tabela extraída de WEG (2021), ilustrada pela Figura 4.8, mostra uma miríade de opções de forma construtiva.

Figura 4.8 – Algumas formas construtivas de um motor elétrico

Forma construtiva	Configuração						
	Referência	B3E	B3D	B3T	B5E	B5D	B5T
Detalhes	Carcaça	Com pés	Com pés	Com pés	Sem pés	Sem pés	Sem pés
	Ponta de eixo	À esquerda	À direita	À direita	À esquerda	À direita	À direita
	Fixação	Base ou trilhos	Base ou trilhos	Base ou trilhos	Flange FF	Flange FF	Flange FF
Forma construtiva	Configuração						
	Referência	V5	V5E	V5T	V6	V6E	V6T
Detalhes	Carcaça	Com pés	Com pés	Com pés	Com pés	Com pés	Com pés
	Ponta de eixo	Para baixo	Para baixo	Para baixo	Para cima	Para cima	Para cima
	Fixação	Parede	Parede	Parede	Parede	Parede	Parede
Forma construtiva	Configuração						
	Referência	B14E	B14D	B14T	B6	B6E	B6T
Detalhes	Carcaça	Sem pés	Sem pés	Sem pés	Com pés	Com pés	Com pés
	Ponta de eixo	À esquerda	À direita	À direita	Para frente	Para frente	Para frente
	Fixação	Flange FC	Flange FC	Flange FC	Parede	Parede	Parede

Fonte: Acervo interno da empresa-alvo

De forma geral, a forma construtiva diz respeito à questão de fixação do motor – através de pés na carcaça ou flange; à direção que o eixo assumirá quando montado na máquina (vertical para cima, vertical para baixo, horizontal); à posição da caixa de ligação em relação ao eixo (esquerda, direita, topo); à posição dos pés (se existirem); e onde o motor será afixado (chão, teto, parede).

A forma construtiva padrão é denominada B3D, onde a carcaça possui pés, o motor é fixado no chão, com o eixo na horizontal, e a ponta de eixo está à direita olhando a partir da caixa de ligação. Ao definir uma forma construtiva, diversos componentes mecânicos podem ser impactados em relação à forma construtiva padrão. Por exemplo, quando o motor é montado na vertical com o eixo para baixo, é necessário que exista um dreno na face da tampa dianteira, de modo que a umidade que condensa dentro do motor possa ser removida na direção da gravidade.

Quando o motor é montado em diferentes posições, como no teto ou parede, tanto o dreno quanto o sistema de relubrificação dos rolamentos (graxeira) são impactados, conforme discutido na seção anterior, pois é necessário considerar a direção da gravidade no escoamento da água e da graxa. Esses aspectos foram considerados no desenvolvimento da linha, incorporando boas práticas de DFMC: de maneira geral, a simples montagem das tampas rotacionadas em 90 ou 180 graus atende o requisito de direcionar adequadamente a graxa e a água. Dessa maneira, não é necessário um componente diferente – apenas uma instrução de montagem diferente.

A troca de posição da caixa de ligação também é possível mesmo após o motor ter sido instalado, sendo necessário remover o suporte intermediário e reposicionar a caixa de ligação conforme desejado, como mostrado na Figura 4.9:

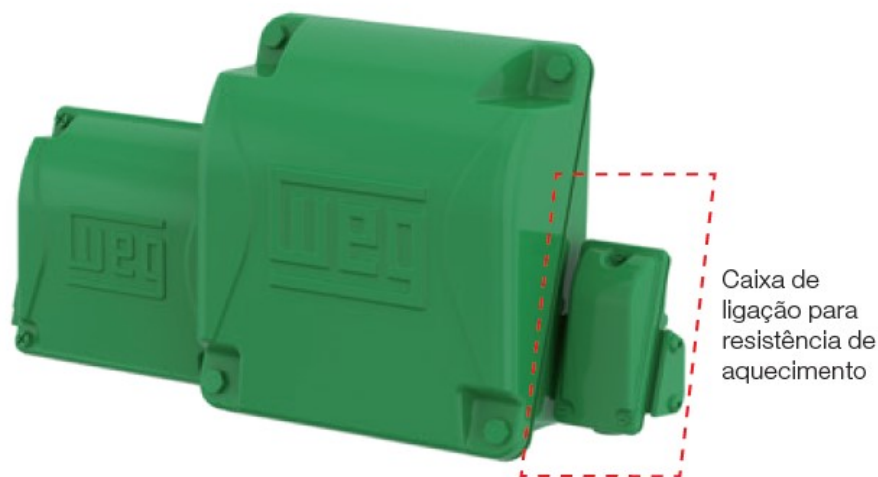


Figura 4.9 – Alteração da posição da caixa de ligação

Fonte: Acervo interno da empresa-alvo

As caixas de ligação também são customizáveis. É possível alterar as furações que existem na parte inferior que servem para conectar os cabos de força. Para casos em que se necessita de mais espaço para placas de bornes, podem ser acrescentadas caixas auxiliares, que são fixadas através de uma furação extra na caixa principal, conforme mostrado na Figura 4.10:

Figura 4.10 – Visão da caixa de ligação principal (ao centro) complementada com caixa auxiliar (à esquerda) e caixa de resistências (à direita)



Fonte: Acervo interno da empresa-alvo

Apesar de ter sido feito um grande esforço em tempo de desenvolvimento da linha de produto para preparar o produto ao máximo para customizações que seriam feitas, a empresa adota uma prática de melhoria contínua, constantemente reavaliando especialidades mais requisitadas e incorporando-as ao produto. Esse processo é denominado internamente de “liberação de opcional”, e pode resultar em alterações de diversos componentes para melhor acomodar as novas customizações permitidas.


4.2.6 Funcionamento do configurador de produto no fluxo atual

A empresa-alvo possui diversos processos distintos para diferentes produtos, mas para o produto-alvo, existe um sistema de configuração de produto dedicado. O dia-a-dia da área comercial consiste em converter especificações e solicitações de clientes em produtos da

empresa, geralmente, recebidos em formato eletrônico (um e-mail, por exemplo), com uma descrição de produtos a serem orçados. A Figura 4.11 ilustra um exemplo dessa solicitação:

Figura 4.11 – Exemplo de pedido de cotação

Motor RFQ – THEMS 330410SR Rev C

-  22-7130
(BR1048-MBH-MEC020-330440PS)
- Qty 3 1 HP(0.75 kW) aux blower motor
- 2 pole, 60Hz motor, 3/60/460V
 - Face mounted
- Qty 3 1/2 HP(0.375 kW) gearbox motor
- 4 pole, 60Hz motor, 3/60/460V
 - Face mounted
- Qty 3 200 HP(150 kW) main motor c/w
- 2P 60 Hz, 3/60/460 V
 - Foot mounted
 - Dual shaft extension
 - Space heater requiring **1/60/127V**

Motors above to be designed to the following:

- IEC Design
- No hazardous area classification required
- Coastal Outdoor installation
- +16 to +40°C ambient conditions
- 0-1000 m installation
- 23-100% RH
- 85 dBA
- Equipment supplied shall be coated per BR1048-ABZ020-357040. using the very high corrosion category C5(M). Equipment shall be provided with the prime coat, intermediate coat, and final Coat.
- For compliance with local Brazilian Codes, NRs shall be followed and complied with BR1048-BEC010-M-00370.
- The electrical requirements for the supplied equipment shall be in accordance BR1048-10221-&EEC010-708101.
 - Junction boxes shall be IP66 and have grounding bars.
 - Motors provided shall be suitable for tropicalized environments.

Fonte: Acervo interno da empresa-alvo

O fluxo de trabalho segue em interpretar esse tipo de especificação e, utilizando-se de conhecimento prévio, acessar o configurador de produto, selecionar uma linha de produto adequada, seguir as recomendações do próprio SCP e salvar a configuração. Na sequência, se o produto for vendido, conforme explorado na Figura 4.2, ele seguirá para as áreas de Engenharia de Produto, onde a configuração será convertida em uma lista técnica completa para depois ser produzido (processo CTO).

No processo ETO, os tempos de entrega aumentam devido à necessidade de avaliação por parte da área de Engenharia de Aplicação. A necessidade de avaliação é controlada pelo próprio configurador de produto, através das regras de consistência e de classificação de valores, que serão discutidas a seguir.

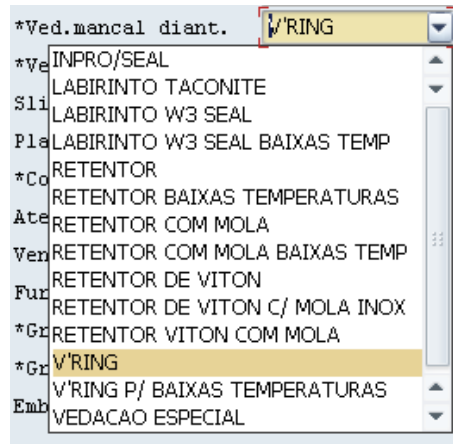
O sistema atual de configuração de produto da empresa-alvo é uma versão customizada, construída e administrada dentro do sistema de gestão da empresa, da empresa alemã SAP. Esse sistema é administrado diretamente dentro do SAP. O configurador do produto-alvo possui 300 atributos e 5 telas de configuração. Para ilustrá-lo, a tela inicial é mostrada na Figura 4.12:

Figura 4.12 – Tela inicial do configurador de produto com as principais divisões

Fonte: Acervo interno da empresa-alvo

Na figura, é possível visualizar a subdivisão da tela em frames (agrupamentos), cujo objetivo é prover algum tipo de organização visual e relacionamento entre os atributos, assim como os atributos propriamente ditas.

No configurador atual, todos os atributos são do tipo caixa de combinação. A Figura 4.13, permite visualizar o atributo “Vedação do Mancal Dianteiro” e parte de suas opções: labirinto Taconite, labirinto W3 Seal, Labirinto W3 Seal Baixas Temperaturas, Retentor, etc., sendo que somente um subconjunto de todas as opções de vedações é mostrado ao usuário.

Figura 4.13 – Atributo com um *combo box*

Fonte: Acervo interno da empresa-alvo

No caso, existem 66 tipos de vedação cadastrados nesse atributo, mas somente 14 valores estão disponíveis para seleção no configurador. Isso, no entanto, não assegura que todas as opções são viáveis, sendo que erros podem surgir na etapa de gravação/consistências do configurador por se tratar de um configurador de primeira geração.

Na utilização do SCP, a área comercial é auxiliada através de três tipos de regras:

I - Regras de dependência: São as regras que definem quais valores são padrões, opcionais ou especiais. Esse conceito é amplamente utilizado na companhia. Valores padrões são valores, como o próprio nome sugere, selecionados automaticamente pelo configurador de produto. Pode-se compreendê-los como valores selecionados pela camada de marketing, discutidos na seção 3.4.2. Dessa maneira, para casos mais simples, pode-se configurar apenas alguns atributos iniciais e os demais estarão preenchidos com valores padrão, não sendo necessário definir todos de forma individual. Os valores opcionais são valores que estão disponíveis para seleção e já foram realizados projetos de praticamente todas as combinações de componentes possíveis. Dessa maneira, ao vender um produto que possui um ou mais atributos com valores opcionais, esse produto passará por um detalhamento automático da instância do produto, não sendo necessário envolvimento de um projetista. Pode-se compreendê-los como valores restritos pelas camadas de engenharia e/ou manufatura. Por último, tem-se os atributos com valores especiais; estes são valores que não foram previamente analisados no momento de desenvolvimento do produto e portanto, não se sabe quais componentes serão afetados pela seleção nem quais serão os demais impactos no produto. Dessa maneira, será necessário que esse projeto seja analisado por um projetista.

Essas regras são construídas com cláusulas do tipo SE-ENTÃO, dispostas em tabelas, conforme mostrado nas Figura 4.14 e Figura 4.15 a seguir:

Figura 4.14 – Condição SE da regra de dependência 6217056

id.val.reg	Rev.valid.	Nome característica	Val.carac.	Oper.Lóg.
6217056	10	ZAPLICACAO_MOTOR_01	00082	EQ
6217056	10	ZCARACA_COM_COMPLEMENTO_01	00953	EQ
6217056	10	ZDETALHE1_01	00009	EQ
6217056	10	ZDETALHE1_01	00011	EQ
6217056	10	ZDETALHE1_01	00058	EQ
6217056	10	ZFASE_01	00012	EQ
6217056	10	ZFREQUENCIA_01	00064	EQ
6217056	10	ZFREQUENCIA_01	00079	EQ
6217056	10	ZINVOLUCRO_01	00003	EQ
6217056	10	ZMATERIAL_INVOLUCRO_01	00004	EQ
6217056	10	ZMERCADO_01	00010	EQ
6217056	10	ZPOLARIDADE_COMPLETA_01	00001	EQ
6217056	10	ZPOLARIDADE_COMPLETA_01	00003	EQ
6217056	10	ZPOLARIDADE_COMPLETA_01	00004	EQ
6217056	10	ZPOLARIDADE_COMPLETA_01	00009	EQ
6217056	10	ZPOLARIDADE_COMPLETA_01	00010	EQ
6217056	10	ZPOLARIDADE_COMPLETA_01	00011	EQ
6217056	10	ZPOLARIDADE_COMPLETA_01	00012	EQ
6217056	10	ZPOLARIDADE_COMPLETA_01	00016	EQ
6217056	10	ZPOLARIDADE_COMPLETA_01	00018	EQ

Fonte: Acervo interno da empresa-alvo

Figura 4.15 – Condição ENTÃO da regra de dependência 6217056

id.val.reg	Rev.valid.	Nome característica	Val.carac.	Id.caract.
6217056	10	ZCHAPEU_01	00001	O
6217056	10	ZCHAPEU_01	00002	P
6217056	10	ZFORMA_CONSTRUTIVA_MERCADO_01	00003	O
6217056	10	ZFORMA_CONSTRUTIVA_MERCADO_01	00007	O
6217056	10	ZFORMA_CONSTRUTIVA_MERCADO_01	00008	O
6217056	10	ZFORMA_CONSTRUTIVA_MERCADO_01	00011	O
6217056	10	ZFORMA_CONSTRUTIVA_MERCADO_01	00015	O
6217056	10	ZFORMA_CONSTRUTIVA_MERCADO_01	00016	P
6217056	10	ZFORMA_CONSTRUTIVA_MERCADO_01	00028	O
6217056	10	ZFORMA_CONSTRUTIVA_MERCADO_01	00032	O
6217056	10	ZFORMA_CONSTRUTIVA_MERCADO_01	00033	O
6217056	10	ZFORMA_CONSTRUTIVA_MERCADO_01	00036	O
6217056	10	ZFORMA_CONSTRUTIVA_MERCADO_01	00040	O
6217056	10	ZFORMA_CONSTRUTIVA_MERCADO_01	00041	O
6217056	10	ZFORMA_CONSTRUTIVA_MERCADO_01	00044	O
6217056	10	ZFORMA_CONSTRUTIVA_MERCADO_01	00048	O
6217056	10	ZFORMA_CONSTRUTIVA_MERCADO_01	00049	O
6217056	10	ZFORMA_CONSTRUTIVA_MERCADO_01	00052	O

Fonte: Acervo interno da empresa-alvo

Como é perceptível, mesmo uma única regra ocupa uma grande quantidade de linhas, contribuindo para formar uma base com mais de 1 milhão de linhas, como exposto na introdução do presente trabalho.

É importante ressaltar que, tanto na Figura 4.14 como na Figura 4.15, vale a seguinte lógica: quando se tratam de duas linhas consecutivas com a mesma característica, será feito uma operação booleana OU entre as mesmas; enquanto se forem características diferentes, será aplicada a operação booleana E.

Portanto, essas regras são lidas da seguinte forma: SE ZAPLICACAO_MOTOR_01 = 00082 E ZCARCACA_COM_COMPLEMENTO_01 = 00953 E ZDETALHE1_01 = 0009 OU 00011 OU 00058... ENTÃO ZCHAPEU_01 = 00001 é um valor opcional, ZCHAPEU_01 = 00002 é um valor padrão, e assim sucessivamente. Percebe-se que os valores foram substituídos por códigos, para viabilizar o requisito de sistema multi-idiomias.

II - Regras de consistência: Esse tipo de regra é executado somente ao final da configuração, no momento em que a configuração é gravada. Tais regras fazem as verificações mais abrangentes possíveis, funcionando num formato SE-ENTÃO. Esse processamento de consistência leva por volta de 12 segundos no sistema atual, devido à grande quantidade de regras existentes. Por exemplo, uma determinada regra pode fazer uma consistência, conforme ilustrado pela Figura 4.16 a seguir:

Figura 4.16 – Exemplo de regra de consistência

Motivo: CLIENTE ZIEHL-ABEGG CLAIM, 417665786

Característica determinante

Característica	Valor Característica	Operador Lógico	Valor Característi...	Classificação
ZMERCADO_01 MERCADO	00010 EUROPA	EQ		Qualquer Tipo
ZAPLICACAO_MOTOR_01 APLICACAO MOTOR	00082 W22	EQ		Qualquer Tipo

Opção Contexto: 2 Pelo menos 1

Opção Restrição: 2 Obrigatório todos

Característica	Valor Característica
ZPLACA_PRINCIPAL_CLIENTE_01 PLACA P...	00185 ZIEHL-ABEGG

Característica	Valor Característica
ZCERTIFICACAO_01 CERTIFICACAO	00317 CE/EAC

Fonte: Acervo interno da empresa-alvo

Para a regra acima, será mostrada a seguinte mensagem para o usuário: PARA O MERCADO EUROPA NA LINHA W22 COM CLIENTE ZIEHL-ABEGG, É OBRIGATÓRIO CONFIGURAR CERTIFICAÇÃO CE/EAC.

Devido às limitações do atual sistema de configuração, a cláusula ENTÃO somente pode ser uma mensagem para o usuário, classificada em dois tipos: aviso ou erro. A diferença é que para a mensagem do tipo erro, o usuário obrigatoriamente terá de corrigir o problema para que a gravação da configuração seja habilitada.

Conforme detalhado na seção 3.4.1, trata-se de um sistema de configuração de primeira geração, pois é necessário realizar a configuração de forma sequencial. Devido à estrutura das tabelas que fazem o filtro dos atributos, é necessário fazer uma seleção inicial de 13 atributos, que são dispostos no início do configurador conforme mostrado na Figura 4.17 a seguir:

Figura 4.17 – Atributos iniciais do configurador de produto

DADOS GERAIS		PRINCIPAIS OPCIONAIS MECÂNICOS		PRINCIPAIS OPCIONAIS ELÉTRICOS	
Mercado	BRASIL	*Forma construtiva	B3D	Fator de serviço	1.25
Alimentação	TRIFASICO	Forma construt. WEG	B3D	Classe de isolamento	F (DT=80K)
Invólucro	FECHADO	Forma construt. NEMA		Resistência de aque...	SEM RESISTEN...
Linha	W22	*Flange	SEM FLANGE	*Proteção térmica a...	SEM PROTECA...
Variante de linha	SEM VARIANTE	Chapéu	NAO	*Proteção térmica d...	SEM PROTECA...
Material invólucro	FERRO	Placa de bornes	SEM	Categoria/design	N
Frequência	60Hz	Grau de proteção	IPW55	OPC CAIXA LIGAÇÃO	
Norma	IEC	*Ved.mancal diant.	V/RING	Tipo caixa	W22
Detalhe 1	NENHUM	*Ved.mancal tras.(d...	SEM VEDACAO...	Rosca caixa princip...	RWG(Rp) 2"
Detalhe 2	IE3	Slinger	NAO	Rosca caixa adicion...	NAO APLICAVEL
Tipo de polaridade	UNICA	Plano de pintura	203A	Rosca caixa resiste...	NAO APLICAVEL
Potência	40 cv	*Cor	RAL 5009	Frensa cabos	SEM
Polos	04	Aterramento	ATERRAMENT...	Tampão	ROSCADO PLA...
Carcaça	200M	Ventilador	PLASTICO	Massa epóxi	SEM MASSA EP...
Carcaça comercial	200M	Furo de centro rosc...	NAO	Saída de cabos da c...	P/ BAIXO OLI...
*Tensão com cabo	220/380 C/ 6 t...	*Graxa	MOBIL POLYRE...	Diâmetro furo cx lig	
Tensão sem cabo	220/380V	*Graxeira	SEM GRAXEIRA	Pintura interna cai...	
Número de cabos lig...	6	Embalagem	ENGRADADO	Caixa de ligação es...	
*Tipo produto	OPCIONAL				

Fonte: Acervo interno da empresa-alvo

Para que os demais atributos sejam preenchidos com valores padrão e tenham seus domínios minimamente filtrados, é necessário realizar essa seleção inicial. Como discutido previamente, não é possível realizar as seleções desses atributos iniciais numa ordem diferente da prevista - um problema clássico dos sistemas de configuração de primeira geração.

Por exemplo, só existe um procedimento para descobrir qual linha atende determinada certificação: através de tentativa-e-erro. Isto é, o usuário configura uma linha e verifica se a certificação desejada está entre as disponíveis; caso não esteja, retorna ao atributo da linha, tenta outro valor e repete o processo. Na prática, o especialista em vendas acaba por memorizar as situações mais comuns, mas esse tipo de situação não deveria ocorrer com um sistema especialista, indo contra um preceito básico que é não depender do conhecimento de

um especialista humano para realizar a tarefa. Nesse tipo de cenário, a informação de certificação por linha está codificada nas regras, mas é inacessível na ordem reversa de seleção. Tal problema não ocorre se utilizado o paradigma de programação por restrições.

No SCP em uso na empresa-alvo, não há uma modelagem mais completa do produto, faltando importantes informações para construção de regras. De maneira geral, existem poucos atributos além dos que são mostradas para o usuário. Consistências de projeto mecânico, exploradas no trabalho de Massirer (2007), por exemplo, o encaixe de rolamento na tampa comparando-se o diâmetro externo do rolamento com o diâmetro interno do cubo na tampa, não são possíveis de serem feitas. Esse diagnóstico leva a conclusão da necessidade de uma modelagem detalhada do produto envolvido e construção de restrições no sistema de configuração de produto. Dessa maneira, evita-se que problemas do produto sejam identificados tardiamente.

4.3 CONCEITUALIZAÇÃO

4.3.1 Conceito geral

O protótipo de configurador de produto a ser desenvolvido no presente trabalho utilizará a *shell* internamente desenvolvida pela empresa-alvo, denominada CBConf, que será explorada de forma mais aprofundada na seção 4.5, abandonando a *shell* antiga construída dentro do SAP. Essa troca tecnológica não resultará, pelo menos como proposta pelo presente trabalho, em uma alteração no fluxo de trabalho descrito anteriormente. Os ganhos de um processo renovado advirão da implementação de um novo sistema com tecnologias atualizadas, descontinuando o sistema antigo com suas limitações e incorporando conhecimento de especialistas durante a implementação do novo sistema de forma mais abrangente e conceitualmente adequada.

Alguns problemas do sistema atual são resolvidos com a troca da tecnologia da *shell*, isto é, substituir um configurador de primeira geração por um de segunda geração. Esse aspecto permite que certas situações relatadas pelos especialistas no domínio sejam simplificadas. Por exemplo: caso o cliente desejasse um produto com uma determinada certificação que só existe em uma linha de produto, o vendedor tinha de saber *a priori* qual linha de produto está apta para aquela certificação. Como o configurador atual funciona de forma unidirecional e esta regra relaciona linha de produto → certificação, um vendedor

inexperiente (ou o próprio cliente final) só conseguiria descobrir qual linha está apta a uma dada certificação através da tentativa e erro. No sistema baseado em restrições, esse problema inexistente. Mesmo um vendedor experiente pode selecionar o campo “certificações” e verificar quais linhas são atendidas de forma imediata, sem valer-se da tentativa e erro.

Outro aspecto relevante a ser discutido é o aspecto exageradamente técnico dos atributos principais da configuração. A solução é uma modelagem mais completa da estrutura de produto, descartando a antiga estrutura de nível único para representar o produto em uma estrutura multinível. Também se ressalta a importância de haver um nó para atributos de aplicação do produto (características não-físicas, por exemplo, o grau de proteção requerido). Dessa maneira, resolve-se outro tipo de problema que ocorre com o vendedor experiente/cliente final: frequentemente, não se conhece o portfólio completo de produtos da empresa, e em alguns casos pode ser difícil determinar qual linha de produto deve ser utilizada. Caso exista um nó de atributos de aplicação, que são mais compreensíveis por um usuário menos experiente, ele pode utilizar esses atributos para filtrar a linha de produto mais adequada.

Os dois casos discutidos anteriormente são problemas que técnicas de Engenharia de Conhecimento procuram solucionar, tipicamente revelando uma incompletude da modelagem do conhecimento corporativo em sistemas que são utilizados pelos colaboradores e clientes. Não à toa, atualmente na empresa-alvo a tradução dos requisitos do produto se torna uma tarefa somente executável por um especialista em Vendas, que tem vasto conhecimento do catálogo de produtos da companhia e que consegue operar o Sistema de Configuração de Produto. No entanto, como discutido no presente trabalho, é possível adotar uma modelagem dos requisitos de cliente *versus* atributos técnicos de configuração. Outros ganhos acontecerão dentro da própria empresa, pois os produtos passam a estar validados de forma mais abrangente do que no processo atual já na etapa de cotação, reduzindo a necessidade de regras de validação, como as exploradas por Massirer (2007), além de padronizar ações entre diferentes especialistas das áreas comerciais. Esses ganhos podem se transformar em maior eficiência interna, com a possibilidade de redução de tamanhos de lotes, menores quantidades de SKU, trazendo resultados quantitativos para a companhia.

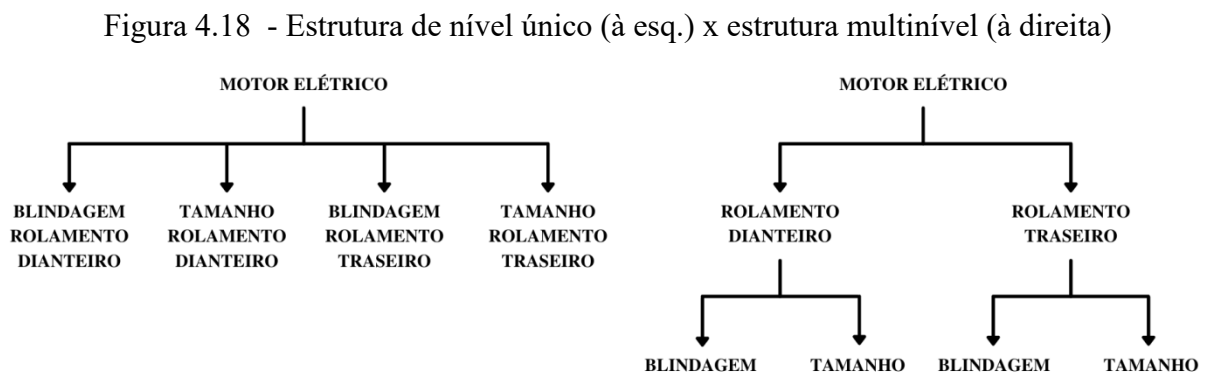
4.3.2 Arquitetura do sistema de configuração

Na pesquisa de referências realizada, identificaram-se alguns pontos principais que podem compor um *método* de funcionamento de um sistema de configuração, descritos abaixo:

- i. Organização dos atributos do produto;

O aspecto de organização dos atributos do produto diz respeito a como o produto será modelado, isto é, como são representadas suas partes relevantes – os módulos – quais atributos serão considerados e quais valores cada atributo contém. Nesse ponto de vista, cabe um direcionamento geral para o propósito do configurador de produto. Se a companhia objetiva um configurador de mais alto nível, utilizado apenas por usuários leigos, não haverá necessidade de uma grande quantidade de atributos; por outro lado, se a finalidade for utilizá-lo para níveis de engenharia, provavelmente haverá um nível de detalhamento maior do produto. Na empresa-alvo, inexistente uma organização em níveis dos atributos citados. Há somente um nível – o produto “motor elétrico” – e todos os atributos estão alocadas nele.

Esse tipo de estrutura de nível único traz problemas principalmente quando a quantidade de atributos é elevada, e também quando existem múltiplas instâncias de uma mesma entidade em diferentes posições do produto. Por exemplo, no motor elétrico, existem dois rolamentos, o dianteiro e o traseiro. Na estrutura de nível único, haverá, portanto, atributos muito similares, apenas com uma diferenciação entre dianteiro e traseiro, tal como, “blindagem do rolamento dianteiro”, “blindagem do rolamento traseiro”, “tamanho do rolamento dianteiro”, “tamanho do rolamento traseiro”, e assim sucessivamente. Explicita-se a diferença entre as estruturas de nível único ou múltiplo na Figura 4.18, mostrada a seguir:

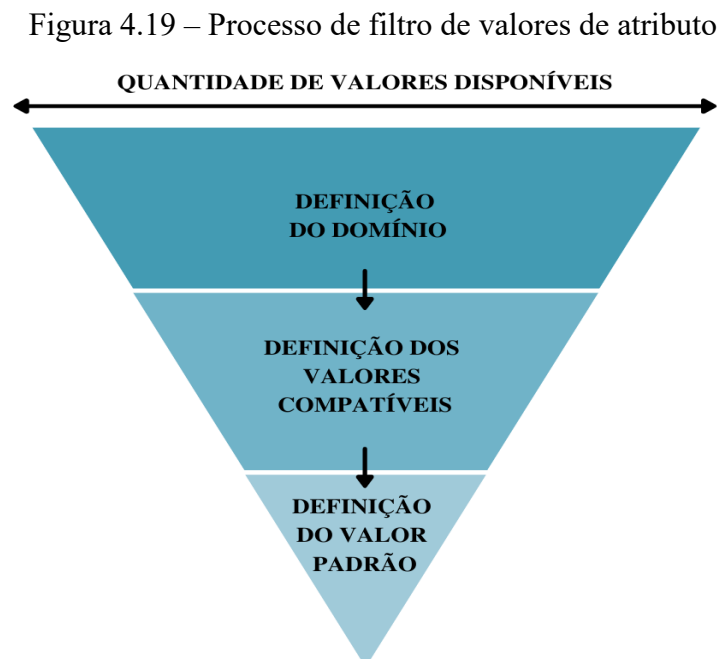


Fonte: Autoria própria

Em uma estrutura multinível, reduz-se a quantidade de domínios de atributos, pois nesse caso seria possível modelar tais atributos como sendo um endereço nessa estrutura acrescida de um atributo reaproveitável. Tem-se, portanto, “rolamento dianteiro → blindagem”, “rolamento dianteiro → tamanho”, “rolamento traseiro → blindagem” e “rolamento traseiro → tamanho”. Nesse exemplo, passa-se a ter somente dois atributos ao invés de quatro.

ii. Filtragem dos valores dos atributos;

Na empresa-alvo, identificou-se que o aspecto de definição de valores disponíveis de cada atributo é controlado por um mecanismo único denominado “regras de dependência”. No entanto, para realizar um controle mais aprimorado, optou-se por separar esse processo. A proposta do processo é representada a seguir na Figura 4.19.



Fonte: Autoria própria

A primeira etapa é definir todos os valores disponíveis – isto é, a definição do domínio; a segunda etapa é filtrar, dentre todos os valores disponíveis, quais são compatíveis com uma configuração que está sendo executada nesse momento; e a terceira etapa é a definição de qual dos valores compatíveis é o valor padrão.

Essa primeira etapa é basicamente um cadastro associado a cada atributo, mantida por especialistas da empresa-alvo. Nesse ponto, percebe-se um ganho da estrutura multinível

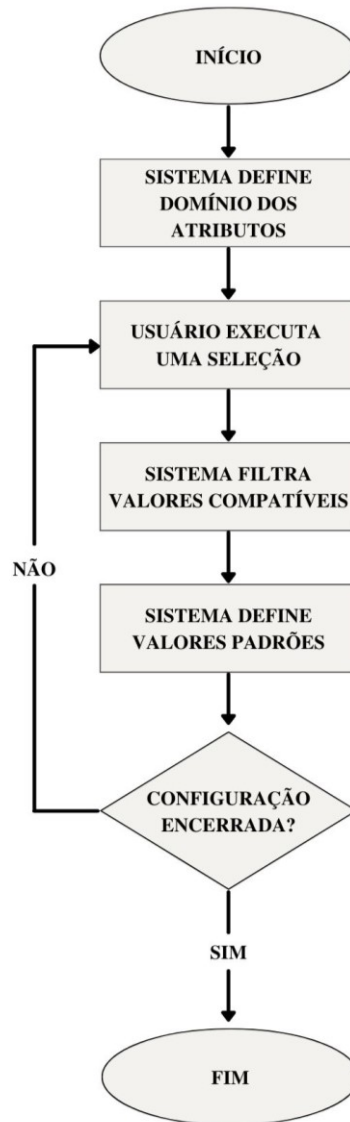
ao reduzir a quantidade de atributos a serem mantidos. Já a segunda etapa depende de um mecanismo de regras que realize a filtragem dos valores, eliminando valores incompatíveis, podendo ser em formato de regras ou tabelas bidirecionais.

A etapa a seguir é a definição de valores padrões. Esse aspecto não é imprescindível ao funcionamento de um SCP, mas acelera o processo de configuração. Também é necessário um mecanismo de regras, pois é preciso criar relações entre os atributos do produto. Ao possuir valores padrões para boa parte dos atributos, com apenas algumas definições pode-se obter um produto completo configurado, independente da complexidade do produto. No caso em estudo, a definição inicial de 13 atributos permite obter um produto completo, mesmo com o configurador contendo mais de 400 atributos.

Na empresa-alvo foi identificado um mecanismo de consistências executadas ao final da configuração, mas estas somente existem devido às limitações do mecanismo de regras de filtragem dos valores disponíveis e devido à modelagem incompleta do produto. Um sistema baseado em restrições não precisará de uma execução ao final da configuração, o que pode resultar na incômoda situação de tentativa-e-erro, pois todas as consistências podem ser executadas durante a configuração e não apenas ao final dela.

Conectando os pontos discutidos acima, tem-se um fluxo de funcionamento conforme indicado pela Figura 4.20:

Figura 4.20 – Fluxo proposto de funcionamento do novo sistema



Fonte: Autoria própria

Nesse fluxo de interação usuário-sistema, o sistema inicialmente carrega o domínio de todos os atributos. Na sequência, inicia-se o ciclo de interação, o usuário executa uma seleção de valor em um atributo, o sistema recalcula todos os valores compatíveis e padrões e mostra as atualizações para o usuário; este pode decidir por encerrar a configuração ou continuar modificando outros atributos. Conforme já discutido, não há etapa de consistências após a configuração ser encerrada.

4.4 FORMULAÇÃO

Na etapa de formulação, objetiva-se aplicar os conceitos discutidos na seção de Conceitualização ao domínio específico do problema. Por exemplo, discutir-se-á como ficará a estrutura de módulos no domínio do motor elétrico. Essa etapa permite atingir um maior nível de detalhamento na proposta, permitindo encontrar eventuais problemas nos conceitos. O objetivo é ter todas as informações necessárias para avançar, na sequência, para a implementação (WATERMAN 1986 apud HASELEIN, 2018).

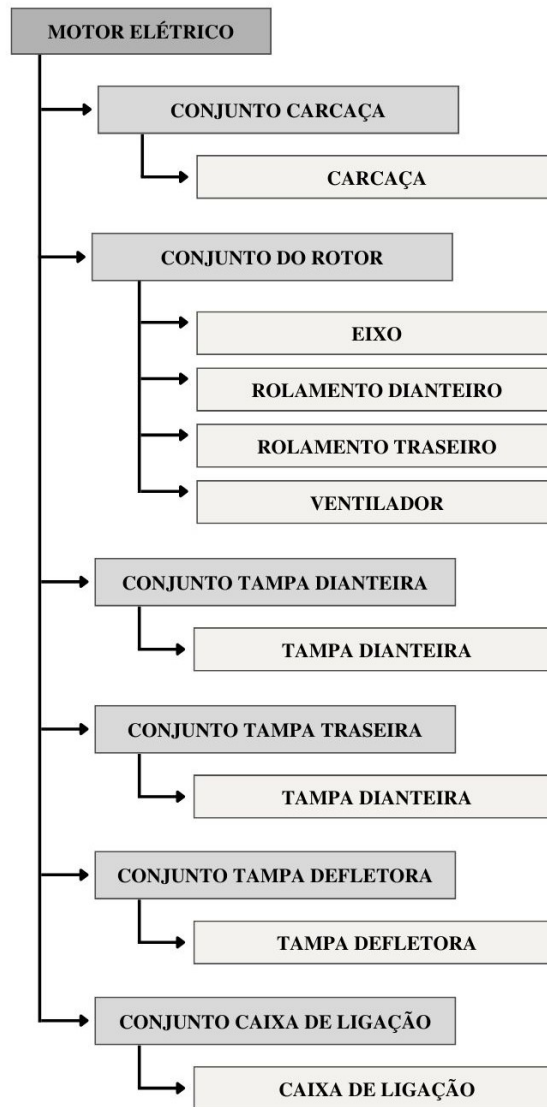
4.4.1 Estrutura do produto

No novo sistema de configuração de produto, será modelada uma estrutura multinível para representação do produto, derivada da *BOM* do motor elétrico (explorada por Massirer, 2007). No presente estudo, propõe-se uma *BOM* simplificada que atende os fins de configuração de produto, podendo ser classificada como uma *E-BOM*. A ideia de utilizar uma *E-BOM* no processo de configuração é discutida por Configit (2020). Os componentes dessa estrutura são agrupados em conjuntos, permitindo a estruturação em níveis.

O conceito de outro tipo de *BOM* além da *M-BOM* inexistente atualmente na empresa-alvo. Entretanto, outras empresas podem ter conceitos diferentes a respeito desse tema, inclusive já possuindo uma *E-BOM* que sirva ao propósito de armazenar os atributos de configuração. No configurador de produto atual, conforme discutido no item 4.2.6, os atributos de configuração estão em uma estrutura de nível único. Assim, considerando o conceito de módulo discutido na seção 2.1.3, objetiva-se, ao adotar uma estrutura multinível, atingir um grau de correspondência entre um módulo e os componentes físicos do motor elétrico.

A família de produtos escolhida para o estudo, a W22, possui um conceito de produto bem definido, permitindo estabelecer uma estrutura bastante clara e estável. Os módulos considerados nesse produto são apresentados na Figura 4.21:

Figura 4.21 – Estrutura geral de módulos da parte mecânica do motor elétrico



Fonte: Autoria própria

Um agrupamento por subsistemas também pode ser visualizado, utilizando a Figura 4.21 como base. Nesse caso, tem-se:

- i. Subsistema invólucro: composto por tampa dianteira e traseira e carcaça
- ii. Subsistema elétrico: composto por estator e pacote do rotor
- iii. Subsistema de refrigeração: composto por ventilador e defletora
- iv. Subsistema de conexão: composto por caixa de ligação
- v. Subsistema de potência: composto por eixo, chaveta e rolamentos.

A visão de subsistemas pode ser útil para uma futura expansão do sistema, considerando incorporar o subsistema elétrico ao novo sistema de configuração de produto.

No entanto, como já discutido, a parte elétrica não é parte do escopo do trabalho e, portanto, o subsistema elétrico não será utilizado. A divisão entre os demais subsistemas será aplicada para agrupar as restrições a serem escritas. Por exemplo, nas restrições do subsistema invólucro, pode ser armazenada a restrição de encaixe entre tampas e carcaças.

4.4.2 Definição dos valores disponíveis na configuração do produto

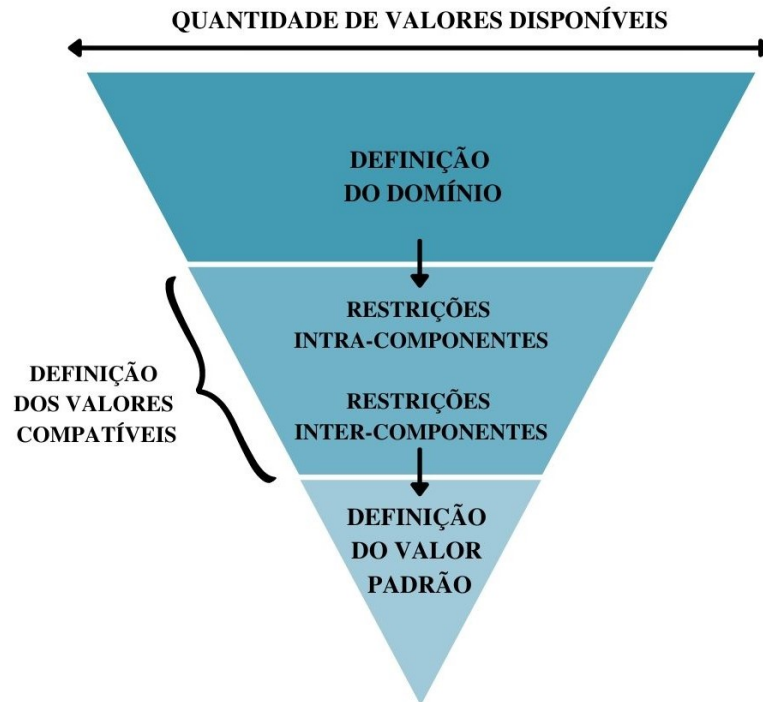
As etapas para definição dos valores disponíveis para um dado atributo no SCP são, conforme discutido na seção 4.3.2, definição do domínio das variáveis, seguida pela filtragem dos valores.

Para a filtragem dos valores será adotada uma nova abordagem, onde serão incorporadas as consistências no próprio processo de filtragem. Diferentemente das regras de dependência do sistema atual de configuração de produto da empresa-alvo descritas na seção 4.2.6, no novo sistema de configuração de produto, propõe-se que os valores disponíveis sejam filtrados com os seguintes passos:

- i. Tem-se a estrutura do produto conforme descrita anteriormente;
- ii. Os componentes possuem realizações físicas, as variantes de componentes. Estas variantes podem ser organizadas em um formato tabular, contendo os atributos representativos de cada componente.
- iii. Através do mecanismo de programação por restrições, essas tabelas são carregadas no sistema utilizando a funcionalidade de restrições tabulares. A essas restrições, denominar-se-á restrições intra-componentes, pois são consistências de existência dos componentes com eles mesmos. Nessas tabelas, sempre se encontra uma coluna dedicada ao código do componente, que posteriormente é utilizado na construção da *BOM* (conforme explorado na seção 2.1.5.9).
- iv. As outrora denominadas “regras de consistência” são convertidas no novo formato de restrições em formato de expressão ou em formato tabular, sendo executadas no momento de filtragem (não há mais uma etapa separada no final do processo de configuração). Essas serão denominadas restrições inter-componentes.

Pode-se expandir o processo demonstrado na Figura 4.19, esclarecendo a divisão das restrições intra e inter-componentes, conforme ilustrado na Figura 4.22:

Figura 4.22 – Processo expandido de filtro de valores de atributos



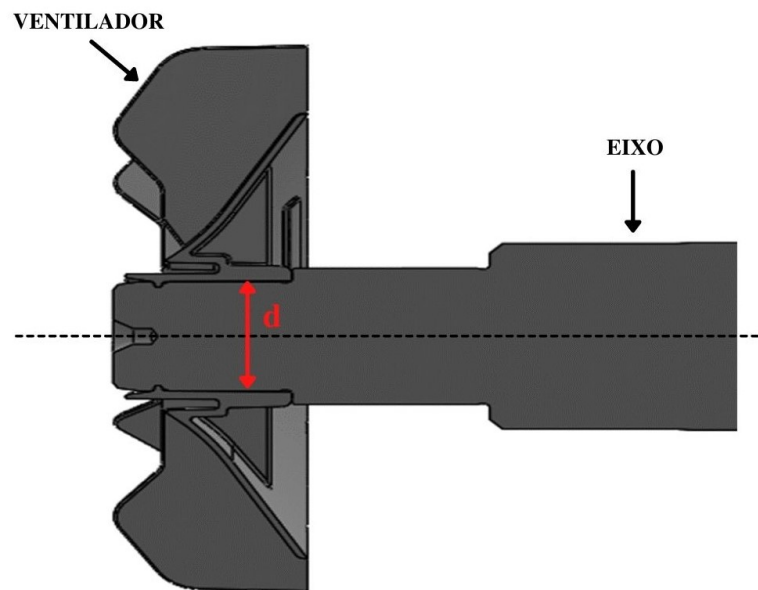
Fonte: Autoria própria

Dessa maneira, os valores filtrados em cada uma das caixas de seleção estão diretamente ligados às variantes de módulos existentes. A esse conceito, denomina-se “configurador baseado em componentes”, pois somente variantes de produto cujos componentes existem são oferecidas como opção. Além disso, as opções para cada componente são sempre mutuamente compatíveis.

Esse aspecto da modelagem é um avanço considerável em relação à situação atual da empresa-alvo. Atualmente, existem mais 15 milhões de linhas em tabelas de regras de dependência, que disponibilizam os valores para seleção no configurador de produto. A manutenção dessas regras, como a própria quantidade elevada de linhas sugere, traz elevado custo administrativo de manutenção e expansão. No sistema ERP, por outro lado, a base de dados de produtos é mantida separadamente, desconectada do configurador de produtos. Com esse conceito, unifica-se a base de componentes e a configuração de produto, evitando dissincronia, divergências e eliminando a dispendiosa manutenção de regras de dependência.

Do ponto de vista de restrições inter-componentes, também se atingem os objetivos desejados com essa abordagem. As consistências do sistema atual podem ser modeladas através de expressões e tabelas. Toma-se como exemplo a montagem do ventilador no eixo, onde o diâmetro interno do cubo do ventilador deve ser igual ao diâmetro do assento do ventilador no eixo, conforme mostra a Figura 4.23:

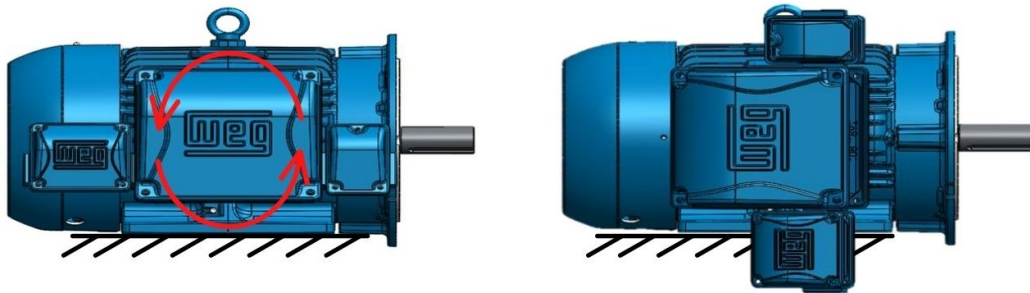
Figura 4.23 – Detalhe da montagem ventilador - eixo



Fonte: Autoria própria

Essa restrição pode ser facilmente modelada através de uma expressão de igualdade e será detalhada na seção de Implementação. Já as restrições tabulares, conforme mencionado, podem ser inter ou intra-componentes. Um exemplo de restrição tabular inter-componentes necessária para aspectos de projeto mecânico são as posições permitidas de montagem da caixa de ligação. Esse componente pode ser rotacionado de 90 em 90 graus para a família-alvo de produtos. No entanto, nem todas as configurações são permitidas, pois pode haver contato com o solo ou outros componentes do motor, como exemplificado na Figura 4.24:

Figura 4.24 – Motor com saída dos cabos para baixo (à esquerda) e saída dos cabos para ponta do eixo (à direita)



Fonte: Autoria própria

Nesta figura, fica evidente que, ao rotacionar a caixa de ligação, a caixa auxiliar (que está aparafusada à caixa de ligação) acaba por ultrapassar o solo, não sendo possível montar o motor diretamente ao chão e inviabilizando o fornecimento do produto. Diferentemente de restrições unidimensionais como da montagem do ventilador no eixo, que podem ser modeladas através de uma expressão de igualdade, seria muito mais complexo construir a consistência de contato da caixa no solo dessa maneira. Por este motivo, é mais adequado construir uma tabela de configurações permitidas e não-permitidas, utilizando uma restrição tabular que combine atributos de tamanho de carcaça, posição da caixa e existência ou não de caixas auxiliares (atributos de diferentes componentes) e assinalar se essa combinação é montável ou não.

Discutindo a questão das camadas de restrição, as restrições mencionadas acima são enquadradas na camada de engenharia. No entanto, como mencionado na seção 3.4.2, existem outras camadas, como de Manufatura, *Compliance* e Marketing. A área de Manufatura da empresa-alvo terá interesse em restringir algumas questões de fabricação, como o comprimento máximo que um eixo pode ter. Já a área de *Compliance* tem diversas restrições, conforme discutido por Haselein (2018). A camada de Marketing será discutida a seguir, na próxima seção.

4.4.3 Definição dos valores padrões na configuração do produto

No protótipo, os valores padrões foram classificados como recomendações da camada de Marketing, conforme discutido anteriormente, sendo convenientes a sua existência

para o usuário final. Os valores padrões podem ser definidos utilizando-se um recurso equivalente à restrição *soft* discutida na seção 3.3.3.

Tais restrições *soft* podem ser expressões ou tabelas, assim como as demais restrições. Nesse ponto também há um ganho na modelagem baseada em restrições, pois existe, desde a primeira seleção, a exclusão de valores não recomendados, além de haver a liberdade na ordem de seleção de valores – diferentemente da obrigatoriedade a ser seguida no sistema atual discutido em 4.2.6.

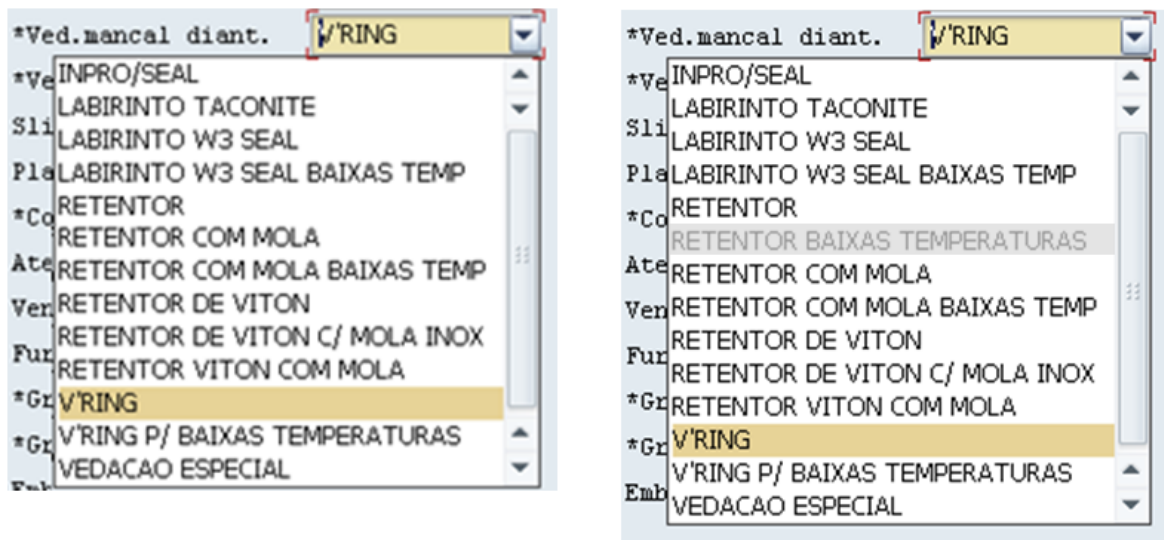
4.4.4 Verificação de inconsistências na configuração e mecanismo de explicações

Segundo a formulação discutida até aqui, a verificação de inconsistências como uma etapa posterior à configuração foi eliminada, sendo incorporadas às restrições gerais do sistema. Essas restrições estão sendo separadas em diferentes camadas – Marketing, Engenharia, Manufatura e *Compliance* – e podem ser classificadas entre inter e intra-componentes.

Trentin, Perin e Forza (2013) postularam a necessidade de ter dois configuradores, um para venda, com atributos mais simples, e outro para engenharia, com maior nível de detalhamento. No entanto, vislumbrou-se a possibilidade de unificar a modelagem, dispondo de um configurador único que se comporta de maneira diferente para diferentes usuários, caso seja possível mostrar uma quantidade maior ou menor de atributos para diferentes perfis de usuário.

Na questão de filtragem de valores de um atributo, pode-se separar dois comportamentos para usuários de nível mais básico e avançados. A título de exemplo, tem-se o atributo “vedação dianteira” mostrada na Figura 4.13, e suposição de que o valor “Retentor para baixas temperaturas” esteja incompatível devido a alguma restrição. Para usuários de nível básico, valores incompatibilizados por uma restrição são removidos das opções; para usuários de nível mais avançado, valores incompatíveis são esmaecidos, conforme exemplificados pela Figura 4.25:

Figura 4.25 – Visão de usuário nível básico (à esquerda) e visão de usuário nível avançado (à direita)



Fonte: Autoria própria

Na visão básica, não é interessante que o valor seja mostrado, pois de qualquer maneira ele não é compatível considerando a configuração atual, o que evita questionamentos e dúvidas. Já na visão avançada, é relevante que seja mostrado, já que desta forma o usuário pode optar por selecionar este valor (caso a *shell* utilizada permita) e receber algum tipo de explicação ou alternativa para que esta seleção aconteça.

4.5 IMPLEMENTAÇÃO

4.5.1 *Shell* CBCConf

A empresa-alvo possui um longo histórico de customização não só de produtos, mas também dos sistemas que suportam as operações. O trabalho de Massirer (2007) mostra uma proposta de sistema especialista desenvolvido para uma aplicação específica, que posteriormente à escrita do trabalho foi desenvolvido dentro do sistema ERP comercial SAP. O SAP é o sistema ERP mais utilizado do mundo, com 50 anos de utilização e milhares de clientes ao redor do mundo (SAP, 2022), e permite customizações em larga escala nos seus programas através de uma linguagem própria, o ABAP (TAKAHASHI, 2020).

Os trabalhos de Haselein (2018) e Vegini (2018), por outro lado, foram construídos dentro de uma *shell* própria da empresa-alvo, o Maestro. Essa *shell* é um motor de regras

sequenciais, com possibilidade de escrever regras de forma *low-code* ou utilizando a linguagem de programação Java, cujo objetivo era aumentar a autonomia de áreas fora da TI no desenvolvimento de aplicações (HASELEIN, 2018), tendo sido desenvolvido em 2010.

Devido à *shell* Maestro adotar somente um paradigma de programação sequencial e baseado em eventos, foi desenvolvido nos últimos anos na empresa-alvo uma nova *shell*, o CBCConf. Essa *shell* é uma camada construída sobre a biblioteca de programação por restrições Choco-Solver, tornando possível programar de forma não-sequencial.

O CBCConf tem o propósito de facilitar a construção de configuradores de produto. O Choco-Solver, por sua vez, é uma biblioteca *open-source*, cuja primeira versão surgiu em 2003 por uma comunidade sem relação com a empresa-alvo, tendo sido construída de forma ampla para o problema de satisfação de restrições. Um exemplo clássico de problema que pode ser resolvido pelo Choco-Solver é o Problema das Oito Damas (CHOCO-SOLVER, 2022).

O CBCConf é uma *shell* bastante recente na empresa-alvo, motivo pelo qual somente algumas ferramentas *low-code* para fazer cargas de dados foram construídas:

- i. *Table-constraints*: aplicação web que permite configurar restrições em formato tabular;
- ii. *Attributes*: aplicação web que permite configurar domínios de variáveis;

Demais dados e parâmetros para construção de um configurador de produto usando o CBCConf ainda exigem a estruturação de um programa Java em uma IDE adequada. A estrutura de componentes, os componentes e seus atributos e algumas restrições de natureza mais complexa são escritas em código-fonte, utilizando uma sintaxe similar à mostrada na seção 3.3.2.

A *shell* possibilita a estruturação de um configurador consistido de componentes – que podem ser interpretados, para o presente trabalho, como módulos – e uma base de conhecimento formada por uma série de restrições. Ao disparar a execução do *solver* da *shell*, indiretamente será chamado a execução do Choco-Solver, que irá encontrar (ou não) a solução para o problema. Apesar de existirem abordagens não-determinísticas para solução de satisfação de restrições (BEEK, 2006), na *shell* utilizada, o processo é inteiramente determinístico, isto é, sempre encontra o mesmo resultado dadas as mesmas entradas. No caso de não haver solução, pode ser disparado um algoritmo de solução de conflitos, que será discutido a seguir.

4.5.1.1 Algoritmo de resolução de conflitos

De acordo com Meseguer et al. (2006), é comum que problemas reais não tenham uma solução (problema sobrerrestrito). Para tanto, existem soluções otimizadas na literatura para encontrar uma solução para o conflito. Os dois algoritmos mais comuns são QuickXplain e FastDiag (FELFERNIG et al., 2014).

Ambos os algoritmos consistem em encontrar um subconjunto mínimo de restrições em uma base de conhecimento que possuam certa propriedade, por exemplo, uma propriedade de gerar um determinado conflito com as seleções realizadas pelo usuário. Para tanto, utilizam uma estratégia de dividir para conquistar, isto é, realiza uma busca dividindo a base de restrições pela metade, aplicando-a e avaliando se o conflito permanece; o sistema faz isso iterativamente, até que o subconjunto mínimo seja encontrado (RODLER, 2022). O algoritmo FastDiag otimiza essa estratégia ao focar em subconjunto mais prováveis (FELFERNIG et al., 2014).

Na *shell* CBConf, ambos os algoritmos estão implementados e são escolhidos automaticamente pela *shell*. O funcionamento, descrito de um ponto de vista de um engenheiro do conhecimento que está modelando um configurador de produto, é o seguinte:

- i. Uma estrutura de produto foi definida;
- ii. Uma série de restrições foi definida;
- iii. Uma tela de configuração para testes foi acessada;
- iv. Alguns campos foram selecionados pelo usuário;
- v. O usuário se depara com um valor que ele deseja destacado como não compatível, mas mesmo assim ele o seleciona;

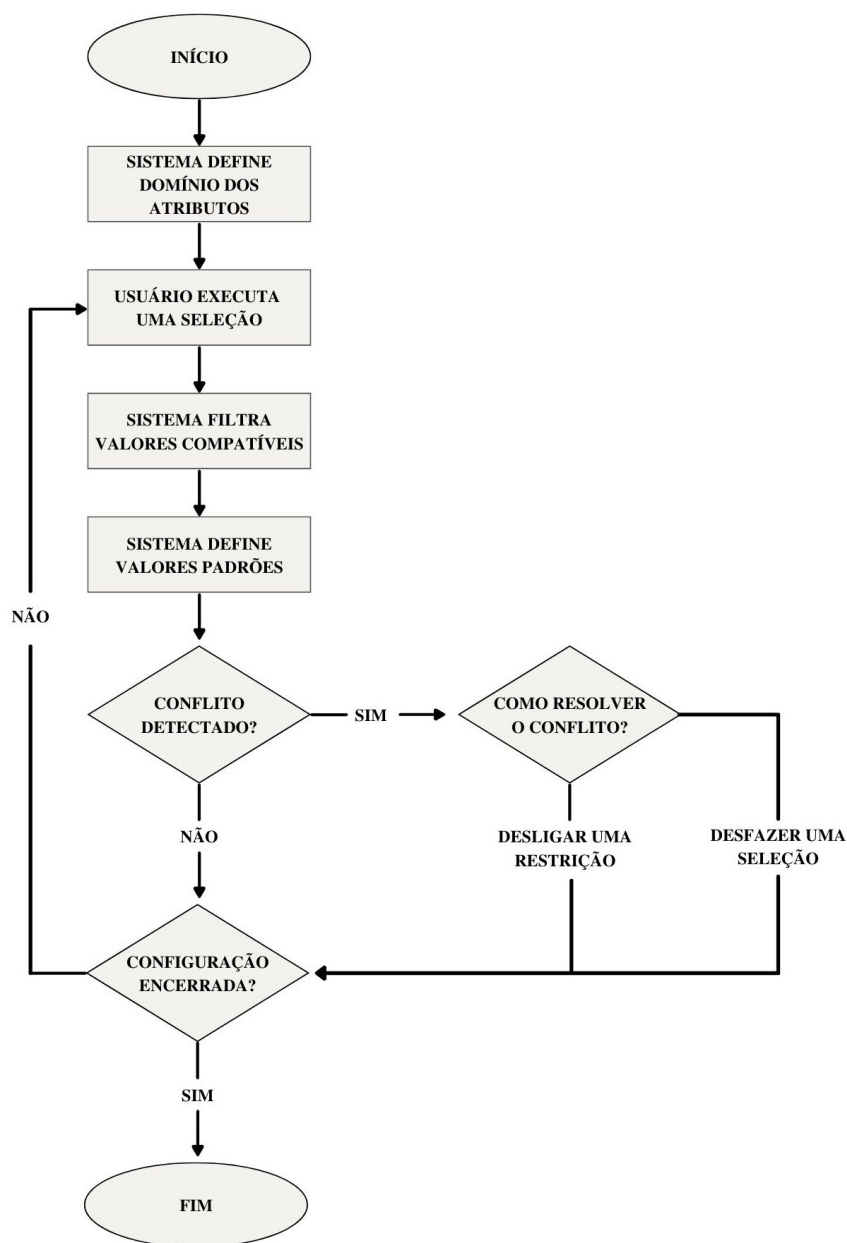
Nesse momento, a *shell* disparará o algoritmo de resolução de conflitos, que resultará em uma lista de diagnósticos possíveis. Cada elemento dessa lista é um diagnóstico, que consiste em uma lista de ações necessárias para permitir o valor incompatível selecionado. Essas ações podem ser de dois tipos:

- i. Desfazer uma seleção prévia feita pelo usuário;
- ii. Desligar uma restrição da base de conhecimento.

Por “desligar uma restrição”, entende-se que a restrição será completamente desconsiderada, funcionando o sistema como se ela não existisse. Uma aplicação prática desse mecanismo será explorada no próximo capítulo.

Pode-se visualizar essa parte do sistema como uma expansão do fluxo anteriormente mostrado na Figura 4.20, conforme apresentado na Figura 4.26:

Figura 4.26 - Fluxo proposto de funcionamento do novo sistema com resolução de conflitos



Fonte: Autoria própria

Nessa figura, ilustra-se o funcionamento de que, a cada nova seleção realizada pelo usuário, o sistema executa toda a propagação de restrições – filtrando valores compatíveis e valores padrões (se houver), e ao longo deste processo, podem ocorrer eventuais conflitos.

Esses conflitos ocorrem quando a aplicação de alguma restrição deixa algum atributo sem valores disponíveis (rede inconsistente em arco como explorado no exemplo da seção 3.3.1). Nesse caso, serão calculadas possíveis soluções para o conflito e apresentados ao usuário; após o usuário definir o que deve ser feito para eliminar a contradição, o processo segue o fluxo normal.

4.5.2 Implementação da estrutura do produto

A implementação da estrutura do produto foi realizada utilizando os recursos da *shell* CBConf, convertendo-se os módulos da estrutura elaborada na seção 4.4.1. Para tanto, os módulos foram implementados como *Components* na *shell*.

Nesta *shell*, não há distinção entre componentes e conjuntos nos termos discutidos na seção 2.1.5, mas adotar-se-á a formalidade de que módulos são *Components* do tipo folha, isto é, não possuem portas para outros módulos. Os conjuntos, por sua vez, são *Components* que possuem portas para outros conjuntos ou módulos. Por exemplo, a Carcaça é um módulo, enquanto o Motor é classificado como conjunto.

Outro aspecto importante é que alguns módulos podem se repetir na estrutura de produto. Por exemplo, no motor elétrico, tem-se um rolamento dianteiro e um rolamento traseiro. Nesse caso, o módulo é o “rolamento”, enquanto o rotor possui as portas “rolamento dianteiro” e “rolamento traseiro”, sendo que cada porta está associada a uma instância de um módulo “rolamento”.

Tendo os módulos e portas definidos – compondo uma estrutura básica – é necessário acrescentar em cada módulo seus atributos. Por exemplo, o módulo “rolamento” possui os atributos de tamanho, blindagem, diâmetro externo, diâmetro interno, largura e folga. Neste exemplo, alguns atributos são numéricos (como o diâmetro externo, interno, largura) e outros atributos são do tipo textual, tendo valores pré-definidos. O tamanho pode assumir diversos valores diferentes conforme catálogos de fabricantes (6201, 6301, NU-301, etc... são alguns exemplos), assim como a blindagem (ZZ, 2RS, etc...) e folga (C3, C4, C5, etc..). A definição dos domínios (os valores disponíveis) será detalhada na próxima seção. Alguns exemplos da implementação da estrutura de módulos podem ser encontrados no Apêndice A1.

4.5.3 Implementação dos domínios

A partir dos módulos e seus atributos definidos anteriormente, foi necessário levantar todos os valores possíveis de cada atributo. Por exemplo, para o atributo “tamanho de rolamento”, podem-se extrair os valores de um catálogo de um fabricante de rolamentos como NSK ou SKF. Há que se ter em mente que esses valores serão apresentados ao usuário final do sistema, portanto deve-se utilizar o nível de detalhe adequado para o público-alvo. No caso de detalhes específicos de nomenclatura de um fabricante, convém suprimi-los em favor de uma descrição mais genérica do tamanho dos rolamentos, por exemplo, tamanho de rolamento “6205” ou “6206” ao invés da descrição completa dos catálogos.

Na *shell* utilizada, esses valores são cadastrados em uma tabela com suas traduções possíveis, e os domínios são associados aos atributos através de instruções específicas diretamente no código-fonte Java. Maiores detalhes de implementação podem ser encontrados no Apêndice A2.

4.5.4 Implementação dos valores disponíveis

Como mencionado anteriormente, os valores disponíveis de cada atributo são filtrados através da propagação de restrições pela *shell*, e este processo é baseado em restrições do tipo intra e inter-componentes.

As restrições tabulares para representação das variantes de módulos (restrições intra-componentes) são definidas na *Shell* CBCConf através da aplicação *TableConstraints*. As informações necessárias para construir essas tabelas são extraídas do próprio ERP da empresa-alvo, o SAP. Foi necessário extrair as informações de todos os módulos considerados na estrutura desenhada previamente. Por exemplo, extraíram-se todos os ventiladores disponíveis no sistema; selecionam-se os atributos relevantes para o sistema de configuração e monta-se uma tabela de variantes disponíveis. Um exemplo de preenchimento de tabela é mostrado na Figura 4.27:

Figura 4.27 – Exemplo de preenchimento de uma restrição tabular com variantes de um componente (ventilador)

Material	Balaceamento	Diâmetro externo	Diâmetro interno cubo	Estado peça	Largura chaveta	Largura cubo venti	Material ventilador
10019575	false - False	184	35	00001 - INJETADO	0	28	00001 - POLIPROPILENO HOMOPOLIMERO
10019576	false - False	270.5	44	00001 - INJETADO	0	30	00001 - POLIPROPILENO HOMOPOLIMERO
10019577	false - False	343.5	55	00001 - INJETADO	0	40	00001 - POLIPROPILENO HOMOPOLIMERO
10021194	false - False	110	10	00001 - INJETADO	0	14.5	00001 - POLIPROPILENO HOMOPOLIMERO
10021195	false - False	120	12	00001 - INJETADO	0	15.5	00001 - POLIPROPILENO HOMOPOLIMERO
10021196	false - False	120	12.7	00001 - INJETADO	0	15.5	00001 - POLIPROPILENO HOMOPOLIMERO
10021197	false - False	132	15	00001 - INJETADO	0	17.5	00001 - POLIPROPILENO HOMOPOLIMERO
10021198	false - False	157	17	00001 - INJETADO	0	17.5	00001 - POLIPROPILENO HOMOPOLIMERO
10021199	false - False	180	23	00001 - INJETADO	0	22	00001 - POLIPROPILENO HOMOPOLIMERO
10021201	false - False	150	28	00001 - INJETADO	8	25	00001 - POLIPROPILENO HOMOPOLIMERO
10021202	false - False	190	28	00001 - INJETADO	8	25	00001 - POLIPROPILENO HOMOPOLIMERO
10021203	false - False	170	29.95	00001 - INJETADO	8	25	00001 - POLIPROPILENO HOMOPOLIMERO
10021204	false - False	230	29.95	00001 - INJETADO	8	25	00001 - POLIPROPILENO HOMOPOLIMERO

Fonte: Autoria própria

As colunas da tabela mostrada na Figura 4.27, que receberam um nome na aplicação *Table Constraints*, são conectadas às propriedades dos modelos através de restrições específicas.

As restrições do tipo expressão, utilizadas para consistências inter-componentes são construídas a partir de relações lógicas entre os componentes. De maneira geral, são restrições de montagem. Todos os pontos de montagem do produto devem ser verificados, para que seja feita a associação correta entre os componentes. Por exemplo, o diâmetro interno do ventilador deve ser igual ao diâmetro do assento do ventilador no eixo; o diâmetro interno dos rolamentos também deve ser compatível com o eixo; os diâmetros externos dos rolamentos devem ser compatíveis com a tampa; os encaixes das tampas devem ser compatíveis com a carcaça.

Em alguns casos existem interferências na montagem entre os componentes, não sendo possível simplesmente igualar uma propriedade de cada componente. Isso ocorre, por exemplo, quando um dos componentes é composto de elastômero e sofrerá uma deformação considerável ao se encaixar a um componente, por exemplo, manufaturado em ferro fundido. Nesse caso, é necessário extrair do sistema atual qual a relação de interferências aceitáveis para então construir uma restrição adequada, que leva em consideração o valor dessa interferência.

Maiores detalhes da implementação das restrições inter e intra-componentes são mostradas no Apêndice A3.

4.5.5 Implementação dos valores padrões

Para a implementação dos valores padrões, buscou-se um comportamento o mais semelhante possível ao do atual sistema de configuração, sendo necessário construir algumas restrições do tipo recomendação (restrição *soft*).

As restrições de valores padrão implementadas no protótipo utilizam os atributos iniciais do configurador atual, definindo uma série de atributos do produto, visto que esse é o comportamento que é observado no sistema atual de configuração.

Nesse ponto também existe um ganho na modelagem baseada em restrições, pois existe, desde a primeira seleção, exclusão de valores não recomendados, além de não haver obrigatoriedade de uma ordem a ser seguida. Ambos os comportamentos não são observados no sistema atual de configuração, pois, como mencionado anteriormente, é necessário realizar a definição de todos os atributos iniciais para ter um produto completamente configurado.

Um exemplo da vantagem de não precisar seguir uma ordem no processo de configuração é o seguinte: é possível iniciar a configuração pela potência do motor sem saber previamente quais linhas de produto atendem determinado valor. No sistema atual, é necessário saber de antemão as potências atendidas por cada linha, pois a linha de produto é solicitada antes da potência. Detalhes da implementação dos valores padrões na *shell* CBCConf podem ser visualizadas no Apêndice A4.

4.5.6 Implementação das consistências e mecanismo de explicações

As ideias sobre consistências e explicações exploradas na 4.4.4 encontraram sustentação parcial na *shell*. Por um lado, a eliminação das consistências apenas ao final do processo de configuração foi atingida, pois todas as consistências acontecem no próprio processo de propagação das restrições intra e inter-componentes; no entanto, o mecanismo de esconder valores para determinados usuários não foi alcançado no momento.

Com a implementação, é possível selecionar os valores incompatíveis que são visíveis em um atributo; nessa situação, a *shell* dispara o mecanismo de resolução de conflito da *shell* CBCConf. Dessa maneira, o motor de resolução de conflito dispara dois processamentos:

- i. Encontrar uma ou mais alterações em valores já selecionados para o usuário que, se desfeitos, permitirão a seleção do valor incompatível;

- ii. Encontrar uma ou mais restrições que, se desligada(s), permitirá a seleção incompatível;

No item i), há uma explicação sugerindo uma correção de caminho para o usuário, mas pode não ficar claro exatamente qual a origem da limitação. Na segunda opção, a *shell* apresenta o nome da restrição ou do conjunto de restrições que será desligado em uma tela. Considerando esse comportamento da *shell*, utilizou-se descrições as mais detalhadas possíveis no campo nome da restrição, de modo que esse recurso sirva como uma explicação para o usuário.

O mecanismo de esconder valores para alguns usuários não foi implementado na *shell*, permanecendo como uma melhoria futura que depende da evolução da própria ferramenta.

Conforme mostrado ao longo do capítulo, apresentou-se o desenvolvimento do protótipo do sistema de configuração de produto para um produto técnico complexo escolhido na empresa-alvo, e utilizando também uma *shell* para construção de sistemas de configuração de produto da empresa-alvo.

Seguindo a divisão de construção de um sistema especialista proposta por Waterman (1986), detalhou-se os processos de Identificação, Conceitualização, Formulação e Implementação do sistema, através da interação com especialistas e os próprios sistemas da empresa-alvo (o sistema prévio de configuração de produto e o sistema ERP), buscando codificar o conhecimento em um sistema especialista.

No capítulo seguinte, focar-se-á a etapa de validação do sistema especialista. Nessa etapa, além de testar o sistema básico desenvolvido, discutir-se-á alguns estudos de caso que exigirão um novo ciclo de identificação-conceitualização-formulação-implementação-validação, mas de menor escala e esforços envolvidos. Em ambos os capítulos, pontuar-se-ão questões que podem divergir caso adotado o mesmo *framework* para uma empresa em um diferente nível de adoção de um sistema de configuração ou um produto de natureza significativamente diferente.

5 VERIFICAÇÃO E AVALIAÇÃO DO SISTEMA

O protótipo discutido no capítulo anterior alcançou um nível de maturidade que permitiu aplicá-lo em situações cotidianas da empresa-alvo. Essas avaliações são importantes não só para atestar a viabilidade do sistema, bem como para comprovar suas vantagens sobre o processo atual. Os procedimentos aqui discutidos também são bastante genéricos, de modo a poderem ser aplicadas em outros tipos de produtos. Nesse capítulo, quando for referido ao sistema proposto, será utilizada a expressão “sistema-protótipo”, enquanto que para o sistema especialista em uso na empresa-alvo, será utilizada a expressão “sistema atual”.

O processo está dividido em duas etapas, uma de verificação (sem o envolvimento de um especialista externo) e uma de validação (com um especialista externo). Durante o processo de validação, foi convidado um especialista em configuração de produto da empresa-alvo com 15 anos de experiência para efetuar uma avaliação qualitativa do sistema-protótipo, comparando-o com o sistema atual. Esse tipo de avaliação por um especialista também foi explorado por Massirer (2007).

5.1 INTERFACE E UTILIZAÇÃO DO SISTEMA-PROTÓTIPO

Apresenta-se a seguir o resultado obtido em termos de interface e utilização do sistema-protótipo. O objetivo é ressaltar a representação visual da modelagem de produto discutida no capítulo anterior, com efetiva aplicação dos conceitos abordados nas referências.

A interface padrão gerada pela *shell* é mostrada na Figura 5.1. Ela é formada por duas áreas principais: à esquerda, há uma árvore de nós mostrando a estrutura codificada, e à direita, as propriedades de um nó selecionado.

Figura 5.1 – Visão geral da tela de configuração

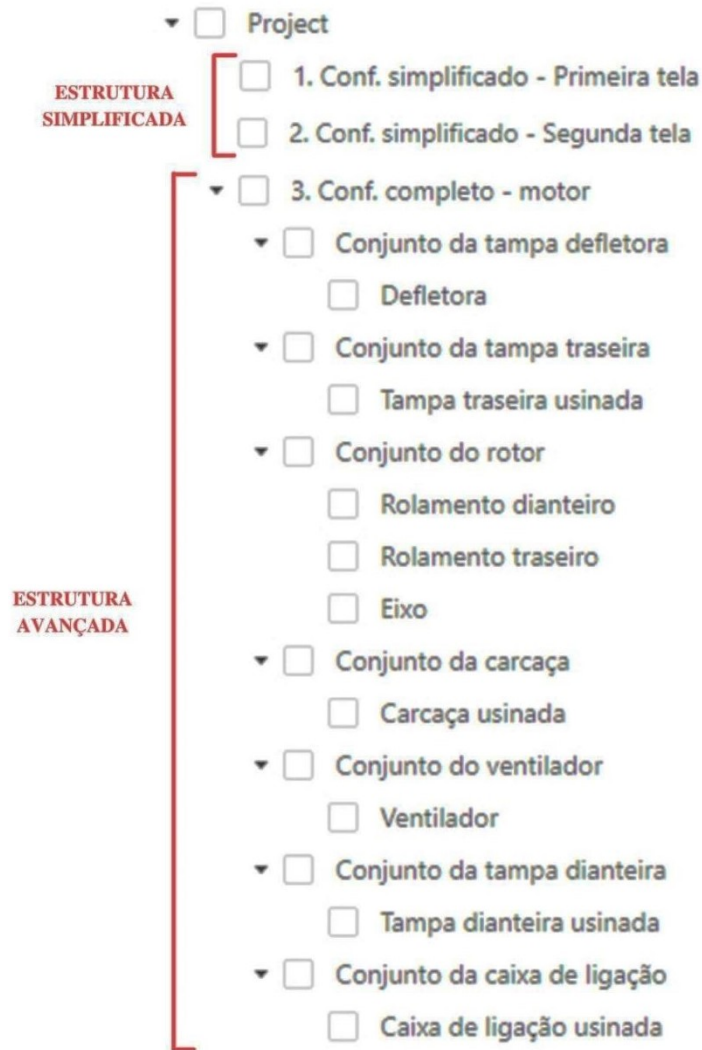


Fonte: Autoria própria

Conforme mencionado anteriormente, optou-se por construir uma estrutura com duas partes. Uma parte é para a versão simplificada do configurador atual, sendo representado pelo nó de nome “Configurador simplificado”. Esse seria o único nó visível para um usuário iniciante no sistema-protótipo. De início, também é possível notar na Figura 5.1 que, com as recomendações codificadas, alguns campos aparecem preenchidos – por exemplo, a forma construtiva está com o valor “B3D”. Estes são os valores recomendados pelo sistema inicialmente pela camada de restrições de marketing, mas é possível alterá-los.

A segunda parte é a estrutura do produto propriamente dita, formada por um nó principal denominado “configurador completo”, com os módulos mostrados na Figura 4.21. Essa seria a estrutura acessada por usuários mais avançados, que podem fazer seleções e investigações mais complexas no produto. Essas duas estruturas são apresentadas de forma ampliada na Figura 5.2:

Figura 5.2 – Árvore completa apresentada no sistema-protótipo



Fonte: Autoria própria

No protótipo, uma estrutura em dois níveis foi elaborada (simplificada e avançada), mas é possível desenvolver mais níveis caso necessário. Por exemplo, poderia haver um nível ainda mais avançado, incluindo todos os componentes do produto. No entanto, para o protótipo, dois níveis foram suficientes para comprovar o conceito.

5.2 VERIFICAÇÃO DAS RESTRIÇÕES

Para confirmar o funcionamento do sistema-protótipo de um ponto de vista da exatidão das restrições construídas, foram selecionados 10 produtos da base da empresa-alvo, um de cada tamanho de carcaça, a fim de verificar se o sistema-protótipo seria capaz de

selecionar, na estrutura avançada do produto, os componentes reais. Todos os produtos e componentes são representados por um código interno de 8 dígitos, sendo que é possível fazer a validação por uma comparação simples.

O processo ocorre da seguinte forma: na tela inicial, faz-se a seleção correspondente a um desses produtos. No nó “configurador simplificado”, isso implica em fazer a seleção do atributo “Carcaça com complemento”, conforme mostrado na Figura 5.3. Os demais atributos já estão preenchidos com recomendações.

Figura 5.3 – Tela inicial após seleção do tamanho da carcaça

Configurador simplificado ⊗ close

Properties

• Alimentação	TRIFASICO	• Blindagem rolament...	ZZ	• Blindagem rolament...	ZZ
• Carcaça com compl...	63	• Com sensor Motor S...	false	• Com sensor de vibra...	false
• Direção de saída do...	PARA BAIXO	• Eixo dupla ponta	false	• Entidade norma	IEC
• Fixação do rolamento	SEM	• Folga interna rolam...	NORMAL	• roiga interna rolam...	NORMAL
• Forma construtiva	B3D	• Frequência	60Hz	• Furo A1 da caixa de ...	ISO 7 - Rp ...
• Furo A2 da caixa de ...	SEM	• Furo A3 da caixa de ...	SEM	• Furo A4 da caixa de ...	SEM
• Grau de proteção	IP55	• Graxa do rolamento	POLYREX EM	• Invólucro	FECHADO
• Linha de produtos	W22	• Material da defletora	ACO CARB...	• Material do eixo	SAE 1040/45
• Material do invólucro	FERRO	• Material do ventilador	POLIPROPI...	• Mercado	BRASIL
• Refrigeração do ma...	false	• Tamanho do rolame...	6201	• Tamanho rolamento...	6201
• Tipo canal de chaveta	CHAVETA B	• Tipo de graxeira	SEM GRAX...	• Tipo e tamanho flan...	SEM FLAN...
• Tipo vedação diante...	ANEL V	• Tipo vedação traseira	ANEL V	• terminalBoxModel	W22

Fonte: Autoria própria

Na sequência, verifica-se na estrutura avançada (nó motor) se os componentes selecionados principais são os itens esperados conforme a *BOM* dos produtos existentes na empresa-alvo. Como componentes relevantes para a verificação, adotou-se nesse teste os componentes tampa dianteira, traseira e defletora, eixo, carcaça e caixa de ligação. A seguir, na Figura 5.4, é mostrada a lista técnica de um produto real da empresa-alvo:

Figura 5.4 – Parte da lista técnica extraída do ERP de um produto

11417328 1100 1 01	MOTOR 0.25cv 4P 63 WFF2
▸ 0001 L 11241460	DESENHO MONTAGEM GENERICO
▾ 1000 L 10922665	CONJ CARACA
▾ 10922665 1100 1 01	CONJ CARACA
▸ 0010 L 10922607	CARACA USIN W22 63 C/PES CX LAT
▸ 1100 L 15043694	ESTATOR BOB F 0.25CV 4P 63
▾ 1500 L 12338964	ROTOR COMPLETO 55mm 4P 63
▾ 12338964 1100 1 01	ROTOR COMPLETO 55mm 4P 63
▸ 0010 L 11400499	ROTOR INJ 55mm 4P 63 DESC
▸ 0020 L 11369708	EIXO TRIF 63 W21/W22
▸ 1521 L 11311049	CONJ CHAVETA
▸ 2000 L 10330190	CONJ ROLAMENTO 6201 ZZ
▸ 2100 L 10330190	CONJ ROLAMENTO 6201 ZZ
▾ 2500 L 10947180	CONJ TAMPA DIANTEIRA ELEM FIX ACO
▾ 10947180 1100 1 01	CONJ TAMPA DIANTEIRA ELEM FIX ACO
▸ 0010 L 10947048	TAMPA DIANT USIN 63 W22 V'RING
• 0020 L 10018555	PARAF SEX M5X0,8X25 RT ZTAM 5.6
• 0280 L 10018929	ANEL V NBR A 10,5X16,5X5,5mm
▸ 2511 L 15845129	CONJ DRENO
▾ 2700 L 10947916	CONJ TAMPA TRASEIRA ELEM FIX ACO
▾ 10947916 1100 1 01	CONJ TAMPA TRASEIRA ELEM FIX ACO
▸ 0010 L 10947909	TAMPA TRAS 63 USI W22 ANEL V
• 0020 L 10018555	PARAF SEX M5X0,8X25 RT ZTAM 5.6
• 0330 L 10018929	ANEL V NBR A 10,5X16,5X5,5mm
• 0550 L 10038186	ARRUELA ONDULADA 26X31
▸ 2711 L 15845129	CONJ DRENO
▸ 5000 L 10446465	CONJ VENTILADOR EXTERNO
▾ 6000 L 13592838	CONJ TAMPA DEFLETORA ELEM FIX ACO
▾ 13592838 1100 1 01	CONJ TAMPA DEFLETORA ELEM FIX ACO
▸ 0010 L 10161599	TAMPA DEFL ESTAMPADO 63
• 0020 L 11604669	PARAF SEX M5X0,8X4 RT ZTAM 5.6
▾ 6500 L 10935016	CONJ CX LIGACAO PRINC ELEM FIX ACO
▾ 10935016 1100 1 01	CONJ CX LIGACAO PRINC ELEM FIX ACO
▸ 0010 L 10897454	CAIXA LIGACAO USIN 63-100 W22

Fonte: Autoria própria

Essa imagem foi extraída do ERP da empresa-alvo, sendo destacados os itens que foram selecionados para fazer a comparação. Os códigos de 8 dígitos também estão em destaque, podendo ser conferidos no Apêndice A5.

Prosseguindo no sistema-protótipo, na Figura 5.5, é apresentado o nó eixo após a seleção da carcaça com complemento. Todas as propriedades ficam definidas, como consequência das demais recomendações:

Figura 5.5 – Tela do nó eixo após seleção do tamanho da carcaça

Eixo

⊗ close

Properties

• Aplicação anel de re...	SEM	▼	• Carcaça com compl...	63	▼	• Com encoder	false	▼
• Comprimento da ch...	18	● ✓	• Comprimento da ch...	0	● ✓	• Comprimento da po...	23	● ✓
• Comprimento da po...	0	● ✓	• Comprimento do as...	10	● ✓	• Comprimento do as...	10	● ✓
• Comprimento do as...	79.5	● ✓	• Comprimento do as...	19.5	● ✓	• Comprimento do en...	20	● ✓
• Comprimento do en...	20	● ✓	• Comprimento rosca...	0	● ✓	• Comprimento rosca...	0	● ✓
• Comprimento total	203	● ✓	• Distância da face até...	0	● ✓	• Distância da guia do...	19.5	● ✓
• Distância da ponta d...	42.5	● ✓	• Distância da ponta t...	41	● ✓	• Distância do encost...	20	● ✓
• Distância do encost...	20	● ✓	• Distância do fundo ...	8.5	● ✓	• Distância do fundo ...	0	● ✓
• Distância entre rola...	119.5	● ✓	• Diâmetro da ponta ...	11	● ✓	• Diâmetro da ponta t...	0	● ✓
• Diâmetro de guia de...	15.6	● ✓	• Diâmetro do assent...	0	● ✓	• Diâmetro do assent...	12	● ✓
• Diâmetro do assent...	12	● ✓	• Diâmetro do assent...	16.1	● ✓	• Diâmetro do assent...	10	● ✓
• Diâmetro do assent...	0	● ✓	• Diâmetro do encost...	15.6	● ✓	• Diâmetro do encost...	15.6	● ✓
• Eixo dupla ponta	false	▼	• Furo de centro diant...	A 3.15 WE...	▼	• Furo de centro trase...	A 3.15 WE...	▼
• Largura da chaveta ...	4	● ✓	• Largura da chaveta t...	0	● ✓	• Linha de produtos	W21/W22	▼
• Material do eixo	SAE 1040/45	▼	• Polaridade do eixo	TODAS	▼	• Sistema de fixação d...	PINO ELAS...	▼
• Tipo canal de chavet...	CHAVETA B	▼	• Tipo canal de chavet...	SEM	▼	• Tipo de fixação do r...	ESTRIADO	▼
• itemNumber	11369708	▼						

Fonte: Autoria própria

Nesta figura, é possível observar que o atributo *itemNumber*, que identifica o produto com o código de 8 dígitos, está selecionada em decorrência da seleção do tamanho de carcaça que foi feita no nó “configurador simplificado”. Esse componente é compatível, portanto, com a seleção de carcaça com complemento com o valor 63 selecionado no nó “configurador simplificado”.

As restrições tabulares que representam as variantes de componentes carregadas para o sistema-protótipo revelam uma grande quantidade de itens possíveis de serem selecionados, conforme quantidades abaixo:

- 2045 tampas dianteiras;
- 4266 tampas traseiras;
- 340 tampas defletoras;
- 766 carcaças;
- 1582 eixos;
- 3752 caixas de ligação

Portanto, para que o processo de configuração de um produto qualquer resulte na seleção correta de cada um dos componentes, é necessário que existam restrições corretamente codificadas. Na Tabela 5.1, reproduz-se parcialmente os resultados obtidos.

Tabela 5.1 - Amostra de itens reais dos produtos versus itens selecionados pelo protótipo

Teste de seleção		Carcaça		Tampa Dianteira	
Produto	Tamanho do produto	Real	Protótipo	Real	Protótipo
1	63	10922607	10922607	10947048	10947048
2	71	10922320	10922320	10942872	10942872

Fonte: Autoria própria

A tabela apresenta, à esquerda, um índice para os produtos (somente os produtos 1 e 2 são apresentados nessa tabela). Os tamanhos de produto 63 e 71 referem-se ao tamanho da carcaça do motor elétrico. E nas colunas da direita, são apresentados os códigos de 8 dígitos utilizados para referência interna dos componentes na empresa-alvo. Nos exemplos mostrados, o componente selecionado pelo sistema-protótipo e pelo sistema atual são os mesmos, conforme se espera para uma codificação correta das regras.

Para os 10 produtos testados, foi possível verificar a concordância esperada, sendo que todos os componentes foram corretamente selecionados. A tabela completa com esses resultados pode ser encontrada no Apêndice A5.

5.3 NÚMEROS DO SISTEMA-PROTÓTIPO

Após a conclusão da implementação dos módulos, restrições e restrições tabulares, alcançaram-se os seguintes quantitativos que servem como referência do tamanho do protótipo elaborado para o presente estudo:

- i. 17 módulos (entre produto, componentes e conjuntos)
- ii. +200 propriedades, sendo 13 do tipo inteiro, 86 do tipo *double*, 34 do tipo booleano e +100 do tipo *String*;
- iii. 141 restrições, sendo 19 tabulares e as demais 122 expressões;

Algumas tabelas também alcançaram tamanho expressivo, por exemplo a tabela que representa a restrição intra-componente da caixa de ligação usinada, com 3752 linhas (diferentes itens) possíveis, ou a tabela que representa a restrição intra-componente da tampa traseira, totalizando 4266 variantes de componentes. Mesmo com essa massiva quantidade de

informações, a *shell* apresentou performance aceitável, realizando a propagação de restrições na ordem de 0,6s¹. Um tempo até 1s é apontado como o tempo máximo aceitável pelos usuários dos sistemas de configuração na empresa-alvo, conforme informação do especialista a ser discutida na seção 5.5, em termos de navegação e seleções sendo executadas no configurador de produto, considerando o processo iterativo de configuração explorado na Figura 4.20.

Segundos dados internos, no processo atual, a configuração de um produto leva em média 8 minutos, sendo necessário selecionar em média 14 campos. No novo processo proposto, esse tempo tem potencial de ser reduzido significativamente. Esse tempo total de configuração pode ser dividido entre duas parcelas, sendo uma para o tempo de processamento do sistema, e outra parcela que é de interação do usuário com o sistema.

Fez-se um teste comparando-se o desempenho do sistema atual e o sistema protótipo. O tempo de processamento durante a configuração foi medido considerando 10 processamentos e tirou-se a média desses processamentos. No sistema-protótipo, a média de tempo foi 0,6s, enquanto no sistema atual foram 0,7s.

Essa diferença no tempo de processamento não é perceptível pelo usuário, mas a vantagem no sistema-protótipo é a inexistência do processo final de validação onde são executadas as regras de consistência, vide seção 4.2.6, que consome um tempo de aproximadamente 12s – com a já mencionada desvantagem de que se tratava de um processo iterativo, sendo muitas vezes necessário retornar à etapa de configuração e alterar outros atributos. Considerando ganhos da eliminação da etapa de validação e melhorias nos atributos, estima-se que o tempo total para criar a configuração de um produto poderia ser reduzido pela metade – para 4 minutos. Computando a quantidade anual de 600 mil configurações de produto por ano executadas na empresa-alvo, é possível totalizar um ganho potencial de 40 mil horas. Uma eventual transição para um novo sistema não ocasionará um esforço substancial em treinamento, pois para o usuário final a interface se apresentará de forma muito similar à interface do sistema atual, devido à existência do nó de características simplificadas.

Outro aspecto positivo de um sistema baseado em restrições é que, com a constituição das restrições intra-componentes, representando as variantes de componentes, uma série de regras do sistema atual poderiam ser eliminadas. Um exemplo é detalhado no Apêndice A6, referente a uma regra existente no sistema atual que limita opções de anel de

1 – Especificação : Intel® Core™ i5-7200 CPU 2.5 Ghz RAM 8GB Windows 10 64 bits

fixação do rolamento em função de ter selecionado um tipo específico de flange. Esse tipo de regra torna-se desnecessária quando as opções de flange e anel de fixação derivam dos próprios componentes existentes.

Uma segunda situação em que é possível eliminar regras do sistema atual é em casos envolvendo bidirecionalidade nas regras. Por exemplo, no sistema atual, a seguinte regra “quando configurado kit ventilação forçada, necessário configurar refrigeração TFVF”, precisa da sua recíproca “quando configurado configuração refrigeração TFVF, necessário kit ventilação forçada”, ou seja, duas regras para indicar uma única relação. No protótipo, pode-se utilizar o operador *iff* (se e somente se) para casos desse tipo.

Nesses aspectos, não é possível calcular uma taxa precisa de eliminação de regras no processo de implantação de um sistema baseado em restrições, pois isso implicaria em construir o sistema completo, e o presente estudo se trata de um protótipo. Para fazer a parte mecânica do produto de forma completa, estima-se que seriam necessários aproximadamente mais 30 módulos, além de uma série de novas restrições. Este seria um trabalho factível para um colaborador em meio-período completar em aproximadamente um ano de trabalho (1000 horas de esforço envolvido).

5.4 RESULTADOS DA VERIFICAÇÃO

Foram elaborados também alguns estudos de caso para testar a validade do sistema-protótipo, bem como para explorar as possibilidades de uso. Dependendo do produto e companhia em estudo, as famílias de produto são entidades em constante evolução, sendo inseridas novas variantes dos componentes ao longo do ciclo de vida do produto.

Para o produto-alvo, essa é uma realidade constante, conforme o conceito de sistema aberto, discutido na seção 2.1.6. Somente para um componente, o eixo do motor elétrico, foram introduzidas mais de 1000 novas variantes de eixo por ano ao longo dos últimos 5 anos. Idealmente, essas novas variantes se tornariam opções selecionáveis para outros clientes que querem solicitar produtos idênticos ou muito similares, com a vantagem de eliminar o processo de desenvolvimento do componente caso um componente similar seja aceito pelo cliente. Contudo, o sistema de configuração de produto atual não tem essa capacidade.

No protótipo desenvolvido, por outro lado, todas essas variantes estão disponíveis no modelo do produto, sendo indiretamente selecionáveis através do nó “configurador

simplificado” pelos usuários de nível básico, e diretamente selecionáveis para os usuários de nível mais avançado através do nó “configurador completo”.

Ao tentar-se configurar um componente/produto ainda não existente, haverá um processo de avaliação submetida a um especialista da Engenharia, e este retornará uma resposta positiva ou negativa para a solicitação. A resposta positiva será o primeiro estudo de caso, denominado “acrescentar nova variante de módulo ao sistema-protótipo”. A resposta negativa será o segundo estudo de caso, denominado “restringir condição impossível”. As situações mencionadas nestes estudos de caso envolvem restrições intra-componentes, mas lógica idêntica é aplicável às restrições inter-componentes.

5.4.1 Estudo de caso 1: Acrescentar nova variante de módulo ao sistema-protótipo

Como discorrido na seção anterior, o acréscimo de novas variantes de um módulo é atividade cotidiana na empresa-alvo. Para simular esse caso de uso, segue-se o fluxo de ações apresentadas na Figura 4.26.

Antes de iniciar a interação com o sistema, é necessário tomar alguns cuidados do ponto de vista de construção das restrições. Para que funcione conforme o processo descrito a seguir, é necessário codificar as restrições da seguinte forma:

- i. A tabela de variantes de caixa de ligação, mostrada abaixo na Figura 5.6, deve possuir uma variante extra, mostrada na primeira linha da tabela da figura, que pode assumir qualquer valor em qualquer atributo (representado pelos valores com asterisco *). O item é ainda inexistente, sendo identificado por um traço. Isso indicará para as áreas subsequentes na empresa-alvo que, conforme explorado na Figura 4.2, se algum item do produto Motor estiver indicado com o traço, o produto terá de passar pelo processo de consulta técnica.

Figura 5.6 – Recorte da tabela de restrições intra-componente para a caixa de ligação

Item	DENOMINACAO CARÇA	LINHA PRODUTOS CX LIG	INCLINADA	BITOLA FURO ROSCA LADO A1	BITOLA FURO ROSCA LADO A2
*	*	*	*	*	*
= 10124161	= 00006 - 225/250	= 00001 - W22	= true	= 00138 - SEM	= 00138 - SEM
= 10594827	= 00006 - 225/250	= 00001 - W22	= true	= 00138 - SEM ⁴	= 00138 - SEM
= 10680886	= 00006 - 225/250	= 00001 - W22	= true	= 00100 - M50X1	= 00100 - M50X1
= 10680972	= 00008 - 280	= 00001 - W22	= true	= 00104 - M63X1	= 00104 - M63X1
= 10681007	= 00010 - 315	= 00001 - W22	= true	= 00104 - M63X1	= 00104 - M63X1

Fonte: Autoria própria

- ii. É necessário construir uma restrição adicional indicando que, por padrão, o valor do traço não será selecionado. Essa restrição recebe uma descrição conforme o seguinte texto: “Necessário avaliação da nova variante de caixa de ligação”, mensagem esta que auxiliará no processo de configuração. Para maiores detalhes do código, ver apêndice A7.

Toma-se como exemplo uma solicitação de cliente que deseja uma caixa de ligação com uma furação especial. A recomendação de marketing, para o mercado brasileiro, é utilizar um único furo roscado na caixa de ligação que segue a norma ISO 7-1 (*British Standard Pipe*), seguindo a Tabela 5.2, que é apresentada abaixo:

Tabela 5.2 – Tamanhos de rosca para caixa de ligação para o mercado brasileiro

Tamanho de carcaça	Rosca
63-100	ISO 7 – Rp ½”
112-132	ISO 7 – Rp 1”
160-180	ISO 7 – Rp 1 ½”
200	ISO 7 – Rp 2”

Fonte: Autoria própria

Como é possível visualizar, o padrão é somente um furo, independentemente do tamanho de carcaça do produto. No entanto, pode ser uma necessidade do cliente que haja mais furos para a ligação do motor, por exemplo, devido a uma maior quantidade de cabos de ligação, ou devido a uma necessidade de separar os cabos de alimentação, conforme mostrado na Figura 5.7:

Figura 5.7 – Motor com caixa de ligação com 3 furos na caixa de ligação



Fonte: Acervo interno (WEG)

Conforme mostrado na figura, há três prensa-cabos montados na caixa, um em cada furo, preparados para a situação de um cabo por prensa-cabos, e esta é uma solicitação do próprio cliente.

Na situação a ser simulada, solicita-se um motor da carcaça 132 em que haja três furos de 1”. Como trata-se de um motor trifásico, desta forma pode-se colocar o cabo de cada fase em um furo, atingindo um bom grau de isolamento entre o ambiente externo e o interior da caixa de ligação. A configuração do nó “configurador simplificado” é realizada selecionando apenas o atributo “Carcaça com complemento” com o valor 132S, conforme mostrado na Figura 5.8:

Figura 5.8 – Seleção inicial para o primeiro estudo de caso

Configurador simplificado close

Properties

Alimentação	TRIFASICO	Blindagem rolament...	ZZ	Blindagem rolament...	ZZ
Carcaça com compl...	132S	Com sensor Motor S...	false	Com sensor de vibra...	false
Direção de saída do...	PARA BAIXO	Eixo dupla ponta	false	entidade norma	IEC
Fixação do rolamento	SEM	Folga interna rolam...	NORMAL	Folga interna rolam...	NORMAL
Forma construtiva	B3D	Frequência	60Hz	Furo A1 da caixa de ...	ISO 7 - Rp ...
Furo A2 da caixa de ...	SEM	Furo A3 da caixa de ...	SEM	Furo A4 da caixa de ...	SEM
Grau de proteção	IP55	Graxa do rolamento	POLYREX EM	Invólucro	FECHADO
Linha de produtos	W22	Material da defletora	ACO CARB...	Material do eixo	SAE 1040/45
Material do invólucro	FERRO	Material do ventilador	POLIPROPL...	Mercado	BRASIL
Refrigeração do ma...	false	Tamanho do rolame...	6308	Tamanho rolamento...	6207
Tipo canal de chaveta	CHAVETA B	Tipo de graxeira	SEM GRAX...	Tipo e tamanho flan...	SEM FLAN...
Tipo vedação diante...	ANEL V	Tipo vedação traseira	ANEL V	terminalBoxModel	W22

Fonte: Autoria própria

Percebe-se que os furos A2 e A3 estão assinalados com o valor SEM, pois este é o padrão para a linha. Na sequência, realiza-se a seleção do valor ISO 7-Rp 1” no furo A2, conforme mostrado na Figura 5.9:

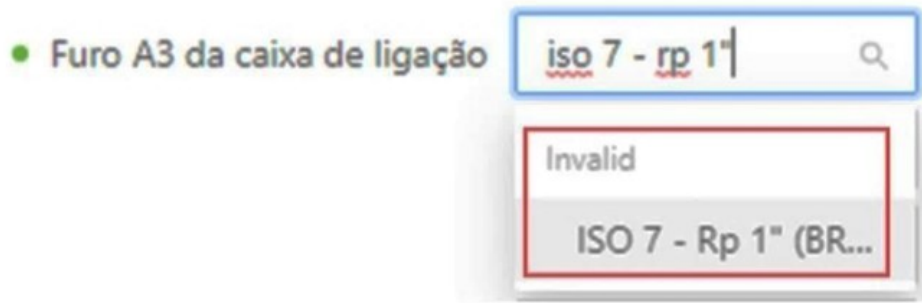
Figura 5.9 – Opções de furos para a posição A2 da caixa de ligação



Fonte: Autoria própria

Como se pode observar, esta seleção está entre os valores “Válidos” deste campo, o que significa que existe na base de variantes uma caixa com dois furos ISO 7-Rp 1”. Na sequência, seleciona-se o terceiro furo, mostrado a seguir na Figura 5.10:

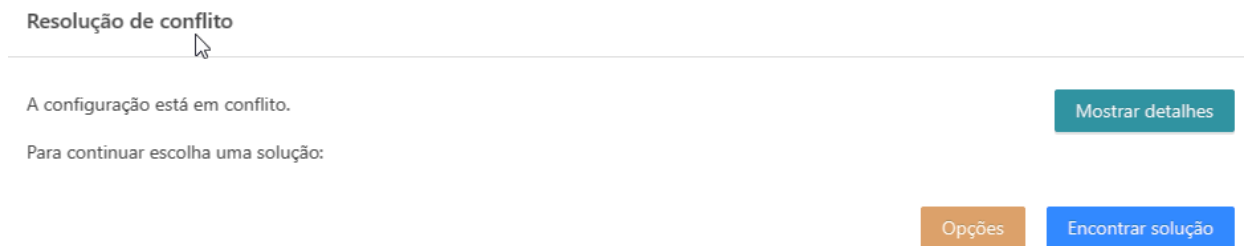
Figura 5.10 - Opções de furos para a posição A3 da caixa de ligação



Fonte: Autoria própria

No entanto, nota-se que esta seleção está como inválida, o que disparará o mecanismo de resolução de conflitos. A tela da Figura 5.11 apresenta que há um conflito na configuração, guiando o usuário para uma solução:

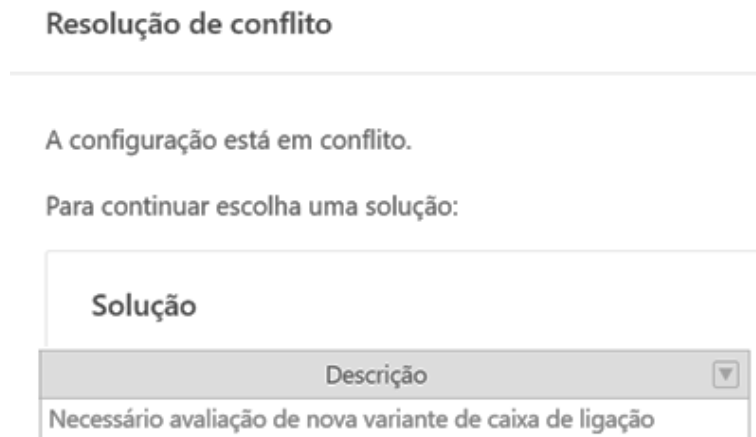
Figura 5.11 – Tela indicando a existência de um conflito na configuração



Fonte: Autoria própria

Após o usuário selecionar “Encontrar solução”, a *shell* processará da forma detalhada na seção 4.5.1.1, procurando um conjunto de restrições a serem desligadas que permitam a seleção, mostrada na Figura 5.12.

Figura 5.12 – Tela de resolução de conflitos indicando a restrição que deve ser desligada



Fonte: Autoria própria

No caso, como não há uma variante de módulo apta a essa configuração, a única restrição que pode ser desligada e que resolve o conflito é a restrição supramencionada que define que o valor “traço” não deveria ser utilizado. Mais detalhes podem ser encontrados no Apêndice A7. Ou seja, ao aplicar a solução proposta para o conflito, estarão configurados os três furos na tela do nó “configurador simplificado”, conforme Figura 5.13:

Figura 5.13 – Resultado após solução do conflito no nó “configurador simplificado”



Fonte: Autoria própria

No nó “caixa de ligação usinada”, o valor “traço” estará selecionado no campo *itemNumber*, e o valor “SOB CONSULTA” estará no campo estado do componente, segundo Figura 5.14:

Figura 5.14 – Resultado após solução do conflito no nó “caixa de ligação”

Caixa de ligação usinada ⊗ close

Properties

• Apto a placa de...	false	• Apto a proteçã...	false	• Bitola furo aterr...	M5
• Caixa inclinada	true	• Denominação c...	112/132	• Distância entre ...	0
• Distância entre ...	0	• Estado do com...	SOB CON...	• Furo A1 da caix...	ISO 7 - ...
• Furo A2 da caix...	ISO 7 - ...	• Furo A3 da caix...	ISO 7 - ...	• Furo A4 da caix...	SEM
• Furo B1 da caix...	SEM	• Furo C1 da caix...	SEM	• Furo C2 da caix...	SEM
• Furo D1 da caix...	SEM	• Furo D2 da caix...	SEM	• Lado fixação co...	SEM
• Linha de produt...	W22	• Posição base da...	NAO A...	• Quantidade de ...	1
• Quantidade de ...	0	• itemNumber	-		

Fonte: Autoria própria

Com esta informação, a configuração corrente pode ser gravada e passada adiante no fluxo interno da empresa-alvo. Devido ao fato dessa configuração possuir um componente cujo número de item está indefinido e assinalado com o valor “SOB CONSULTA”, será disparado o processo de consulta técnica, sendo necessário que um especialista no domínio avalie a possibilidade de incluir um terceiro furo na caixa. No caso específico dessa avaliação, ela é realizada utilizando softwares CAD e o conhecimento prévio do especialista.

A partir do processo de consulta, duas respostas são possíveis. Em caso de ser possível produzir o componente, ele será criado com um código de item único (*itemNumber*) e cadastrado na tabela de variantes de caixa de ligação. No entanto, caso não seja possível, será necessário proceder de maneira diferente. No estudo de caso corrente, assume-se que essa impossibilidade implicará em não criar o item, mas simplesmente não criar o item e não registrar a impossibilidade de sua criação não é a situação ideal, pois uma solicitação idêntica no futuro implicará em repetição de atividades internas na empresa-alvo. Para lidar com essa situação, no próximo estudo de caso, será explorado como registrar a impossibilidade de produzir determinado componente, evitando essa repetição de avaliações.

Retomando o caso de ser possível produzir o componente, proceder-se-á da seguinte forma:

- i. O especialista em domínio avaliará o componente; no caso específico citado, avaliará o espaço disponível para executar três furos na caixa de ligação. É possível, nesse ponto, construir algoritmos ou rotinas pré-definidas de cálculo, envolvendo as mais diversas técnicas, a fim de automatizar essa etapa.
- ii. O especialista fará o cadastro do novo item, com um novo código de item. Por simplicidade, supor-se-á que o código será 1700000, conforme Figura 5.15:

Figura 5.15 – Cadastro da caixa com três furos

Item	DENOMINACAO CARC	LINHA PRODUTOS	INCLINADA	BITOLA FURO ROSCA LADO A1	BITOLA FURO ROSCA LADO A2	BITOLA FURO ROSCA LADO A3
= 17000000	= 00001 - 112/132	= 00001 - W22	= true - True	= 00080 - ISO 7 - Rp 1" (BROCA 30.5)	= 00080 - ISO 7 - Rp 1" (BROCA 30.5)	= 00080 - ISO 7 - Rp 1" (BROCA 30.5)

Fonte: Autoria própria

- iii. Retorna-se a resposta para o cliente, indicando a viabilidade da solicitação.

Nesse caso, uma nova configuração com a mesma solicitação não incorrerá na tela de resolução de conflitos, sendo possível selecionar os valores dos 3 furos como ISO 7 – Rp 1”. Destaca-se que, após o cadastro, ao navegar até o nó da caixa de ligação, o *itemNumber* selecionado é o 1700000, conforme mostrado na Figura 5.16:

Figura 5.16 – Nó caixa de ligação usinada com item 17000000 selecionado

Caixa de ligação usinada

⊗ close

Properties

• Apto a placa de...	false	• Apto a proteçã...	false	• Bitola furo aterr...	M5
• Caixa inclinada	true	• Denominação c...	112/132	• Distância entre ...	0
• Distância entre ...	0	• Estado do com...	LIBERADO	• Furo A1 da caix...	ISO 7 - ...
• Furo A2 da caix...	ISO 7 - ...	• Furo A3 da caix...	ISO 7 - ...	• Furo A4 da caix...	SEM
• Furo B1 da caix...	SEM	• Furo C1 da caix...	SEM	• Furo C2 da caix...	SEM
• Furo D1 da caix...	SEM	• Furo D2 da caix...	SEM	• Lado fixação co...	SEM
• Linha de produt...	W22	• Posição base da...	NAO A...	• Quantidade de ...	1
• Quantidade de ...	0	• itemNumber	17000000		

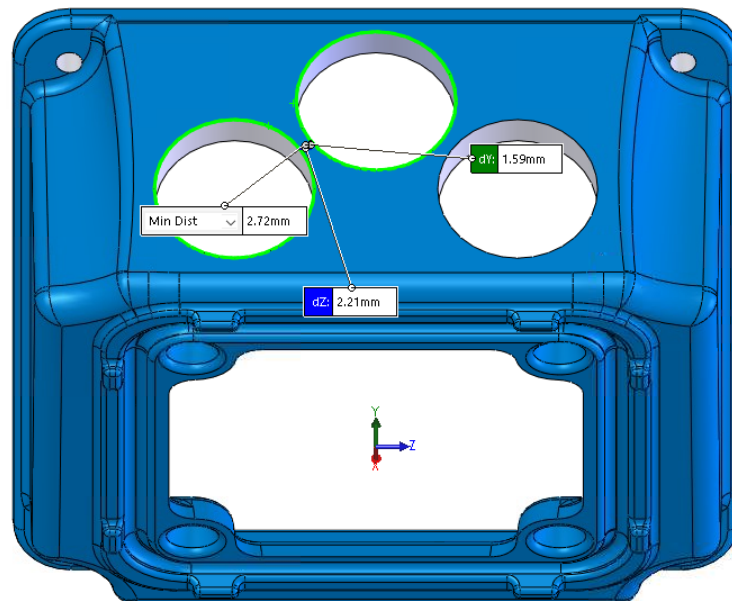
Fonte: Autoria própria

Esse mesmo processo pode ocorrer para outros componentes ou situações de restrições inter-componentes, seguindo processo similar de avaliação e cadastro nas tabelas correspondentes. Por fim, após o cadastro, a configuração torna-se possível, sem que se incorra na resolução de conflitos.

5.4.2 Estudo de caso 2: Restringir condição impossível

Situações impossíveis são comuns em diversos tipos de produto. Essas limitações podem dar-se devido às limitações físicas e/ou geométricas, ou ainda no escopo de projeto mecânico, questões estruturais, de fabricação ou térmicas. Para a caixa de ligação descrita na seção anterior com três furos, a limitação estrutural ou de fabricação inviabiliza o componente. Na Figura 5.17, é possível visualizar uma possível disposição para os três furos na caixa de ligação, mas a distância entre os furos é reduzida demais e isso inviabiliza a usinagem dos furos.

Figura 5.17 – Caixa de ligação com três furos



Fonte: Acervo interno (WEG)

Esse seria um caso de resposta negativa de um processo de consulta técnica, pois o especialista avaliou a impossibilidade no fornecimento dessa configuração em específico. Para emular esse processo no protótipo, é necessário codificar uma restrição adicional às mencionadas no estudo de caso anterior.

Para a combinação de roscas citadas em um escopo similar (neste caso, mesmo tamanho de componente), é necessário acrescentar uma restrição que indique que o atributo “estado de componente” deve assumir o valor “IMPOSSÍVEL” na combinação de atributos que representa o componente hipotético com três furos.

As demais etapas da configuração acontecem de maneira idêntica às operações descritas no estudo de caso anterior, com a diferença que, ao configurar as três roscas, será visível na Figura 5.18, que o *itemNumber* aparecerá como indefinido, visto que não há componente para esta combinação, e o atributo “estado de componente” recebeu o valor “IMPOSSÍVEL”:

Figura 5.18 – Nó caixa de ligação com configuração impossível

Caixa de ligação usinada

⊗ close

Properties

• Apto a placa de bor...	false	▼	• Apto a proteção tér...	false	▼	• Bitola furo aterrame...	M5	▼
• Caixa inclinada	true	▼	• Denominação carca...	112/132	▼	• Distância entre furos...	0	⊗ ✓
• Distância entre furos...	0	⊗ ✓	• Estado do compone...	IMPOSSÍVEL	▼	• Furo A1 da caixa de ...	ISO 7 - Rp ...	▼
• Furo A2 da caixa de ...	SEM	▼	• Furo A3 da caixa de ...	SEM	▼	• Furo A4 da caixa de ...	SEM	▼
• Furo B1 da caixa de l...	SEM	▼	• Furo C1 da caixa de ...	SEM	▼	• Furo C2 da caixa de ...	SEM	▼
• Furo D1 da caixa de ...	SEM	▼	• Furo D2 da caixa de ...	SEM	▼	• Lado fixação conector	SEM	▼
• Linha de produtos	W22	▼	• Posição base da caix...	NAO APLI...	▼	• Quantidade de furos...	1	▼
• Quantidade de furos...	0	▼	• itemNumber	-	▼			

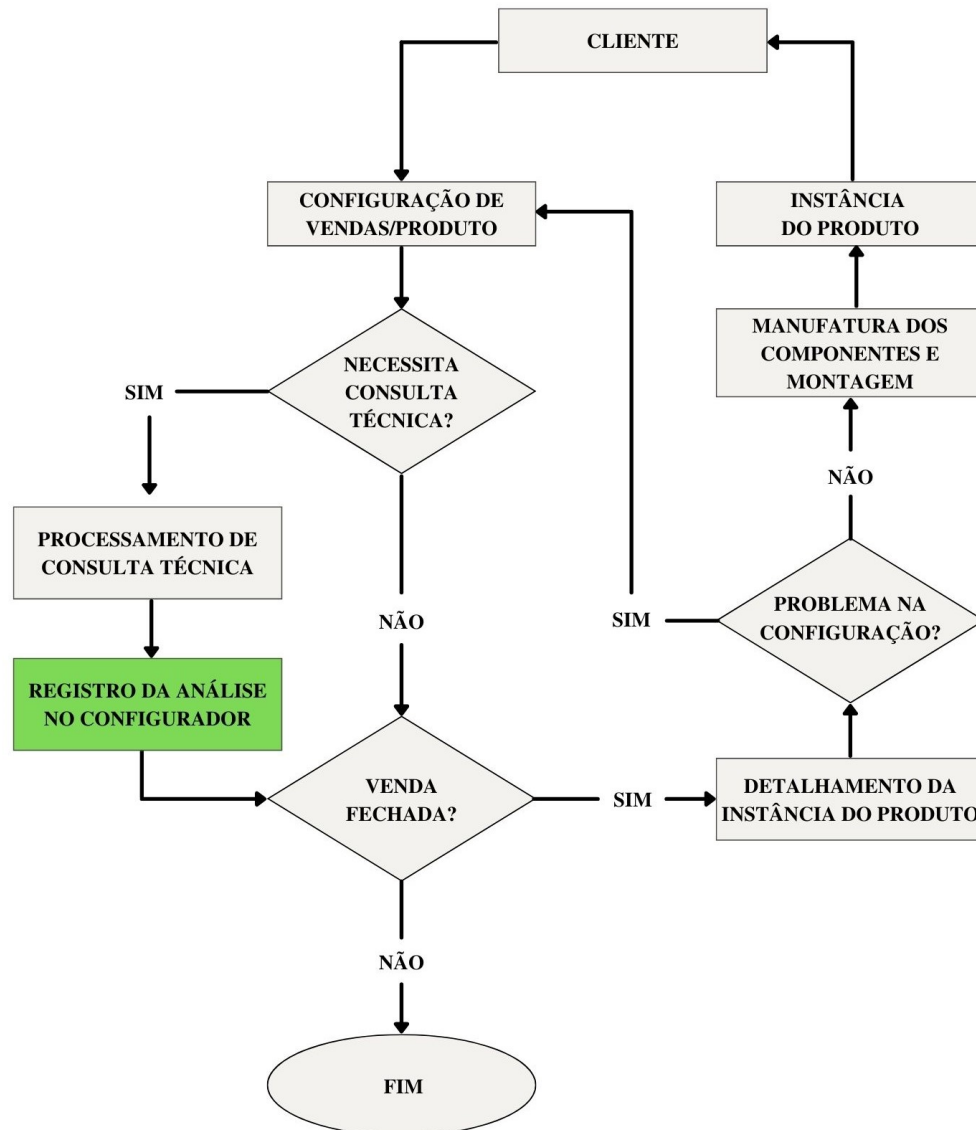
Fonte: Autoria própria

Nesta situação, a configuração não poderá prosseguir, e o usuário será imediatamente informado de que não poderá receber este produto, sendo necessário informar o cliente imediatamente sobre a situação.

5.4.3 Alteração no processo de consulta técnica na empresa-alvo

Para incorporar o registro da análise no configurador, propõe-se que o processo apresentado na Figura 4.2 será revisto, incluindo o registro do resultado análise no sistema-protótipo de configurador de produto, conforme ressaltado na Figura 5.19:

Figura 5.19 - Fluxo de processamento proposto de pedidos na empresa-alvo



Fonte: Autoria própria

Na figura, é possível visualizar que se altera o processo de consulta técnica, incorporando o resultado da avaliação da consulta técnica no próprio sistema de configuração. Passar-se-ia a ter adição do conhecimento de especialistas do produto ao longo do tempo no sistema, o que representa um dos outros ganhos previstos com a adoção do sistema-protótipo.

5.5 VALIDAÇÃO DO SISTEMA-PROTÓTIPO

Conforme exposto no início do capítulo, convidou-se um especialista da empresa-alvo para realizar a validação do sistema. O especialista em configuração de produto é um

colaborador da empresa-alvo com quinze anos de experiência, graduado e pós-graduado em áreas correlatas. Essa validação consistiu em entregar o sistema para testes por um período de duas semanas, de modo que foi possível percorrer tópicos similares aos tópicos da verificação, bem como outros aspectos que não haviam sido considerados somente do ponto de vista de engenharia do conhecimento.

Os critérios utilizados para a validação foram baseados em Silva et al. (2014), que consistem em tabelar uma pontuação e comentários do especialista. Essa pontuação varia de 1 a 5, sendo 1 um nível ruim e 5 um nível excelente de atendimento do critério. Como no presente estudo comparou-se um sistema existente com o sistema-protótipo, na tabela de avaliação também foi criada uma pontuação para os dois sistemas.

Os critérios extraídos de Silva et al. (2014) foram adaptados para a realidade do sistema atual, sendo incluído dois critérios que envolvem o administrador de regras (manutenção fácil e quantidade de regras necessárias), pois os critérios originais não levavam em consideração o ponto de vista do administrador do sistema.

Os critérios são: uso fácil, o que envolve como o sistema é apresentado para o usuário final; manutenção fácil, que envolve aspectos de expansão, construção das regras por parte do administrador do sistema; capacidade de lidar com situações de conflito, que envolve a codificação de regras para situações restritas e como isso é apresentada para os usuários finais; a clareza do conhecimento codificado sob a forma de regras; a quantidade de regras necessárias para modelar um volume similar de conhecimento; por fim, o tempo de processamento durante a configuração, aspecto que impacta diretamente o usuário final.

A seguir, na Tabela 5.3, são apresentadas as pontuações dadas pelo especialista e seus comentários:

Tabela 5.3 – Avaliação do especialista

Característica	Sistema atual	Comentários sistema atual	Sistema protótipo	Comentários sistema protótipo
Uso fácil (usuário final)	3	Sem comentários	4	Sem comentários
Manutenção fácil (administrador)	3	As ferramentas nativas do sistema são bastante limitadas, mas a possibilidade de baixar os dados, alterar e carregar no sistema torna a sua manutenção mais fácil.	3	A tecnologia utilizada permite evoluir muito em relação à manutenção, mas ainda faltam ferramentas "low-code" para este fim.
Capacidade de lidar com situações de conflito	2	A falta de informações dos componentes dificulta a identificação de conflitos.	4	O sistema sugere possíveis soluções para os conflitos, mas existem situações pontuais onde a sugestão não é a mais adequada.
Clareza das regras	3	Sem comentários	4	Sem comentários
Quantidade de regras necessárias	1	O sistema atual não permite o reaproveitamento de regras, ocasionando a replicação dos registros.	5	O sistema permite que as regras sejam criadas de diversas formas, ficando sob a responsabilidade do usuário utilizar a melhor alternativa para simplifica-las.
Tempo de processamento	3	Apesar do baixo tempo de processamento durante a configuração, há um grande tempo de espera na gravação, pois nesse momento são executadas diversas regras para garantir a consistência da configuração.	4	Sem comentários

Fonte: Autoria própria

Nota-se de imediato que as pontuações obtidas pelo sistema-protótipo foram melhores que o sistema atual em todos os aspectos.

No aspecto de uso fácil, houve uma pequena melhora na pontuação, mas entende-se que como se tentou manter atributos e descrições similares às atuais, realmente não houve grandes avanços; a melhoria de um ponto provavelmente se deu pelo visual mais moderno da ferramenta.

No aspecto de manutenção, o sistema atual é bastante limitado de forma nativa, mas diversas automações foram construídas ao longo de anos de uso, permitindo contornar certas

limitações através do tratamento de dados fora da ferramenta. No sistema-protótipo, devido até o estágio incompleto da *shell*, não apresenta todas as ferramentas do sistema atual. Por outro lado, a codificação superior do conhecimento facilita o processo, o que permitiu que o especialista pontuasse as duas ferramentas como equivalentes.

No aspecto de lidar com conflitos, o especialista ressaltou a falta de informações sobre os componentes como uma das principais limitações do sistema atual. Esse aspecto foi superado no sistema-protótipo, pois como explorado nas seções anteriores, as variantes dos componentes são representadas por restrições tabulares.

No aspecto de clareza das regras, entende-se que houve uma pequena evolução, devido à organização da estrutura em níveis.

No aspecto da quantidade de regras necessárias, houve um grande avanço, representado pela pontuação dada pelo especialista. Foi ressaltado o baixo reaproveitamento de regras no sistema antigo, e a possibilidade de fazer as regras no sistema-protótipo de diversas formas, havendo aí um aspecto de curva de aprendizado do lado do administrador do sistema.

No aspecto desempenho, há que se ressaltar o comentário deixado pelo especialista. O sistema atual é de fato igualmente rápido para uso em tempo de configuração, mas sempre apresentará um tempo de gravação/consistências ao final do processo, tempo este inexistente no sistema sob o novo paradigma.

Considerando os comentários e pontuações do especialista, é possível depreender que o desenvolvimento do sistema-protótipo alcançou seus objetivos, representando um avanço em diversas frentes para o sistema atual. Algumas situações ainda podem ser evoluídas, principalmente no que tange à própria *shell*, o que foge do escopo do presente trabalho.

Neste capítulo, são apresentados brevemente os resultados do protótipo do sistema de configuração de produto, que foi verificado em casos reais da empresa-alvo. Inicialmente, foi apresentada a interface do sistema-protótipo e como se daria sua utilização, destacando aspectos como a árvore de módulos.

Na sequência, foram avaliadas as restrições construídas através de testes com uma amostra de produtos variados da linha-alvo de produtos. Alguns números do sistema-protótipo também foram apresentados, evidenciando a sua extensão e aspectos não-funcionais como o desempenho.

Apresentaram-se também dois estudos de caso, envolvendo uma modificação no processo de consulta técnica, válido para situações de restrições intra ou inter-componentes que representam casos possíveis ou impossíveis para o produto.

Por fim, repassou-se o sistema para que um especialista com quinze anos de experiência avaliasse o protótipo, o que foi bastante positivo como validação da proposta. Alcançou-se uma melhor pontuação em comparação com o sistema atual na maioria dos aspectos investigados.

No capítulo a seguir, fechar-se-á o trabalho, concluindo as etapas do método de desenvolvimento de um sistema especialista para configuração de produto, e trazendo propostas para trabalhos futuros.

6 CONCLUSÕES

Os resultados apresentados pelo protótipo do sistema de configuração baseado em restrições são bastante satisfatórios, comprovando que é possível codificar tanto as restrições de produto, manufatura e marketing, bem como sinalizadores que auxiliam nos processos internos de uma companhia. O atributo “estado do componente”, por exemplo, foi utilizado para indicar se um componente sendo selecionado durante a configuração assumiria os valores “liberado” (que permite a produção imediata do produto), “sob consulta” (que exige a avaliação de um especialista) e “impossível” (que já passou pela avaliação de um especialista que considerou o produto inviável).

De um ponto de vista dos objetivos iniciais do presente trabalho, considera-se tê-los alcançado.

O objetivo (i) foi desenvolvido ao longo dos capítulos 4 e 5, executando-se as etapas previstas na teoria de construção de sistemas especialistas. Os passos apresentados podem ser reaproveitados para outros produtos altamente configuráveis. Os objetivos (ii) e (iii) diziam respeito justamente à configuração de componentes mecânicos, e após o protótipo desenvolvido, entende-se que o domínio mecânico é de mais fácil visualização em comparação com outros domínios do conhecimento como o domínio elétrico, pois representam geralmente variáveis de natureza geométrica ou relacionadas à manufatura. No exemplo explorado no segundo estudo de caso, foi possível representar visualmente a impossibilidade de usinar três furos no componente.

Os objetivos (iv) e (v) propunham a construção do protótipo e aplicação em estudos de caso, o que também se considera atingido. Por fim, a discussão sobre o objetivo (vii) vem com uma reflexão sobre as limitações na modelagem do conhecimento que a própria ferramenta existente impunha. A não-integração entre variantes existentes no ERP da empresa-alvo e a configuração, a não-possibilidade de construir bidirecionalidade com as regras levavam a duplicação de informações, erros e retrabalhos, o que justifica a adoção de um novo paradigma de programação.

No entanto, é importante ressaltar que mesmo a flexibilidade fornecida no sistema-protótipo não é ilimitada – e isso continua coerente com os conceitos de customização em massa apresentados no começo do trabalho. Somente as variantes de módulo que não acrescentem novas características são relativamente fáceis de incluir; casos mais complexos,

tipicamente do tipo *ETO*, exigirão uma revisão total da modelagem e das restrições, não tendo sido abordados nos estudos de caso, pois fogem do escopo do trabalho.

No estudo com a empresa-alvo, notou-se que já havia elevado grau de codificação de conhecimento em sistemas existentes, mas com certa sobreposição de funções entre diferentes sistemas, além de codificação ineficiente desse conhecimento. Ao se introduzir um novo conceito de sistema de configuração de produto, abre-se a oportunidade para que especialistas nos sistemas atuais possam contribuir na construção de um sistema mais robusto, superando falhas nas modelagens antigas.

Por mais que já exista um sistema especialista em uso para configuração de produto na empresa-alvo, a construção de um novo sistema sempre impõe seus próprios desafios. Nesse momento, a atuação de um especialista da área continua sendo fundamental, pois é necessário que essa pessoa decodifique os conhecimentos que estão em um modelo antigo, muitas vezes representados de forma parcial devido a limitações das ferramentas antigas. A atuação do engenheiro do conhecimento, nesse ponto, se concentra em recodificar o sistema sob novos paradigmas.

6.1 SUGESTÃO PARA TRABALHOS FUTUROS

Não há pretensão de esgotar o tema explorado neste trabalho, pois existem diversos aspectos que podem ser aprofundados em trabalhos futuros. A seguir, apresentam-se algumas alternativas de futuras discussões, dando continuidade ao presente estudo.

i. Expansão do sistema

Como o sistema desenvolvido é um protótipo, abrangendo uma linha de produto, com componentes limitados e com enfoque no subsistema mecânico do motor elétrico, uma sugestão natural seria a expansão do sistema para as demais linhas de produto. O desafio nesse caso seria um esforço de replicação e levantamento de dados, sem a necessidade de explorar novos conceitos, pois os conceitos fundamentais foram comprovados com o presente trabalho.

ii. Sistema de configuração de terceira geração

Do ponto de vista acadêmico, alguns conceitos encontrados na literatura poderiam ser adotados em um trabalho futuro. Por exemplo, o conceito de sistemas de configuração pré-compilados (denominados de terceira geração na seção 3.4.1), que permitem performance superior mesmo com grandes volumes de dados, seria uma alternativa a ser considerada,

especialmente se considerados a expansão do sistema conforme discutido. Como o protótipo ainda apresentou bom desempenho, na ordem de 0,6s por execução, não houve a necessidade de estudar essa tecnologia no presente trabalho. Contudo, em uma aplicação mais complexa, que inclua maior quantidade de restrições e módulos, o desempenho pode inviabilizar a utilização do sistema, sendo necessário evoluir a *shell* para incluir um mecanismo de compilação. No entanto, de um ponto de vista de conceitos de modelagem de produto e restrições, não haveria impactos.

iii. Identificação de padrões

A emergência da subárea de reconhecimento de padrões na Inteligência Artificial também permite vislumbrar algumas possibilidades de trabalhos futuros. Como existem uma grande quantidade combinações de componentes e atributos especiais que podem ser solicitadas, seria interessante se fosse possível automatizar parte deste trabalho utilizando reconhecimento de padrões. Por exemplo, se a combinação de três furos de determinado tamanho só se torna possível a partir de certo tamanho de carcaça do motor, uma rede neural poderia “aprender” esse padrão após algumas consultas respondidas, sendo possível responder casos similares (mas ainda não consultados) utilizando a própria rede neural. Tal trabalho, evidentemente, teria de lidar com a natureza não-numérica do problema de configuração aqui explorado, tendo que comprovar aceitável precisão nas suas respostas. Esse aspecto seria um desafio interessante para a área de Redes Neurais, pois muitos trabalhos nessa área são focados em variáveis numéricas.

iv. Integração dos cálculos de aplicação ao configurador de produto

O sistema de cálculo de aplicação apresentado brevemente na Figura 4.7 denominado *iMech* é apenas um dos vários sistemas similares que são utilizados na etapa de consulta técnica na empresa-alvo para os mais diversos fins mecânicos (por exemplo, cálculo de vida e relubrificação de mancais, fator de segurança do eixo submetido às cargas, etc..). Esses sistemas estão codificados de forma procedural em programação tradicional, seguindo rotinas consagradas na literatura, neste caso, de Elementos de Máquina.

Ao invés de buscar automatizar o disparo desse cálculo, seria interessante que um trabalho futuro avaliasse a viabilidade de integrar o cálculo diretamente ao configurador, convertendo as equações em restrições. Dessa maneira, ter-se-ia respostas imediatas a solicitações do cliente, sob a forma da limitação das variáveis do modelo. Por exemplo, uma carga causada por uma polia na ponta de eixo poderia limitar diâmetros e comprimentos disponíveis de ponta de eixo do motor.

v. Aplicação em outros produtos e domínios do conhecimento

É possível aplicar o mesmo método de modelagem do sistema especialista para outros produtos ou domínios do conhecimento. Como mencionado no início deste capítulo, do ponto de vista da parte mecânica de um produto, pode haver uma maior facilidade em visualizar as interações entre variáveis, pois elas geralmente representam aspectos geométricos, físicos e de fabricação do produto. No exemplo explorado na seção 5.4.2, foi possível visualizar a impossibilidade de alocar três furos na face da caixa de ligação mantendo uma distância razoável entre os furos. Em outros domínios, certas restrições podem ser mais difíceis de modelar e visualizar. Para um configurador completo de um motor elétrico, ainda faltaria modelar a parte elétrica do mesmo. Outros exemplos de produtos industriais comuns que podem ser customizados em massa são redutores, bombas, compressores, trocadores de calor e válvulas, podendo se beneficiar da adoção de sistemas de configuração de produto em outras companhias. Alguns deles geralmente são puramente mecânicos (como redutores e trocadores de calor mais simples) e outros podem precisar de outros domínios de conhecimento, podendo ser utilizando o conceito de subsistema discutido na seção 2.1.5.3 para organizar a modelagem.

vi. Integração do sistema de configuração de produto ao sistema ERP

Algumas informações tipicamente armazenadas em sistemas ERP como preço, custo, estoque e prazo de entrega de um componente podem ser associados ao sistema de configuração de produto para oferecer informações mais completas ao usuário do sistema. Tais informações podem balizar certas escolhas, como optar por um rolamento blindado com um tipo de graxa especial, para evitar um prazo de entrega muito grande. Esse tipo de informação também ajuda do ponto de vista de manufatura enxuta, pois se o usuário do sistema pode configurar filtrando itens que existem no estoque, contribui-se para manter estoques mais reduzidos.

REFERÊNCIAS

ANADIOTIS, G. **DeepMind aims to marry deep learning and classic algorithms**. 2021. Disponível em: <<https://venturebeat.com/2021/09/10/deepmind-aims-to-marry-deep-learning-and-classic-algorithms/>>. Acesso em: 10 de fevereiro de 2022.

BACK, N., OGLIARI, A., DIAS, A., SILVA, J. C. **Projeto integrado de produtos – planejamento, concepção e modelagem**. Barueri: Manole, 2008.

BĂDICĂ, C., LEON, F., BĂDICĂ, A. **Freight transportation broker agent based on constraint logic programming**. *Evolving Systems*, n.11, p. 363-382, 2020.

BAKER, A., FISCHER, A., LEWIS, C. **The customer’s workshop: the future of American manufacturing**. 1ª ed. 2007.

BANERJEE, S., KANKARIA, G. H., KT, S. **From forecasting demand to ordering – An automated machine learning approach with Amazon Forecast to decrease stockouts, excess of inventory and costs**. *AWS Machine Learning Blog*, 12 março de 2021. Disponível em <<https://aws.amazon.com/pt/blogs/machine-learning/from-forecasting-demand-to-ordering-an-automated-machine-learning-approach-with-amazon-forecast-to-decrease-stock-outs-excess-inventory-and-costs/>>. Acesso em: 05 de dezembro de 2021.

BEEK, P. Backtracking Search Algorithm. In: ROSSI, F., BEEK, P., WALSH, T. **Handbook of Constraint Programming**. Elsevier: 1ª ed. 2006.

BERTSIMAS, D., MISIC, V. V. **Exact First-Choice Product Line Optimization**. *Operations Research*, v.67, n. 3, p. 651-670, 2019.

BESSIERE, C. Constraint Propagation. In: ROSSI, F., BEEK, P., WALSH, T. **Handbook of Constraint Programming**. Elsevier: 1ª ed. 2014.

BONEV, M., HVAM, L., CLARKSON, J., MAIER, A. **Formal computer-aided product Family architecture design for mass customization**. Computers in Industry, n. 74, p. 58-70, 2015.

BOTEGA, L. F. C. **Knowledge-based System for Categorization and Selection of Creativity Support Techniques in Design**. 2016. (Dissertação de mestrado). Pós-graduação em Engenharia Mecânica, UFSC, Brasil.

BRUUN, H. P. L., MORTENSEN, N. H., HARLOU, U., WÖRÖSCH, M., PROSCHOWSKY, M. **PLM system support for modular product development**. Computers in Industry, 2014.

CAO, J., BUCHER, D. F., HALL, D. M., LESSING, J. **Cross-phase product configurator for modular buildings using kit-of-parts**. Automation in Construction, v. 123, 2021.

CHOCO-SOLVER. **An Open-Source java library for constraint programming**. 2021. Disponível em <<https://choco-solver.org/>>. Acesso em: 06 de dezembro de 2021.

CONFIGIT. **Integration of Configit ACE® and Aras Innovator**. 2020.

COPPIN, B. **Inteligência Artificial**. Tradução Valério, Jorge Duarte Pires. Rio de Janeiro: LTC, 2013.

DAABOUL, J., DA CUNHA, C., BERNARD, A., LAROCHE, F. **Design for mass customization: Product Variety vs. process variety**. CIRP Annals – Manufacturing Technology (60), p. 169-174, 2011.

DICKSON, B. **Common sense is a huge blind spot for AI developers - Building bigger neural networks won't lead us artificial general intelligence**. 2021. Disponível em: <<https://thenextweb.com/news/common-sense-is-a-huge-blind-spot-for-ai-developers>>. Acesso em: 10 de fevereiro de 2022.

DING, G. L. **Recent developments in simulation techniques for vapour-compression refrigeration systems.** *International Journal of Refrigeration*, v. 30, n. 1, p. 1119–1133, 2007.

DURAN, O., PÉREZ, L., BATOCCHIO, A. **Optimization of modular structures using Particle Swarm Optimization.** *Expert System with Applications*, v. 39, p. 3507–3515, 2012.

FALKNER, A., FRIEDRICH, G., HASELBÖCK, A., SCHENNER, G., SCHREINER, H. **Twenty-five years of successful application of constraint Technologies at Siemens.** *AI Magazine*, p. 67-80, 2016.

FELFERNIG, A., REITERER, S., REINFRANK, F., NINAUS, G., JERAN, M. **Conflict detection and diagnosis in Configuration.** In: FELFERNIG, A., HOTZ, L., BAGLEY, C., TIIHONEN, J. **Knowledge-Based Configuration.** Elsevier: 1^a ed. 2014.

FORGY, C. L. **Rete: A fast algorithm for the many pattern/many object pattern match problem.** *Artificial Intelligence*, p. 17-37, 1982.

FORZA, C., TRENTIN, A. **Supporting product configuration and form postponement by grouping components into kits: the case of MarelliMotori.** *International Journal of Mass Customisation*, Vol. 1, No. 4, p. 427-444, 2006.

FRANCO, C, R. **Inteligência Artificial.** Londrina: Editora e Distribuidora Educacional S.A, 2014.

FREUDER, E. C., MACKWORTH, A. K. **Constraint Satisfaction: An Emerging Paradigm.** In: ROSSI, F., BEEK, P., WALSH, T. **Handbook of Constraint Programming.** Elsevier: 1^a ed. 2014.

GALBUSERA, F., CASAROLI, G., BASSANI, T. **Artificial intelligence and machine learning in spine research.** *JOR Spine*, 2019.

GIOVANNINI, A., AUBRY, A., PANETTO, H., HAOUZI, H., PIERREL, L., DASSISTI, M. **Approach for rationalisation of product lines variety**. 19th IFAC World Congress, Cape Town, **South Africa**, p. 24-29, 2014.

GIARRATANO, J. C., RILEY, G. **Expert Systems: Principles and Programming**. 4. Thomson Course Technology, 2005.

GOOGLE. **About OR-Tools**. 2020. Disponível em <<https://developers.google.com/optimization/introduction/overview>>. Acesso em: 06 de janeiro de 2022.

GUO, S., CHOI, T., CHUNG, S. **Self-design fun: Should 3D printing be employed in mass customization operations?** European Journal of Operational Research, 2021.

HANSEN et al. **Calculation of complexity costs – An approach for rationalizing a product program**. NordDesign 2012, Aalborg, Denmark, 2012.

HASELEIN, B. Z. **Sistema especialista para verificação da conformidade de motores elétricos** (Dissertação de Mestrado). Programa de Pós-Graduação em Engenharia Mecânica. UFSC, Florianópolis. 2018.

HASELEIN, B. Z., SILVA, J. C. **Knowledge-based system for electric motors compliance verification: a design for mass customization scenario**. 25th ABCM International Congress of Mechanical Engineering, Uberlândia, 2019.

HENDLER, J., KITANO, H., NEBEL, B. **Handbook of Constraint Programming**. Elsevier: 1^a ed. 2006.

HENDLER, J. Avoiding Another AI Winter. **IEEE Intelligent Systems**, v. 23(2), p. 2-4, 2008.

HOTZ, L., FELFERNIG, A., GÜNTER, A., TIIHONEN, J. A Short History of Configuration Technologies. In: FELFERNIG, A., HOTZ, L., BAGLEY, C., TIIHONEN, J. **Knowledge-Based Configuration**. Elsevier: 1^a ed. 2014.

HULGAARD, H. **Virtual Tabulation – the 3. Generation Configuration Technology**. 2019. Disponível em <<https://configit.com/virtual-tabulation/>>. Acesso em: 09 de fevereiro de 2022.

JAIN, J., **Overview of SAP planning strategy**. SAP Blog community, 2014. Disponível em: <<https://blogs.sap.com/2014/09/16/overview-of-sap-planning-strategy/>>. Acesso em: 26 de fevereiro de 2022.

JIAO, J., TSENG, M. M. **Customizability analysis in design for mass customization**. Computer-Aided Design (36), p. 745-757, 2004.

JUNKER, U. Configuration. In: ROSSI, F., BEEK, P., WALSH, T. **Handbook of Constraint Programming**. Elsevier: 1^a ed. 2006.

LEE, C., LEEM, C. S., HWANG, I. **PDM and ERP integration methodology using digital manufacturing to support global manufacturing**. International Journal of Advanced Manufacturing Technology (53), p. 399-409, 2010.

LEHTONEN, T. **Designing modular product architecture in the new product development**. (Tese de Doutorado). Tampere University of Technology, Tampere, 2007.

LI, Z., FENG, Y., TAN, J., WEI, Z. **A Methodology to Support Product Platform Optimization using Multi-Objective Evolutionary Algorithms**. Institute of Measurement and Control, p-295-312, 2008.

LI, H., JI, Y., LUO, G., MI, S. **A modular structure data modeling method for generalized products**. International Journal of Advanced Manufacturing Technology, p-197-212, 2015.

LIKER, J. K. **The Toyota way: 14 Management Principles From the World's Greatest Manufacturer**. McGraw-Hill: 1ª ed. 2004.

LOVÁSZ, L. **Combinatorial problems and exercises**. AMS Chelsea Publishing: 2ª ed. 2007.

MASSIRER, D. A. **Sistema Especialista Protótipo para Auxílio à Verificação da Estrutura de Produto de Motores Elétricos**. 2007 (Dissertação de Mestrado), Programa de Pós-Graduação em Engenharia Mecânica. UFSC, Florianópolis. 2007.

MATELLI, J. A. **Sistemas Baseados em Conhecimento para Projeto de Plantas de Cogeração a Gás Natural**. 2008 (Tese de doutorado), Programa de Pós-Graduação em Engenharia Mecânica. UFSC, Florianópolis. 2008.

MATERIAL HANDLING AND LOGISTICS. **Delivery time top priority for online shoppers**, 2016. Disponível em <<https://www.mhlnews.com/transportation-distribution/article/22051729/delivery-time-top-priority-for-online-shoppers>>. Acesso em: 31 de dezembro de 2021.

MESEGUER, P., ROSSI, F., SCHIEX, T. Soft Constraints. In: ROSSI, F., BEEK, P., WALSH, T. **Handbook of Constraint Programming**. Elsevier: 1ª ed. 2014.

MLADINEO, M., ZIZIC, M. C., ALJINOVIC, A., GJELDUM, N. **Towards a knowledge-based cognitive system for industrial application: Case of personalized products**. Journal of Industrial Information Integration, 2021.

MORRIS, R. **The fundamentals of product design**. AVA Publishing: 1ª ed. 2009.

MYRODIA, A., KRISTJANSDOTTIR, K., HVAM, L. **Impact of product configuration systems on product profitability and costing accuracy**. Computers in Industry, n. 88, p. 12-18, 2021.

NGUYEN, Q. N., SIDOROVA, A., TORRES, R. **User interactions with chatbot interfaces vs Menu-based interfaces: An empirical study.** Computers in Human Behavior, n. 128, 2021.

NORDLANDER, T. E. **AI Surveying: Artificial Intelligence in Business.** Dissertação de Mestrado. Universidade de Montfort. 2001.

OLSEN, T. L., TOMLIN, B. **Industry 4.0: Opportunities and Challenges for Operations Management.** Manufacturing & Service Operations Management, p. 113-122, 2020.

PAKKANEN, J., LEHTONEN, T., JUUTI, T. **Observations on modular product structures based on teaching product modularisation.** 31st CIRP Design Conference 2021, 2021.

PIAGET, J. **O nascimento da inteligência na criança.** Ed. 4. Rio de Janeiro: Zahar, 1982.

PILLER, F. T, BLAZEK, P. **Core Capabilities of Sustainable Mass Customization.** Knowledge-Based Configuration, 2014.

PINTEA, C. M. **Advances in Bio-inspired Computing for Combinatorial Optimization Problems.** Springer: 1^a ed. 2014.

RICH, E., KNIGHT, K., NAIR, S. B. **Artificial Intelligence.** McGraw Hill Education: 3^a ed. 2009.

RODLER, P. **A formal proof and simple explanation of the QuickXplain algorithm.** Artificial Intelligence Review, v. 55, p. 6185-6206, 2022.

ROSENFELD, H. **Engenharia Simultânea.** Instituto de Gestão de Desenvolvimento de Produtos, São Paulo, 2014. Disponível em: <http://www.portaldeconhecimentos.org.br/index.php/por/Conteudo/EngenhariaSimultanea#etzoc121572_4>. Acesso em: 30 de fevereiro de 2022.

ROSENFELD, H., FORCELLINI, F. A., AMARAL, D. C., TOLEDO, J. C., SILVA, S. L., ALLIPRANDINI, D. H, SCALICE, R. K. **Gestão de Desenvolvimento de Produtos – uma Referência para Melhoria do Processo**. Saraiva: 1ª ed. São Paulo, 2006.

ROSSI, F., BEEK, P., WALSH, T. **Handbook of Constraint Programming**. Elsevier: 1ª ed. 2006.

SALVADOR, F., FORZA, C., RUNGTUSANATHAM, M. **How to mass customize: Product architectures, sourcing configurations**. Business Horizons, July-August, 2002.

SALVADOR, F., PILLER, F. T., AGGARWAL, S. **Surviving on the long tail: An empirical investigation of business model elements for mass customization**. Long Range Planning, n. 53, 2020.

SALVATIERRA, G., MATEOS, C., CRASSO, M., ZUNINO, A., CAMPO, M. **Legacy systems Migration Approaches**. IEEE Latin America Transactions, v. 11, p. 840-851, 2013.

SANDRIN, E., TRENTIN, A., GROSSO, C., FORZA, C. **Enhancing the consumer-perceived benefits of a mass-customized product through its online sales configurator**. Industrial Management & Data Systems, v. 117, n. 6, p. 1295-1315, 2017.

SHAO, X. **What is the right production strategy for horizontally differentiated product: Standardization or mass customization?** International Journal of Production Economics, n. 223, 2020.

SILVA, J. C. **Expert System Prototype for Hydraulic System Design Focusing on Concurrent Engineering Aspects**. 1998. (Tese de Doutorado). Pós-graduação em Engenharia Mecânica, UFSC, Brasil. 1998.

SILVA, J. C., MATELLI, J. A., BAZZO, E. **Development of a knowledge-based system for cogeneration plant design: Verification, validation and lessons learned.** Knowledge-Based Systems, 2014.

SILVER, D., HUBERT, T., SCHRITTWIESER, J., ANTONOGLU, I., LAI, M., GUEZ, A., LANCTOT, M., SIFRE, L., KUMARAN, D., GRAEPEL, T., LILICARP, T., SIMONYAN, K., HASSABIS, D. **Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm.** 19 p., 2017.

SLOAN, D. **Whatever Happened to Mass Customization?** 2014. Disponível em <<https://dave-sloan.medium.com/what-ever-happened-to-mass-customization-8f057e73fb5>>. Acesso em: 31 de dezembro de 2021.

STÄBLEIN, T., HOLWEG, M., MIEMCZYK, J. **Theoretical versus actual product variety: how much customization do customers really demand?** International Journal of Operations & Production Management, v. 31, n. 3, p. 350-370, 2011.

TAKAHASHI, R. **Do you know ABAP as a Programming Language?**, 26 de fevereiro de 2020. Disponível em <<https://blogs.sap.com/2020/02/26/do-you-know-abap-as-a-programing-language/>>. Acesso em: 28 de abril de 2022.

TIIHONEN, J., LEHTONEN, T., SOININEN, T., PULKKINEN, A. **Modeling configurable product families.** 4th Workshop on Product Structuring, Delft University of Technology, Delft, 1998.

TRATTNER, A., HVAM, L., FORZA, C., HERBERT-HANSEN, Z. N. L. **Product complexity and operational performance: A systematic literature review.** CIRP Journal of Manufacturing Science and Technology, v. 25, p. 69-83, 2019.

TRENTIN, A., PERIN, E., FORZA, C. **Sales configurator capabilities to avoid the product variety paradox: Construct development and validation.** Computers in Industry, ed. 64, p. 436-447, 2013.

TURING, A. M. **Computing Machinery and Intelligence**. Mind 49: p. 433-460. 1950.

VEGINI, F. F. R. **Protótipo de Sistema Baseado em Conhecimento para Configuração de Produto: Aplicação na Indústria de Revestimentos**. 2018 (Dissertação de Mestrado), Programa de Pós-Graduação em Engenharia Mecânica. UFSC, Florianópolis. 2018.

VULKOSAVIC, S. N. **Electrical Machines**. Nova Iorque: Springer, 2013.

WEG, 2019. **WEG em números**. Disponível em <<https://www.weg.net/institutional/BR/pt/weg-in-numbers>>. Acesso em: 11 de fevereiro de 2022.

WU, C. **Parallelizing a CLIPS-based course timetabling expert system**. Expert Systems with Applications, ed. 38, p. 7517-7525, 2011.

ZIPKIN, P. **The Limits of Mass Customization**. MIT Sloan Management Review, ed. 42, 2001.

APÊNDICE A – Detalhamento das implementações

APÊNDICE A1 – DETALHES DE IMPLEMENTAÇÃO DA ESTRUTURA DE MÓDULOS

Na *shell* CBBConf, essa estrutura é modelada utilizando-se alguns recursos. Para cada módulo e conjunto, é necessário criar uma classe Java, utilizar a anotação `@Component` como na Figura A.1:

Figura A.1 – Declaração de uma classe representando o componente motor

```
@Entity(name = "MOTOR")
@Audited
@ComponentType
public class Motor implements Component {
```

Fonte: Autoria própria

As portas, por sua vez, são modeladas utilizando a seguinte estrutura, conforme Figura A.2:

Figura A.2 – Declaração de uma porta para o conjunto da carcaça na classe do motor

```
@TranslatableNode
@OneToOne(fetch = FetchType.EAGER, cascade = CascadeType.ALL, orphanRemoval = true)
@PartPort
private FrameSet frameSet;
```

Fonte: Autoria própria

As características de cada módulo ou conjunto são modelados através de propriedades da classe Java, anotadas com uma das seguintes opções: `@RealProperty` para variáveis do tipo real (números reais), `@StringProperty` para variáveis do tipo *String* (no caso, a *shell* só permite *strings* pré-cadastradas do tipo chave-valor, não são aceitos textos livres), `@BooleanProperty` para variáveis do tipo booleanas (verdadeiro-falso) e `@IntegerProperty` para variáveis do tipo inteira (números inteiros), seguindo exemplo da Figura A.3:

Figura A.3 – Declaração de algumas propriedades na classe do ventilador

```

@BooleanProperty
private MaestroBoolean fanBalancing = new MaestroBoolean();

@RealProperty(ub = 10_000.0, optional = true)
private MaestroDouble fanHubInternalDiameter = new MaestroDouble();

@StringProperty(source = "WESTADO_PECA", repository = WDD3DAttributeDatabase.class)
private MaestroString componentState = new MaestroString();

```

Fonte: Autoria própria

No caso, são mostradas algumas propriedades do componente ventilador: o balanceamento do ventilador, que é uma propriedade booleana; o diâmetro inteiro do cubo do ventilador, que é uma propriedade real; e o estado do componente, que é uma propriedade textual. O diâmetro do ventilador é limitado pela instrução “*ub*”, que significa *upper boundary*, no valor de 10.000; apesar de extremamente improvável existir um ventilador tão grande na base de dados, é utilizado como padrão esse valor em todas as propriedades por simplicidade. No caso do estado do componente (que assume valores como “*usinado*”, “*fundido*” ou “*injetado*”), os valores são cadastrados na aplicação *Attributes* e buscados através da instrução “*source*”.

APÊNDICE A2 – DETALHES DE IMPLEMENTAÇÃO DOS DOMÍNIOS

A definição do domínio das variáveis é realizada usando a aplicação *Attributes* da *shell* da empresa-alvo, conforme exemplo da Figura A.4:

Figura A.4 – Aplicação Attributes administrando o atributo “Design”

DESIGN						
Propriedades						
* Nome:	DESIGN	ID	Value	Java	PT	EN
Tipo:	String	24957	00001	W22	W22	W22
Descrições das Linguas		24958	00002	W21	W21	W21
Java:	AttrDesign	24959	00003	HGF	HGF	HGF
PT:	Design	24960	00004	W22_NEMA	W22 NEMA	W22 NEMA
EN:	Design	24961	00005	W22_BRAKEMOTOR	W22 Motofreio	W22 Brakemotor
ES:		24962	00006	W22_SINGLE_PHASE	W22 Monofásico	W22 Single phase
		24963	00007	W22_GEARMOTOR	W22 Motorreductor	W22 Gearmotor
		24964	00008	W50	W50	W50
		24965	00009	W22_P_BASE	W22 P-Base	W22 P-Base
		24966	00010	W23	W23	W23
		24967	00011	W22_COOLING_TOWER	W22 Cooling Tower	W22 Cooling Tower
		24968	00012	W12	W12	W12
		48963	00013	NA	NA	NA
		78952	00014	W51	W51	W51

Fonte: Acervo interno da empresa-alvo

Na figura, é possível visualizar as capacidades dessa aplicação, definindo de maneira similar ao SCP atual da empresa-alvo, os valores em um formato de código (00001, 00002, etc...) e suas traduções. Esse método de cadastro de valores é importante, pois permite facilmente que o sistema opere em diferentes idiomas.

APÊNDICE A3 – DETALHES DE IMPLEMENTAÇÃO DOS VALORES DISPONÍVEIS

No código-fonte, é necessário fazer uma conexão entre as colunas da tabela preenchida na aplicação TableConstraints e as variáveis do modelo. Esse tipo de conexão é exemplificado na Figura A.5:

Figura A.5 – Exemplo de codificação de uma restrição tabular

```

@Constraint(description = "Fan cover model")
public void fanModulesVariants() {
    globalConstraints.table( content: "MAT_VRBIW_001.json",
        fanMaterialNumber,
        fanBalancing,
        fanExternalDiameter,
        fanHubInternalDiameter,
        fanComponentState,
        fanKeyWidth,
        fanHubWidth,
        fanMaterial,
        fanBladeQuantity,
        fanCoating,
        fanFixingSystem).
        post();
}

```

Fonte: Autoria própria

A lógica de conexão das colunas com os parâmetros é a seguinte: o primeiro parâmetro dentro da instrução *table* é o nome da tabela, no caso, “MAT_VRBIW_001”. O segundo parâmetro corresponde à primeira coluna da tabela cadastrada na aplicação *Table Constraints*, no caso, a propriedade *Material*; o terceiro parâmetro corresponde à segunda coluna, e assim sucessivamente.

As restrições do tipo expressão, utilizadas para consistências inter-componentes são exemplificadas na Figura A.6. Nessa figura, tem-se a relação de igualdade entre diâmetros do ventilador e eixo:

Figura A.6 - Restrição de montagem ventilador – eixo

```

@Constraint(description = "Fan must fit in shaft")
public void fanMustFitInShaft() {
    fanHubInternalDiameter.eq(shaftFanSeatDiameter).post();
}

```

Fonte: Autoria própria

Nesse trecho de código, equiparam-se os dois diâmetros através do operador “eq”, que implica igualdade entre as variáveis *fanHubInternalDiameter* e *shaftFanSeatDiameter*. A

instrução *post* usada ao final da linha serve para que o *solver* de restrições considere essa restrição na propagação. Casos mais complexos que envolvem operações aritméticas ou trigonométricas também podem ser construídas de maneira similar.

APÊNDICE A4 – DETALHES DE IMPLEMENTAÇÃO DOS VALORES PADRÕES

Na *shell* CBCConf, os valores padrões podem ser definidos utilizando-se um recurso equivalente à restrição *soft* discutida na seção 3.3.3, mas na *shell* ela é um tipo específico de restrição, sendo denominada restrição do tipo “recomendação”. Por este motivo, é equivalente a uma restrição comum, mas anotada com a instrução *recommendation = true*. Pode ser também uma tabela, conforme exemplo mostrado na Figura A.7:

Figura A.7 – Exemplo de recomendação na shell CBCConf

```
@Constraint(description = "Marketing initial definition", recommendation = true)
public void marketingInitialDefinition() {
    globalConstraints.table(String.valueOf(motorMarket),
        motorPhases,
        motorEnclosure,
        motorLine,
        motorVariantLine,
        motorEnclosureMaterial,
        motorFrequency,
        motorStandard).
    post();
}
```

Fonte: Autoria própria

Nesta figura, é possível visualizar a restrição tabular que relaciona oito características iniciais do configurador, de maneira similar à que existe no configurador atual, mostrado na Figura 4.17.

APÊNDICE A5 – ITENS SELECIONADOS PELO CONFIGURADOR DE PRODUTO

Conforme discutido, na Tabela A.1 são apresentados os itens encontrados na *bill-of-materials* dos 10 produtos selecionados para validação, comparando-os com os itens selecionados pelo protótipo.

Tabela A.1 – Itens reais dos produtos versus itens selecionados pelo protótipo

Teste de seleção		Carcaça		Tampa Dianteira		Tampa Traseira		Eixo		Tampa Defletora		Caixa de ligação	
Produto	Tamanho do produto	Real	Protótipo	Real	Protótipo	Real	Protótipo	Real	Protótipo	Real	Protótipo	Real	Protótipo
1	63	10922607	10922607	10947048	10947048	10947909	10947909	11369708	11369708	10161599	10161599	10897454	10897454
2	71	10922320	10922320	10942872	10942872	10943118	10943118	10334513	10334513	10025638	10025638	10897454	10897454
3	80	10922239	10922239	10949331	10949331	10951490	10951490	10023206	10023206	10025640	10025640	10897454	10897454
4	90L	10835888	10835888	10841257	10841257	10844339	10844339	10322805	10322805	10025642	10025642	10897454	10897454
5	100L	10826135	10826135	10811235	10811235	10822674	10822674	10023193	10023193	10161580	10161580	10897454	10897454
6	112M	10868964	10868964	10869397	10869397	10877327	10877327	10023197	10023197	10025643	10025643	10897568	10897568
7	132S	11296914	11296914	11035723	11035723	10024020	10024020	10322806	10322806	10025645	10025645	10897568	10897568
8	160M	11132009	11132009	10865816	10865816	11077833	11077833	10023204	10023204	10865947	10865947	11226408	11226408
9	180M	11001659	11001659	10945637	10945637	10946874	10946874	10372314	10372314	10958149	10958149	11226408	11226408
10	200M	10933437	10933437	10886288	10886288	10886932	10886932	10024471	10024471	10887600	10887600	11226435	11226435

Fonte: Autoria própria

Como é possível visualizar, todos os itens selecionados são os mesmos, comprovando que com o conjunto de restrições inter e intra-componentes construídas, o sistema pode se comportar da forma esperada.

APÊNDICE A6 – EXEMPLO DE REGRA DESNECESSÁRIA NO SISTEMA PROPOSTO

Abaixo, na Figura A.8, é apresentada a regra ID 9 do sistema atual de configuração, com a seguinte descrição “MOTORES COM FLANGE FF OU D NEMA SUPERIOR E TIPO DE VEDAÇÃO RETENTOR COM MOLA, OBRIGATÓRIO CONFIGURAR SEM ANEL DE FIXAÇÃO”.

Figura A.8 – Regra desnecessária no novo sistema

The screenshot displays the configuration tool interface for Rule ID 9. The rule is active and has a status of 'Released'. The reason for creation/review is 'ALTERADO CONFORME CLAIM 414844759, PMC- 09 /2016'. The rule is associated with the user 'FABIULA' and was created on '16.01.2018'.

The 'Determining characteristic' section shows the following conditions:

Rule ID	Rv	Active	Characteristic	Value	Logical operator	Value if Logical. op.	Product type	Date	User
9	4	Yes	ZGRUPO_CARACA_01	CARCACA 71	Equal to	Any product type	16.01.2018	FABIULA	
9	4	Yes	ZGRUPO_CARACA_01	CARCACA 80	Equal to	Any product type	16.01.2018	FABIULA	
9	4	Yes	ZGRUPO_CARACA_01	CARCACA 90	Equal to	Any product type	16.01.2018	FABIULA	
9	4	Yes	ZTIPO_TAMANHO_FLANGE_01	FLANGE FF OU D NEMA SUPERIOR	Equal to	Any product type	16.01.2018	FABIULA	
9	4	Yes	ZTIPO_VEDAÇÃO_DIANTEIRA_01	RETENTOR COM MOLA	Equal to	Any product type	16.01.2018	FABIULA	

The 'Condition 1' section shows the following conditions:

ID	Rv	Nome caract.	Value	Condition 1 - options
9	4	ZAPLICACAO_MOTOR_01	COOLING TOWER W22	3.The rule does not run if any value is true.
9	4	ZAPLICACAO_MOTOR_01	INVERTER DUTY W22	3.The rule does not run if any value is true.
9	4	ZAPLICACAO_MOTOR_01	JM/JP/JET PUMP W22	3.The rule does not run if any value is true.
9	4	ZAPLICACAO_MOTOR_01	MINING W22	3.The rule does not run if any value is true.
9	4	ZAPLICACAO_MOTOR_01	VENTILACAO W22	3.The rule does not run if any value is true.
9	4	ZAPLICACAO_MOTOR_01	W22	3.The rule does not run if any value is true.
9	4	ZAPLICACAO_MOTOR_01	W22xec SEGURANCA AUMENTADA	3.The rule does not run if any value is true.
9	4	ZAPLICACAO_MOTOR_01	W22xtb DIP	3.The rule does not run if any value is true.

The 'Condition 2' section shows the following conditions:

ID	Rv	Nome caract.	Value	Condition 2 - options
9	4	ZFIXACAO_ROLAMENTO_01	ANEL DE FIXACAO	1.The rule runs if any value is true.

Fonte: Autoria própria

Como mostrado na figura, essa regra possui diversas condições e características, acrescentando complexidade para o sistema atual e exigindo manutenção desses valores. A eliminação dessa e outras regras de caráter redundante para com as restrições intra-componentes traz um enorme benefício ao sistema proposto.

APÊNDICE A7 – DETALHAMENTO DAS RESTRIÇÕES PARA PRIMEIRO ESTUDO DE CASO

Para que o sistema se comporte conforme mencionado, é necessário estruturar duas restrições. A primeira é mostrada abaixo na Figura A.9:

Figura A.9 – Primeira restrição para o primeiro estudo de caso

```
@Constraint(description = "Necessário avaliação de nova variante de caixa de ligação", breakable = true)
public void terminalBoxModulesVariants() { terminalBoxItemNumber.ne( y: "-").post(); }
```

Fonte: Autoria própria

Conforme mostrado no excerto de código, trata-se de uma restrição quebrável, isto é, o mecanismo de resolução de conflitos irá avaliar a possibilidade de desliga-la para resolver o conflito. Além disso, na descrição, é colocada a mensagem de que é necessária avaliação de nova variante de caixa de ligação, pois isso só ocorrerá na situação dessa restrição ser desligada pelo mecanismo de resolução de conflitos, momento no qual essa mensagem será apresentada na tela para o usuário. E na expressão, a regra indica que o *itemNumber* da caixa de ligação não deve ser o traço, pois normalmente ele deve assumir um código de 8 dígitos.

A segunda restrição é mostrada na Figura A.10:

Figura A.10 – Segunda restrição para o primeiro estudo de caso

```
@Constraint(description = "Lógica de seleção do estado de componente da caixa de ligação")
public void terminalBoxComponentStateLogic() {
    terminalBoxItemNumber.eq( y: "-").imp(terminalBoxComponentState.member( ...strValues: "00002", "00003")).post();
    terminalBoxItemNumber.ne( y: "-").imp(terminalBoxComponentState.member( ...strValues: "00001")).post();
}
```

Fonte: Autoria própria

Essa restrição é uma lógica de negócio, de modo que se o valor do *itemNumber* for o traço, o estado do componente assume os valores 00002 ou 00003 (SOB CONSULTA ou IMPOSSÍVEL, respectivamente), enquanto se não for o traço, o valor é “LIBERADO”.

